Learning to Predict Lidar Intensities

Patrik Vacek, Otakar Jašek, Karel Zimmermann, *Member, IEEE*, and Tomáš Svoboda, *Member, IEEE*

***Abstract*— We propose a data-driven method for simulating lidar sensors. The method reads computer-generated data, and (i) extracts geometrically simulated lidar point clouds and (ii) predicts the strength of the lidar response – *lidar intensities*. Qualitative evaluation of the proposed pipeline demonstrates the ability to predict systematic failures such as no/low responses on polished parts of car bodyworks and windows, or strong responses on reflective surfaces such as traffic signs and license/registration plates. We also experimentally show that enhancing the training set by such simulated data improves the segmentation accuracy on the real dataset with limited access to real data. Implementation of the resulting lidar simulator for the GTA V game, as well as the accompanying large dataset, is made publicly available.**

***Index Terms*— Robotics, simulation, sensor development, machine learning, neural network applications, intelligent transportation systems.**

I. Introduction

There have been over 1.2 billion vehicles in use over the world in 2015.[1] When a novel autonomous functionality, such as autonomous emergency braking, is to be put into operation, its reliability has to be thoroughly tested, because the impact on the accident rate is enormous. For example, if the new functionality exhibit 1 failure out of 1 million testing frames (9 hours of operating time of 30Hz sensor), the expected number of failure cases over the world per single day is over 150 million.[2] Consequently, testing on billions of frames in advance of real deployment is highly desired. It is hardly feasible to create testing set with billions of annotated frames which would cover all possible cases. In addition to that, many tasks comprise online control, which cannot be tested offline. A trustworthy simulation is the only technically tractable option.

There are several open-source simulators such as CARLA [2] or AirSim from Microsoft [3], which offer viable autonomous driving simulation with a realistic RGB camera model in a small synthetic world with a limited variety of textures and structures. In contrast to these open-source simulators, research community also reverse-engineered GTA V game engine. The mentioned game has been recently shown [4] to have a world model realistic enough for generating annotated training RGB images that improve performance on well known semantic segmentation challenges KITTI [5] or VOC [6]. Nevertheless, the simulation of other sensors, which are also essential for autonomous-driving such as lidars, is either missing (GTA V) or it is strictly geometry-based (CARLA).

Unfortunately, lidar point clouds consisting of geometry only lack information about the power of a receiving signal (*lidar intensity*) and therefore are not fully descriptive for modeling and evaluation of lidar sensor with full properties. Importance of including this lidar intensity as a feature has been demonstrated by [7] as it increases performance in semantic segmentation. The naive approach to model intensity feature is to map it as a monotonically decreasing function of depth. However, depth-based intensity undesirably underestimates behavior in the corner cases with unusual dispersion of the active signal, such as polished hoods, windows, and shallow puddles, registration plates, or traffic signs, see Figure 1 for a few examples. The material behavior is described by another contributing factor of the received signal, the reflectivity of the scanned objects [8]. However, the procedure of acquiring realistic material responses to the lidar beam in the simulation world would require large-scale physical specifications of generated objects. We propose to leverage other information about the object, such as its color and label description and study benefits of these modalities in prediction of lidar response learned from driving scenarios of the real world.

To close the gap between the real and synthetic data, we introduce and publicly release a GTA V lidar simulator. The simulator is trained on the real data to estimate realistic responses on unusual surfaces. The proposed method builds on top of the geometrical model, which re-projects the existing world into the lidar sensor. We enhance the geometrical model by modeling the strength of the lidar response. Modeling intensities allows injecting systematic failures and measurement noise into the geometrically simulated measurements. We experimented with two deep learning architectures [9] and [10] to learn the intensity estimation in a data-driven way. The intensity model is further used for enhancing synthetic data. We show that such data, when combined with the real training set, improves the segmentation accuracy on real testing data.

Contributions of this paper are four-fold: (i)

1) We propose a way of modeling intensity from the lidar geometry, RGB images and class label.

Manuscript received October 30, 2019; revised July 10, 2020; accepted October 14, 2020. Date of publication January 28, 2021; date of current version March 29, 2022. This work was supported in part by the OP VVV MEYS "Research Center for Informatics," under Grant CZ.02.1.01/0.0/0.0/16_019/0000765; in part by the Czech Science Foundation under Grant 20-29531S; and in part by the Grant Agency of the CTU Prague under Grant SGS20/128/OHK3/2T/13. The Associate Editor for this article was J. W. Choi. *(Corresponding author: Patrik Vacek.)*

The authors are with the Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, 16636 Prague, Czechia (e-mail: vacekpa2@fel.cvut.cz).

Digital Object Identifier 10.1109/TITS.2020.3037980

[1] https://www.statista.com/statistics/281134/number-of-vehicles-in-use-worldwide

[2] This number is estimated as follows: Expected number of failures of a single car during one day is $\mathbb{E}[\mathrm{Bin}(24, 1/9)] = 2.5$. 95 % of vehicles are parked while the remaining 60 million cars are in motion [1], therefore the expected number of failures is $2.5 \times 60 \cdot 10^6$.

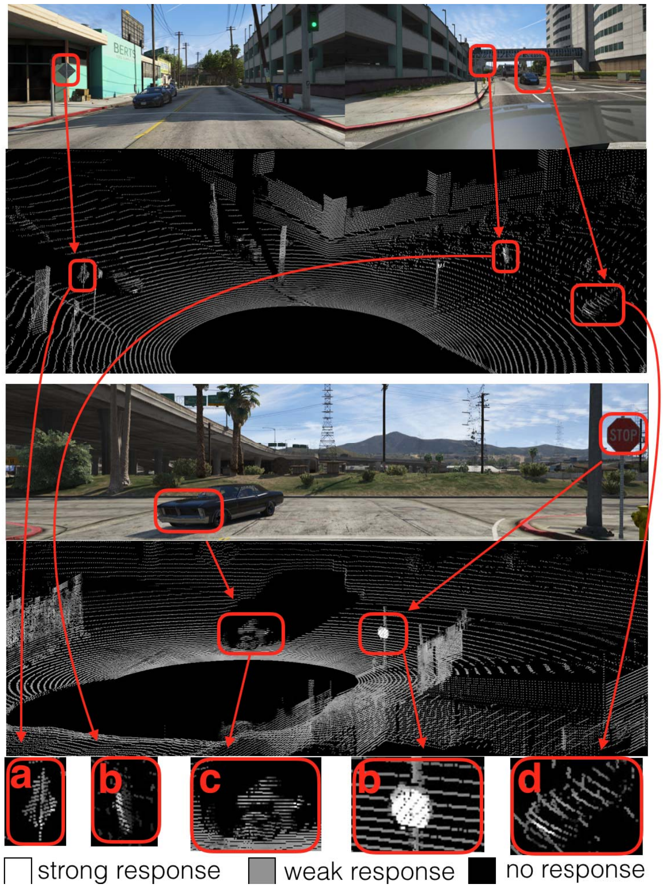


Fig. 1. Examples of simulated data: Simulated RGB image and close-up of corresponding lidar scan with intensity encoded in grayscale. Strong responses appear consistently on reflective surfaces such as traffic signs facing towards the lidar (b) and license plates (d) despite the shadows in RGB images. Notice also correctly simulated systematic failures: (i) no or weak responses on the hood (c+d), (ii) weak response on the frontal mask of the bottom car which does not have a license plate (c), (iii) weak response on the traffic signs in the top image, which is facing from the lidar (a).

