

# Fog Simulation on Real LiDAR Point Clouds for 3D Object Detection in Adverse Weather

Martin Hahner<sup>1</sup>

mhahner@vision.ee.ethz.ch

Christos Sakaridis<sup>1</sup>

csakarid@vision.ee.ethz.ch

Dengxin Dai<sup>1,2</sup>

ddai@mpi-inf.mpg.de

Luc Van Gool<sup>1,3</sup>

luc.vangool@kuleuven.be

<sup>1</sup> ETH Zürich<sup>2</sup> MPI for Informatics<sup>3</sup> KU Leuven

## Abstract

This work addresses the challenging task of LiDAR-based 3D object detection in foggy weather. Collecting and annotating data in such a scenario is very time, labor and cost intensive. In this paper, we tackle this problem by simulating physically accurate fog into clear-weather scenes, so that the abundant existing real datasets captured in clear weather can be repurposed for our task. Our contributions are twofold: 1) We develop a physically valid fog simulation method that is applicable to any LiDAR dataset. This unleashes the acquisition of large-scale foggy training data at no extra cost. These partially synthetic data can be used to improve the robustness of several perception methods, such as 3D object detection and tracking or simultaneous localization and mapping, on real foggy data. 2) Through extensive experiments with several state-of-the-art detection approaches, we show that our fog simulation can be leveraged to significantly improve the performance for 3D object detection in the presence of fog. Thus, we are the first to provide strong 3D object detection baselines on the Seeing Through Fog dataset. Our code is available at [www.trace.ethz.ch/lidar\\_fog\\_simulation](http://www.trace.ethz.ch/lidar_fog_simulation).

## 1. Introduction

Light detection and ranging (LiDAR) is crucial for the implementation of *safe* autonomous cars, because LiDAR measures the precise distance of objects from the sensor, which cameras cannot measure directly. Thus, LiDAR has found its way into many applications, including detection [23, 34], tracking [7, 50], localization [25, 8], and mapping [48, 15]. Despite the benefit of measuring exact depth information, LiDAR has a significant drawback. The light pulses that LiDAR sensors emit in the invisible near infrared (NIR) spectrum (typically at 850 and 903 to 905 nm wavelength [4]) do *not* penetrate water particles, as opposed to automotive radars. This means as soon as there are water particles in the form of fog in the air, light pulses emitted by the sensor will undergo backscattering and attenuation.

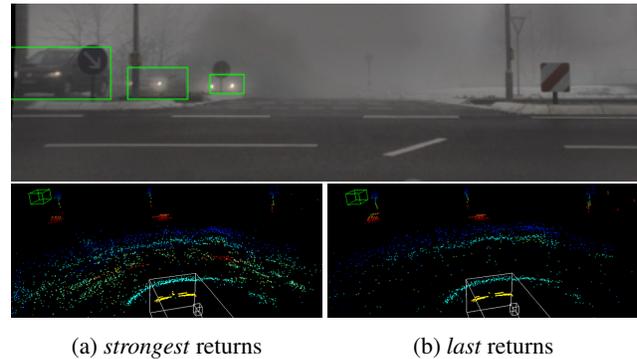


Figure 1: LiDAR returns caused by fog in the (top) scene. (a) shows the *strongest* returns and (b) the *last* returns, color coded by the LiDAR channel. The returns of the ground are removed for better visibility of the points introduced by fog. Best viewed in color (red  $\hat{=}$  low, cyan  $\hat{=}$  high, 3D bounding box annotation in green, ego vehicle dimensions in gray).

Attenuation reduces the received signal power that corresponds to the range of the solid object in the line of sight which should be measured, while backscattering creates a spurious peak in the received signal power at an incorrect range. As a result, the acquired LiDAR point cloud will contain some spurious returns whenever there is fog present at the time of capture. This poses a big challenge for most outdoor applications, as they typically require robust performance under all weather conditions.

In recent years, several LiDAR datasets for 3D object detection [10, 3, 19, 5, 39, 27, 11, 16] have been presented. Although many of them contain diverse driving scenarios, none of them allows an evaluation on different kinds of adverse weather. Only recently, the Canadian Adverse Driving Conditions (CADC) Dataset [28] and the Seeing Through Fog (STF) Dataset [2] address the need for such an evaluation. While CADC focuses on snowfall, STF is targeted towards *evaluation* under fog, rain and snow. Consequently, there is still not a large quantity of LiDAR foggy data available that could be used to *train* deep neural networks.

The reason for this is obvious: collecting and annotating large-scale datasets per se is time, labor and cost intensive, let alone when done for adverse weather conditions.

This is exactly the shortfall that our work addresses. In Sec. 3, we propose a physically-based fog simulation that converts real clear-weather LiDAR point clouds into *foggy* counterparts. In particular, we use the standard linear system [30] that models the transmission of LiDAR pulses. We distinguish between the cases of clear weather and fog with respect to the impulse response of this system and establish a formal connection between the received response under fog and the respective response in clear weather. This connection enables a straightforward transformation of the range and intensity of each original clear-weather point, so that the new range and intensity correspond to the measurement that would have been made if fog was present in the scene. We then show in Sec. 4 that several state-of-the-art 3D