2) We show that the data-driven simulation of lidar measurements, when combined with real training dataset, improves the segmentation accuracy on the real data.
3) We provide a publicly available lidar interface for the GTA V game, which allows for the automatic generation of synthetic annotated training and evaluation datasets.
4) We provide a large public GTA V dataset for object detection and semantic segmentation from RGB+lidar data, which consists of approximately 40 000 frames.

Both source codes and dataset are available for download at https://github.com/vras-group/lidar-intensity.

II. Related Work

A. Large-Scale Lidar Datasets

Recent advancements in the field of autonomous driving were influenced by large-scale datasets and benchmarks. This phenomenon is even more significant with the thriving success of deep learning. Kitti benchmark has set standards among public automotive datasets [5]. Besides regular RGB images of driving scenes, it also includes calibrated lidar readings. However, the annotation is done only from the RGB cameras and therefore limited to the frontal view only. This limitation has been eliminated by the very recent SemanticKitti dataset [11], where all points in lidar point clouds were annotated, excluding a few anomalies. NuScenes [12] is a recently published dataset that contains thousand driving scenarios. It is composed of 360 thousand lidar readings, which also include full annotations.

However, in order to build a fully autonomous vehicle, datasets of much larger magnitudes and different scenarios are necessary. Manual annotation is costly and consumes a large amount of man-hours [13], which makes it intractable for such a large scale. On top of that, datasets alone do not provide options for *validation* of autonomous driving capabilities with respect to the interpreted scene. These constraints point to the necessity of realistic and automatically annotated simulators.

B. Simulators With Lidar Point Cloud Properties

It was shown by numerous papers, that many state of the art detectors use intensity channel as a useful feature in learning segmentation from lidar [14]–[17] measurements. Intensity can provide a decisive distinction between two objects of a similar geometry by providing peak values on specific object parts similar to attention models [18]–[20] and therefore constitutes a valuable feature for classification tasks. Unfortunately, most of the current lidar simulators capable of creating a variable driving scene do not compute intensity values and offer geometry only [2], [7], [21]. Carla simulator [2] contains information about surface material, however, as far as we can tell, it cannot be leveraged to acquire lidar intensity. The simulator Blensor [22] offers information about material reflectivity. However, it is not possible to simulate different weather conditions. The Blensor also suffers from the fact that its base Blender was not developed for large scenes, but rather for smaller objects, and therefore, it is difficult to model a large world at the needed scale. The Virtual KITTI dataset [23] provides synthetically generated sequential images with depth information and pixel-wise annotation. The depth information can also be used to generate point clouds. However, the point clouds do not show the same characteristics as a real rotating lidar, including reflections.

Another option is to use computer games with state-of-the-art graphics, such as GTA-V. Driving in the Matrix [24] and Playing for Data [25] unlock the possibility of using a game engine for data gathering. However, Driving in the Matrix lacks finer annotation as it only extracts the stencil layer from the game, which does not differentiate between many object classes, and Playing for Data still requires a semi-manual labeling procedure. Both of these works also lack the ability of the lidar sensor, however, it can be circumvented by placing virtual cameras at the desired locations, as will be shown in this work. GTA-V engine was also exploited by [7], where geometrical lidar has been simulated in the frontal camera view. It comes, however, without intensity properties, and class labels of objects' 3D shape are approximated by bounding boxes only. GTA-V world has no concept of reflectivity of the material, and therefore, the returned lidar reflections are missing.

C. Simulation of Intensity

Lidar intensity is derived from three main components: geometric, physical, and environmental model of lidar [8]. The Geometric part is usually solved by basic computer vision algorithms such as ray-casting and projection of the points [26]. Physical and environmental models consist of various sensors and surrounding constants and target properties, which is usually not available in simulation [27]. These modalities are mainly reflectivity of material and beam divergence of a laser.

Work of SqueezeSegv2 [9] tried to model intensity using data; however, it resorted to using geometric information only. A recent work [28] tries to close domain differences between real and synthetic data by modeling echo pulse width (EPW) of the laser via [10] to substitute intensity. However, despite the fact that EPW is part of the lidar resulting intensity, the work [29] shows its lack of representativeness as a sole intensity indicator. Two objects can share the same EPW but have different reflectivity, so they do not cause the same resulting intensity. This work also shows that the implementation of EPW did not improve the model performance, as it is not descriptive enough feature for classification algorithms.

To the best of our knowledge, there is no other previous work trying to model a lidar intensity data-driven way. Also, none considered modeling intensity from RGB information or any other modalities besides geometric.

III. Methods

The proposed pipeline is summarized in Figure 6. The lidar simulation employs four virtual roof-mounted cameras, which provides four temporally synchronized streams of RGBD images at a user-defined framerate. Four-tuples of depth images are converted into 360° point clouds respecting the geometry of the simulated lidar. This part is briefly summarized in Section III-A. The resulting point clouds are deprived by random drop noise to rays following the same procedure from [9]. Finally, the lidar intensity is predicted by a single deep convolutional network. The intensity predicting network, as well as the learning procedure, are detailed in Section III-B.

A. Geometrical Simulation

The geometrical simulation of the lidar consists of two consecutive steps, which are briefly illustrated in Figure 2. First, a dense point cloud is generated from four temporally synchronized RGBD images and the known camera calibration matrices. For each pixel of the virtual RGBD camera, we generate a corresponding 3D point $\mathbf{x}_{ego}$ in the camera coordinate frame as follows

$$\bar{\mathbf{x}}_{ego} = \begin{bmatrix} x_{ego} \\ y_{ego} \\ z_{ego} \\ 1 \end{bmatrix} = \mathbf{P}^{-1} \begin{bmatrix} x_{cam} \\ y_{cam} \\ D \\ 1 \end{bmatrix}. \tag{1}$$

$\bar{\mathbf{x}}_{ego}$ are homogenous coordinates of the 3D point in a car coordinate system, $\mathbf{P} \in \mathbb{R}^{4\times4}$ is a camera projection matrix, $\{x, y\}_{cam}$ are coordinates of each pixel in an image (normalized to the range [-1, 1]) and D is the depth of each pixel.

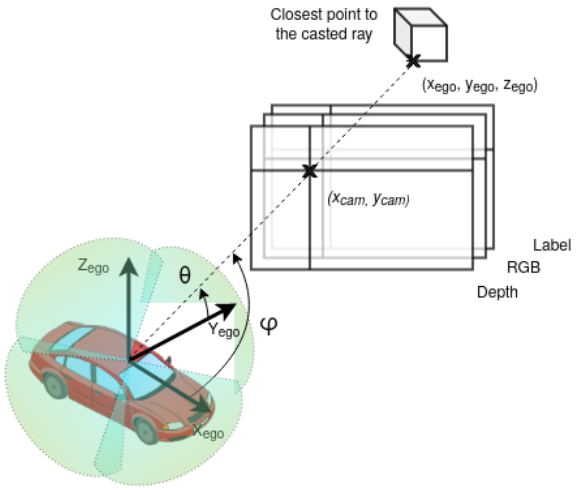
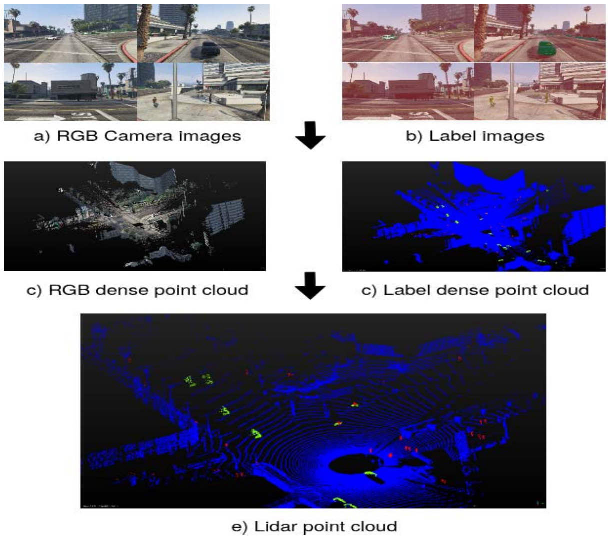


Fig. 2. Extracting of lidar point clouds - By placing four virtual cameras on top of the car, we acquire images of a surrounding scene with depth, RGB, and label information. From these depth images, we construct dense point clouds in car-ego coordinates using a camera projection matrix (1). Then ray-casting procedure chooses the closest point in dense point cloud corresponding to lidar's angular resolution ϕ, θ and maximum range. As a result, newly created lidar point cloud of specific sensor parameters is obtained with all game source information (e.g. coordinates x_{ego}, y_{ego}, z_{ego}, RGB, Label) for every scan point.



a) RGB Camera images

b) Label images

c) RGB dense point cloud

c) Label dense point cloud

e) Lidar point cloud

Fig. 3. Example of point cloud extraction - we model Velodyne HDL lidar with 64 layers in 360° FOV. The procedure consists of the following steps: (a) Place four virtual cameras, which cover 360° FOV, to the position of the lidar. (b) Extract corresponding labels from the stencil buffer. (c+d) Reconstruct dense point clouds from all four cameras. (e) Estimate final point cloud by ray casting and dumping points exceeding maximal the range of the sensor.

Resulting dense point cloud of $1920 \times 1200 = 2304000$ 3D points is transformed into world coordinate system using

$$\bar{\mathbf{x}}_{world} = \mathbf{W}^{-1}\bar{\mathbf{x}}_{ego}, \tag{2}$$

where $\bar{\mathbf{x}}_{world}$ are homogenous coordinates of the 3D point in the world coordinate system and a world matrix $\mathbf{W} \in \mathbb{R}^{4\times4}$ is a transformation matrix. Matrices $\mathbf{W}$ and $\mathbf{P}$ are obtained from the RAGE engine of GTA V.

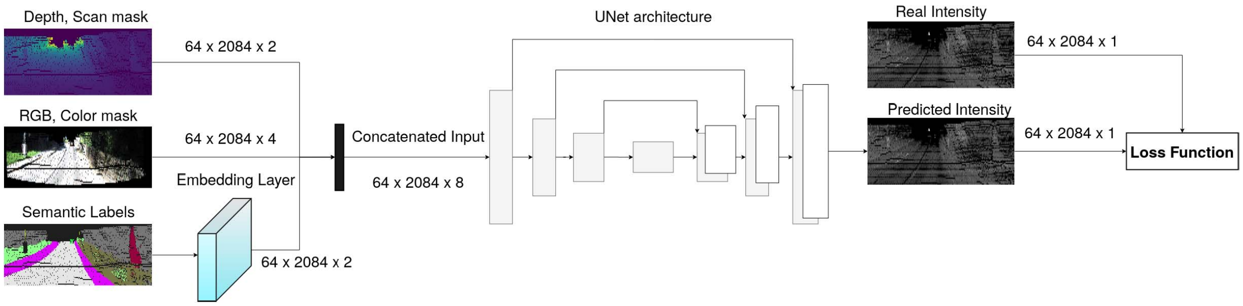


Fig. 4. Modeling intensity from modalities - We use depth lidar measurements, calibrated images from the camera with lidar reading and existing labels from the SemanticKitti dataset as a source of input channels to the neural network. Inputs are sent into the neural network in form of an image-like grid with channels corresponding to the modalities, where label modality is embedded through the embedding layer resulting in a two-dimensional channel grid. We then compare intensity from real data to prediction in L2 loss function and train the model with backpropagation.

3D points from all four cameras in the world coordinates are then concatenated into one dense point cloud, and points which are further than 130 m (operating range of commercial lidars) from the cameras' centers are then discarded. This results in a dense 3D point cloud with approximately 7×10^6 points. More technical details on extracting these dense point clouds can be found in [30].

Second, rays corresponding to the real lidar geometry (i.e., angular resolution and vertical field of view) are cast on the dense point-cloud, and the closest corresponding 3D points are extracted. Since horizontal FOV of the RGBD cameras is 91° and image width is 1920 pixels, the angular resolution of the dense point cloud is approximately 0.047°, which is approximately 3.65× finer horizontal resolution than that of the commercial lidar (Velodyne HDL-64E has a horizontal angular resolution of 0.1728°). The output of this procedure is a geometrically consistent point cloud.

We found that even though it is much more computationally demanding to generate this geometrically precise lidar representation *outside* the RAGE engine, it is also much more precise since ray-casting implemented within the RAGE engine approximates the 3D shape of the object by a bounding box as in [7].

B. Data-Driven Intensity Simulation

Since we do not know the exact parameters of the lidar sensor and the reflectivity of surfaces in the simulated world, we cannot calculate intensity values directly during the simulation process. We overcome this drawback by learning to predict intensity levels from the real measurements in a data-driven way. The physical properties of "beamed" laser and received signal energy can be described as fixed sensor configuration and inconsistent environmental parameters using the lidar equation [8]. This lidar equation models the power of received lidar signal P_r, which is directly correlated to resulting intensity value I via normalization and calibration of the specific sensor. Since we model intensity using real data, this conversion will be included when modeling the same sensor. Lidar equation [8] models the power of the received signal as follows:

$$P_r = \frac{P_t D_r^2}{4\pi r^4 \beta_t^2} \eta_{sys} \eta_{atm} \sigma \tag{3}$$

The received signal intensity P_r can be calculated from transmitted signal power P_t, receiver aperture diameter D_r, traveled distance of laser to the target r, laser beam width β_t, sensor-specific parameter η_{sys}, atmospheric transmission factor η_{atm}, and back-scattering cross-section σ, which depends entirely on the target characteristics. Except for σ, all other parameters are defined in constant lidar configuration. Signal power (P_t), laser beam width (β_t), sensor parameter η_{sys} and aperture diameter (D_r) are constants for specific lidar. Environment factor (η_{atm}) does not diverse along measuring sequence in the same weather conditions and range from the target is known from our geometric simulation. Then we need to consider target contribution to the intensity, which is modeled by previously mentioned cross-section σ, denoted as follows:

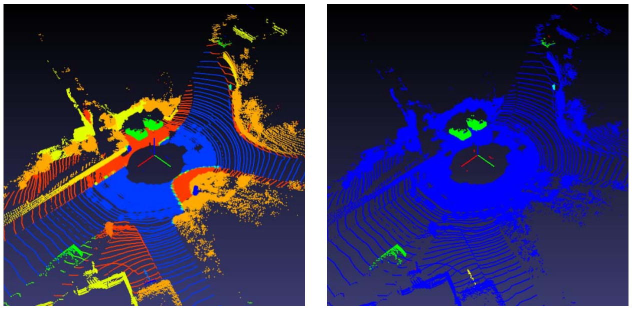
$$\sigma = \frac{4\pi}{\Omega} \rho_s A_s \tag{4}$$

where Ω is the scattering solid angle (divergence) of a laser beam, A_s is the target area, and ρ_s is the target's material spectral reflectance. The parameters depend on the geometry and reflectivity of the scanned object, i.e., the property of its material. We can leverage geometry information in our simulator, but it does not offer any information about reflectivity. We assume that material can be estimated based on its color and possibly by information about the type of the object consisting of that material, i. e. class label. Lidar does not contain any information about RGB color. To compensate for the lack of RGB, we use a multi-sensor dataset [11], which has camera images calibrated with respect to the lidar. From these camera images, we project RGB channel to lidar scan points. This dataset also comes already annotated with class labels.

In contrast to others, we suggest exploiting all modalities available during the simulation – RGB colors, depth, *and* semantic labels. We train a deep convolutional neural network to predict the intensity from the multi-modal data.

C. Learning of the Intensity-Predicting Network

The intensity-predicting network is trained on the real data obtained from the SemanticKitti dataset [11]. This dataset contains 360° lidar scans, pixel-level labels, and RGB images,



(a) Kitti frame with all 19 labels (b) Kitti with GTA labels only

Fig. 5. Label diversity in real and synthetic domain - Example of annotations in both worlds. The Real dataset has a much richer description of present objects, but thanks to the nature of the simulation world, it is feasible to introduce new categories in the engine in the future.

which are however available for the forward view only. We argue, that the training on the forward view generalizes well on other views because the testing accuracy in other views is comparable.

The learning process is outlined in Figure 4. To simplify the learning process, all measured modalities are projected to the cylindrical projection with a center placed at the position of the lidar and mapped as channels in a grid consisting of single lidar beams. Since there is a natural dropout in rays during lidar sweeps, we add a binary logic mask of successfully returned rays. Similar binary mask is also added for RGB color, which is assigned to lidar rays that correspond to RGB in camera projection. Consequently, the SemanticKitti dataset is converted to the set of multi-channel 2D images containing depth (D), red (R), blue (B), green (G), label (L), intensity (I), ray mask (M) and color mask (CM) values.

We work with the four following input combinations of the intensity-predicting network: D, D+L, D+RGB, D+RGB+L, which are all trained to predict the intensity channel (I) from the aforementioned inputs. The proposed network extends the existing architecture of Unet [10]. We also experiment with SqueezesegV2 architecture for comparison with [9]. Contrary to [9], we omitted XYZ channels, because these do not generalize when trained only on the forward view. In particular, D network is identical to the SqueezeSegV2 without XYZ channels, and the other networks extend the dimensionality of the input layer accordingly while keeping the other layers the same. Especially 4-class label modality (L), where every class value is transformed through embedding layer into a two-dimensional vector (i.e., it adds two additional input channels).

SemanticKitti dataset contains 19-class label descriptions. However, our GTA simulator is able to produce 4-class unique labels, see the comparison in Fig 5. With a more diverse object categories, label modality increase precision of intensity prediction, as can be seen in Table II. That implies the potential usefulness of label feature in intensity prediction, however due to lack of categories in the current GTA simulator, we stick to the 4-class label (car, pedestrian, bicycle, background - all others).

Predicting the intensity can be seen as a regression problem. Work [9] proposes the hybrid loss, which classifies intensity values to bins and also regresses the deviation from the classified bin, as it, according to [9], should yield better results of prediction compared to L2 loss. We experiment with both types of losses. As done in [9], our classification in hybrid loss is split into 10 bins distributed over the density of intensity value, and deviation from classified bin was predicted as another output channel from the model. Therefore in the case of hybrid loss, our prediction model has ten outputs channels for bin classification and one for regression of the deviation. The channels are then summed to the resulting intensity value, which is compared to real intensity value by a mean squared error (MSE). We trained and validated the model of intensity on the SemanticKitti dataset, compared them with training using masked L2 loss (5). Mask in L2 loss corresponds to the (M) input channel. Opposed to [9], masked L2 loss showed to be superior in our case, as can be seen in Table II.

$$\mathcal{L} = \frac{1}{n} \sum_{i,j} (I_{i,j} - \hat{I}_{i,j})^2 \cdot m_{i,j}, \tag{5}$$

where i, j denotes pixel coordinates in grid-like image, I real intensity value, $\hat{I}$ predicted intensity value, m binary mask of returned scan points and n number of successfully returned rays in grid frame (i.e. the sum of m) to get mean value of loss function.

IV. Experiments

We evaluate the proposed intensity predictor in two ways: i) intensity prediction accuracy, see Section IV-A which shows how close are the predicted intensities to the real ones, and ii) improving segmentation accuracy when using intensity prediction see Section IV-B which demonstrates that extending the real training set by simulated point clouds improves the segmentation accuracy. The intensity prediction model was trained on 10000 lidar frames and tested on 2792 lidar frames recorded in the spatially distinct environments from the SemanticKitti dataset. In the segmentation experiment, we used a smaller portion of real dataset to study the impact of highly scalable simulated data.

A. Intensity Prediction Accuracy

We evaluate intensity prediction accuracy on every pixel from the lidar grid in terms of the mean squared error (MSE). As a prediction model, we use neural networks with encoder-decoder structure that contains skip connections in order to preserve high as well as low-level features. As long as the model is expressive enough and has a specific number of inputs (D, DL, D+RGB, D+RGB+L), it is possible to adopt different model architectures such as [18], [20] and fine-tune them for high performance.

We compare four different input combinations D, D+L, D+RGB, D+RGB+L and two different loss functions for SqueezeSegV2 - Hybrid loss from [9] and L2 loss. See Table I for details on classification values distribution and architectures. We also experiment with the aformentioned architectures of Unet neural network [10] with L2 loss and omit the hybrid loss, since we achieved consistently better performance with L2 loss in all tested modalities with SqueezeSegV2.

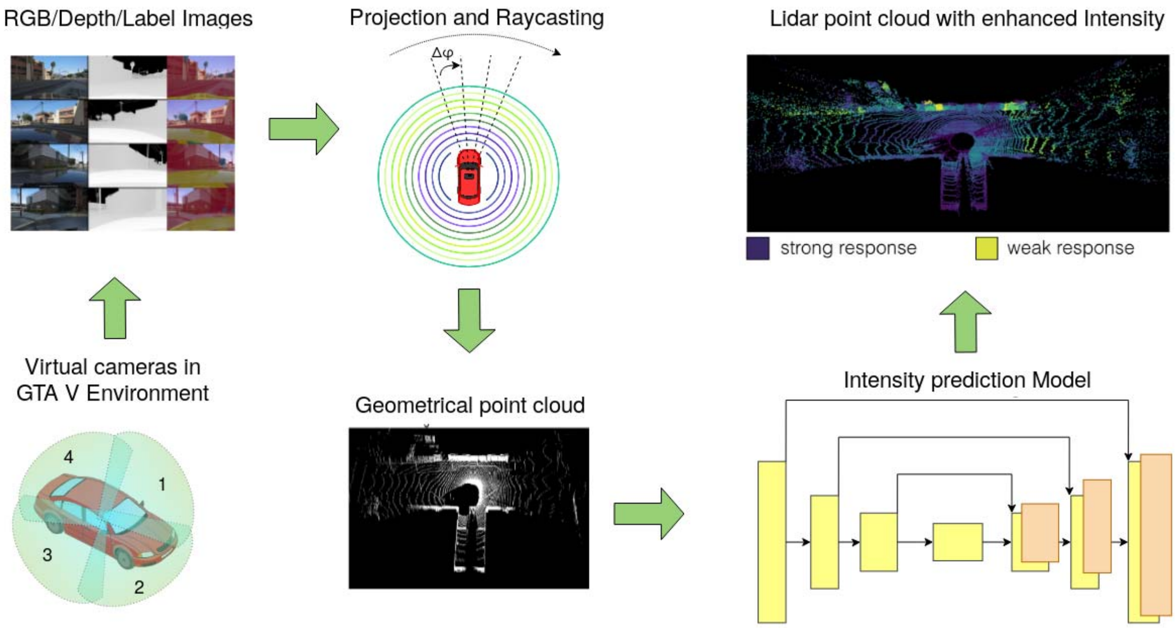
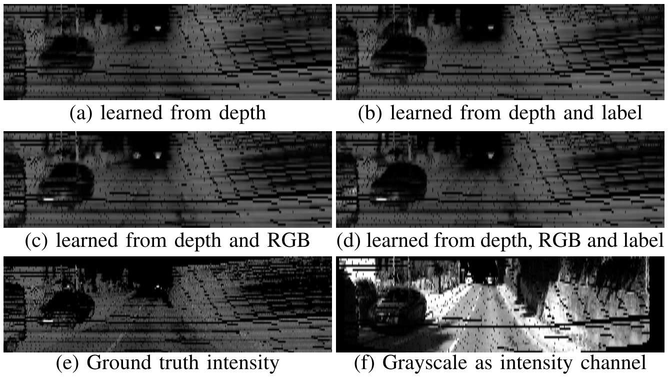

RGB/Depth/Label Images
Projection and Raycasting
Δφ
Lidar point cloud with enhanced Intensity
strong response
weak response
Virtual cameras in GTA V Environment
4
1
3
2
Geometrical point cloud
Intensity prediction Model

Fig. 6. Pipeline overview: At first, we collect depth, RGB, and label images in 360° view by placing four cameras to the desired sensor position and from depth information create dense point cloud with RGB and label channels. With ray-casting, we choose points corresponding to lidar parameters and estimate intensity by the deep convolutional network from depth, RGB and label input grids. For intensity prediction we used Unet architecture [10].


(a) learned from depth (b) learned from depth and label

(c) learned from depth and RGB (d) learned from depth, RGB and label

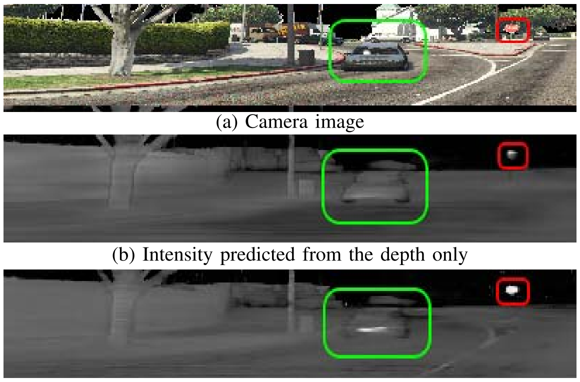
(e) Ground truth intensity (f) Grayscale as intensity channel

Fig. 7. Comparison of lidar intensities on the cars in the SemanticKitti dataset - These figures represent our intensity prediction according to used modalities in the same setting. Without RGB modality (a), (b), there is no high response from the license plate, see the real intensity in (e) for comparison. Whereas (c) and (d) successfully predict the high intensity of the received beam. Substitution grayscale value for intensity failed entirely due to light conditions, as can be seen in (f).

TABLE I
DISTRIBUTION OF HYBRID LOSS CLASSIFICATION BINS

Bin	1	2	3	4	5
Min value	0.0	0.079	0.163	0.225	0.264
Bin	6	7	8	9	10
Min value	0.289	0.310	0.334	0.368	0.546

For the optimization task, we experimented with different setups and finally used Adam algorithm [31] with a learning rate 0.003 and weight decay 0.001 with both models. Training set is divided to 7500 training and 2500 validation frames.


(a) Camera image

(b) Intensity predicted from the depth only

(c) Intensity predicted from depth and RGB

Fig. 8. Example of GTA scene with simulated lidar intensity - on the camera RGB image (a) is a car (green mark) and a traffic sign (red) mark. Predicted intensity from depth (b) and depth + RGB (c) showed different values on objects of interest, car's license plate, and traffic sign, where we expect greater values of intensity. Adding RGB modality helps to recognize licence plate (c) and more realistic values on the sign. RGB also differentiates lane marking.

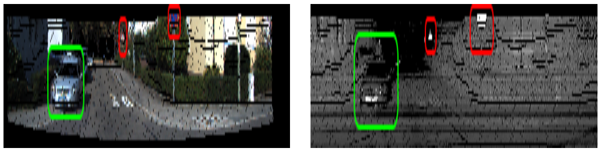

(a) SemanticKitti scene - camera image (b) SemanticKitti scene - real intensity

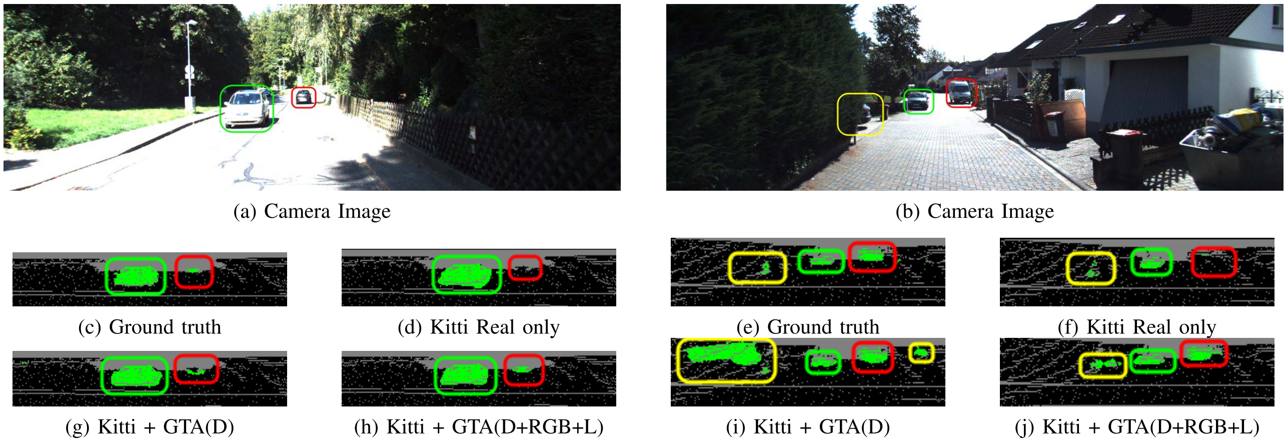
Fig. 9. Comparison of real scene intensity - Generating intensity across different domains keep systematic failures consistent, see Figure 8 for comparision. Traffic signs and license plates generate high signal feedback, while rest of the scene remains uniform. Therefore we can assume preservation of intensity characteristics.

Experiments reveal L2 - loss on the D+RGB+L input combination achieves the lowest MSE error on the testing data, see Table II for details. Examples of predicted intensities are provided in Figure 7. The intensities are projected into the camera frame for better readability. The images demonstrate that using the RGB information allows predicting stronger responses on license plates and traffic signs

TABLE II

MSE Error on Intensity Prediction - Comparison of Different Variations of Modalities and Loss Functions for Intensity Prediction, All Numbers in Percentage

Architecture and Loss function	D	D+L	D+RGB	D+RGB+L	D+RGB + Kitti labels [9] (D)
SqueezeSegV2 + Hybrid loss	1.11 [9]	1.11	1.02	1.023	0.98
SqueezeSegV2 + L2 loss	0.745	0.744	0.692	0.693	0.677
Unet + L2 loss	0.644	0.671	0.623	**0.621**	0.638



(a) Camera Image

(b) Camera Image

(c) Ground truth

(d) Kitti Real only

(e) Ground truth

(f) Kitti Real only

(g) Kitti + GTA(D)

(h) Kitti + GTA(D+RGB+L)

(i) Kitti + GTA(D)

(j) Kitti + GTA(D+RGB+L)

Fig. 10. Example of segmentation of distant and covered car - Adding GTA data increase segmentation performance mainly on cars in greater ranges, see the red markings on (a) and (b). There is a cropped detail on ground truth in (c). As you can see on (d), training only on the part of this dataset is not sufficient for greater range detection, there are probably not many distant cars in the dataset. However, with more training examples gathered from synthetic data, which may include similar scenes, we are able to segment it in (h),(i),(j).

while using only the depth modality yields limited results. Both proposed architectures significantly outperform a simple method, such as the intensity estimated as grayscale values (see Figure 7(f)).

We are particularly interested in objects with high reflectivity, namely car license plates and traffic signs as they consistently show high lidar intensity and are valuable for scene interpretation and car detection. Intensity prediction on synthetic data can be seen in Figures 8, where adding RGB modality to the learning and inference showed to be superior in distinguishing these objects. With color information in model prediction, it is also possible to differentiate lane markings on the street. This can be especially valuable in segmenting other instances, that can be used for navigation in the scene. Predicting from D+RGB looks also qualitatively more realistic and closer to real lidar intensity, see Figure 7.

B. Segmentation Accuracy Improvement

This experiment demonstrates that extending the costly real training data by easier accessible simulated point clouds improves the segmentation accuracy. Input to the segmentation network is a 2D image-like grid with channels containing depth, lidar intensity, and pixel mask, which serves as an indicator of a valid return of the ray. Some rays do not return in real lidar, and some exceed the maximum distance of sensor measurement in the GTA simulation.

The architecture of the segmentation network is SqueezeSeg with the CRF module. As a loss function, we used Focal loss [32] which happened to bring better results in training as opposed to the standard Cross entropy loss function. Focal loss is described in Equation (6). The value of parameter γ is set to 2.

$$FL(p_t) = -(1 - p_t)^{\gamma} \log p_t \tag{6}$$

The output contains pixel-level semantic labels. We compare several segmentation networks trained with different combinations of training datasets. First, we evaluate networks trained on synthetic data: GTA without intensity, GTA(D) and GTA(D+RGB+L). Second, we train the prediction model on 1k real frames from SemanticKitti (K). Last, we add 40k synthetic frames to the real ones, K + GTA(D), K + GTA(D+RGB+L).

We stick to the standard evaluation metric used in autonomous driving research [33] – Intersection-over-Union (7) – and evaluate segmentation performance on the car category.

$$IoU = \frac{TP}{TP + FP + FN}, \tag{7}$$

where TP denotes true positive points of a certain class, FP denotes false positives points and FN false negatives points of the class. The results are shown in Table III.

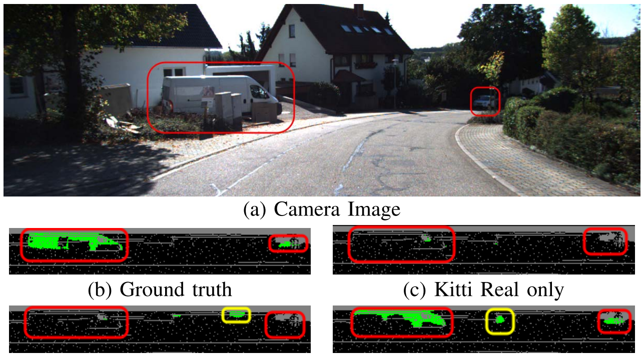
Adding artificial GTA data with generated lidar intensity improved the performance of the segmentor, especially for vehicles in a greater distance. Adding RGB and Label modality to intensity prediction proved to be superior to the baseline – using only depth for the intensity prediction. It yields better performance with both learning from synthetic data only and adding synthetic together with real data.

An example of a boost in segmentation can be seen in Figure 10, where the RGB modality improves the

TABLE III

EVALUATION OF DIFFERENT MODELED INTENSITIES ON SEGMENTATION ON TESTING SPLIT. ALL NUMBERS IN PERCENTAGE

Semantic Segmentation performance on our SemanticKitti split			
Training set	Real data	Synthetic data	IoU
GTA w/o intensity [34]	0	40k	23.83
GTA(D) [9]	0	40k	31.38
GTA(D+RGB+L)	0	40k	**33.04**
K [9]	1k	40k	75.16
K + GTA(D)	1k	40k	78.79
K + GTA(D+RGB+L)	1k	40k	**79.76**


(a) Camera Image

(b) Ground truth

(c) Kitti Real only

(d) Kitti + GTA(D)

(e) Kitti + GTA(D+RGB+L)

Fig. 11. Segmentation of a covered van - In this situation, we see a van, which is parked behind electric panel and its wheels are covered by bush and carton (a). Detail of ground truth is shown in (b). Segmentation trained solely on real data failed to detect van in this setup (c), together with GTA intensity, learned from the depth only (d). On the other hand, the addition of RGB modality benefits in segmentation of van and also correctly detect distant car.

distinction between a vehicle that is covered by an object and the object itself. The van in Figure 11 is not segmented if using depth only. Red marking means improved detection with enhanced data, yellow means false detection and green shows positive detection.

There is a large disproportion in *IoU* performance between using GTA data only compared to using real date which implies a significant domain gap between the two worlds. Our intensity predictor improves the results, however the domain gap between simulated and real lidar scans still dominates.

V. CONCLUSION

We proposed a new way of modeling lidar intensity from scene geometry, RGB images and generated labels. It has been shown that adding proposed synthetic lidar point clouds with enhanced intensity to learning improves segmentation results on the real lidar dataset. Predicted intensity based on RGB and label had an increase in segmentation performance over depth-based intensity. We have also shown that new modalities and masked L2-loss increase the accuracy of intensity prediction. All results were evaluated on the real data only.

Simulation interface and the synthetic training set consisting of panoramic RGB images and lidar point clouds have been made publicly available. There is still an insufficient domain adaptation between real-world and GTA simulation, mainly in geometrical properties and ray dropout.

Future work will focus on showing the results in challenging weather and visibility conditions such as fog, rain, and shallow puddles. We will also address problems of color and geometry domain shift as it will improve our intensity prediction model.

ACKNOWLEDGMENT

The authors also thank colleagues from Valeo Prague Research and Development centre for valuable discussions.

REFERENCES

[1] ReinventingParking. (2013). *Cars are Parked 95% of the Time*. [Online]. Available: https://www.reinventingparking.org/2013/02/cars-are-parked-95-of-time-l ets-check.html

[2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. 1st Annu. Conf. Robot Learn.*, 2017, pp. 1–16.

[3] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Proc. 11th Conf. Field Service Robot.* Zürich, Switzerland: Springer, 2017.

[4] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 1–8.

[5] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[6] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[7] X. Yue, B. Wu, S. A. Seshia, K. Keutzer, and A. L. Sangiovanni-Vincentelli, "A lidar point cloud generator: From a virtual world to autonomous driving," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2018, pp. 458–464.

[8] A. Kashani, M. Olsen, C. Parrish, and N. Wilson, "A review of LIDAR radiometric processing: From ad hoc intensity correction to rigorous radiometric calibration," *Sensors*, vol. 15, no. 11, pp. 28099–28128, Nov. 2015.

[9] B. Wu, X. Zhou, S. Zhao, X. Yue, and K. Keutzer, "SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 4376–4382.

[10] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241.

[11] J. Behley *et al.*, "SemanticKITTI: A dataset for semantic scene understanding of LiDAR sequences," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9297–9307.

[12] H. Caesar *et al.*, "NuScenes: A multimodal dataset for autonomous driving," 2019, *arXiv:1903.11027*. [Online]. Available: http://arxiv.org/abs/1903.11027

[13] L. Fridman *et al.*, "MIT advanced vehicle technology study: Large-scale naturalistic driving study of driver behavior and interaction with automation," *IEEE Access*, vol. 7, pp. 102021–102038, 2019.

[14] Y. Wang, T. Shi, P. Yun, L. Tai, and M. Liu, "Pointseg: Real-time semantic segmentation based on 3D LIDAR point cloud," 2018, *arXiv:1807.06288*. [Online]. Available: https://arxiv.org/abs/1807.06288

[15] Z. Wang, H. Fu, L. Wang, L. Xiao, and B. Dai, "SCNet: Subdivision coding network for object detection based on 3D point cloud," *IEEE Access*, vol. 7, pp. 120449–120462, 2019.

[16] B. Yang, M. Liang, and R. Urtasun, "Hdnet: Exploiting HD maps for 3D object detection," in *Proc. Mach. Learn. Res. (PMLR), Conf. Robot Learn. (CoRL)*, Zürich, Switzerland, vol. 87, Oct. 2018, pp. 146–155.

[17] B. Yang, W. Luo, and R. Urtasun, "PIXOR: Real-time 3D object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7652–7660.

[18] W. Wang and J. Shen, "Deep visual attention prediction," *IEEE Trans. Image Process.*, vol. 27, no. 5, pp. 2368–2378, May 2018.

[19] J. Shen, X. Tang, X. Dong, and L. Shao, "Visual object tracking by hierarchical attention siamese network," *IEEE Trans. Cybern.*, vol. 50, no. 7, pp. 3068–3080, Jul. 2020.

[20] W. Wang, J. Shen, and H. Ling, "A deep network solution for attention and aesthetics aware photo cropping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1531–1544, Jul. 2019.

[21] J. Fang *et al.*, "Simulating LIDAR point cloud for autonomous driving using real-world scenes and traffic flows," 2018, *arXiv:1811.07112*. [Online]. Available: https://arxiv.org/abs/1811.07112

[22] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "BlenSor: Blender sensor simulation toolbox," in *Proc. 7th Int. Symp. Vis. Comput.* vol. 2. Berlin, Germany: Springer-Verlag, 2011, pp. 199–208.

[23] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. CVPR*, Jun. 2016, pp. 4340–4349.

[24] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May/Jun. 2017, pp. 1–8.

[25] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *Proc. Eur. Conf. Comput. Vis. (ECCV)* (Lecture Notes in Computer Science), vol. 9906, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Amsterdam, The Netherlands: Springer, 2016, pp. 102–118.

[26] M. R. Zofka, M. Essinger, T. Fleck, R. Kohlhaas, and J. M. Zollner, "The sleepwalker framework: Verification and validation of autonomous vehicles by mixed reality LiDAR stimulation," in *Proc. IEEE Int. Conf. Simul., Model., Program. Auto. Robots (SIMPAR)*, May 2018, pp. 151–157.

[27] M. Hadj-Bachir and P. De Souza, "LIDAR sensor simulation in adverse weather condition for driving assistance development," working paper or preprint, Jan. 2019. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01998668/document

[28] K. Elmadawi, M. Abdelrazek, M. Elsobky, H. M. Eraqi, and M. Zahran, "End-to-end sensor modeling for LiDAR point cloud," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 1619–1624.

[29] M. F. Holder, P. Rosenberger, F. Bert, and H. Winner, "Data-driven derivation of requirements for a lidar sensor model," in *Proc. Grazer Symp. Virtuelles Fahrzeug*. May 2018.

[30] M. Računský, "3D map estimation from a single RGB image," M.S. thesis, Dept. Cybern., Prague, Comput. Inf. Centre, Czech Tech. Univ., Prague, Czech Republic, 2018.

[31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.

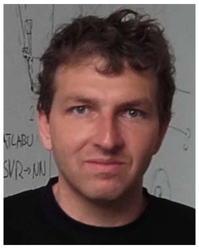
[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[33] W. Zhou, J. S. Berrio, S. Worrall, and E. Nebot, "Automated evaluation of semantic segmentation robustness for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1951–1963, May 2020.

[34] B. Wu, A. Wan, X. Yue, and K. Keutzer, "SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 1887–1893.

Patrik Vacek received the M.Sc. (Ing.) degree in mechatronics from Czech Technical University (CTU), Prague, in 2018. He is currently pursuing the Ph.D. degree with the Department of Cybernetics, Faculty of Electrical Engineering, CTU, under the supervision of Tomás Svoboda and Karel Zimmermann, where he works as a Research Assistant. He also collaborate with the Valeo Research and Development Centre, Prague. His main research interests include machine learning and deep learning techniques in the field of autonomous driving and robotics, mainly using these methods for interpretation of objects in driving scenes and simulation of sensor systems.

Otakar Jašek received the M.Sc.Ing. degree in computer vision from Czech Technical University, Prague, in 2018. He is currently pursuing the Ph.D. degree with the Department of Cybernetics, Czech Technical University (CTU), Prague, under the supervision of Karel Zimmermann and Tomás Svoboda. His main research interests include processing point clouds by means of machine learning and deep learning, especially for the purposes of autonomous driving. He is currently working for Avast Software.



Karel Zimmermann (Member, IEEE) received the Ph.D. degree in cybernetics in 2008. He is currently an Associate Professor with Czech Technical University, Prague. He worked as a Postdoctoral Researcher with Katholieke Universiteit Leuven (2008–2009) in the group of prof Luc van Gool. His current H-index is 14 (google-scholar) and about 1000 citations. He serves as a reviewer for major impacted journals such as TPAMI or IJCV and rank-A* conferences such as CVPR, ICCV, AAAI, and IROS. He was leading the team who won DARPA Subterranean Challenge Tunnel Circuit 2019 among non-sponsored teams. He received the best lecturer award in 2018, the best reviewer award at CVPR 2011, and the best Ph.D. work award in 2008. His journal article has been selected among 14 best research works representing Czech Technical University in the government evaluation process (RIV). Since 2010, he has been the Chair of Antonin Svoboda Award. He was also with the Technological Education Institute of Crete (2001), with the Technical University of Delft (2002), with the University of Surrey (2006). His current research interests include learnable methods for robotics.

Tomáš Svoboda (Member, IEEE) received the Ph.D. degree in artificial intelligence and biocybernetics from the Czech Technical University (CTU), Prague, Czech Republic, in 2000. He spent three post-doctoral years with the Computer Vision Group, ETH Zurich, Switzerland. He is currently an Associate Professor and the Chair of the Department of Cybernetics, CTU, and the Director of EECS study program. He is also on the Board of the Open Informatics programme. He has published articles on multicamera systems, omnidirectional cameras, image-based retrieval, learnable detection methods, and USAR robotics. His recent research interests include multimodal perception for autonomous systems, object detection, and related applications in the automotive industry.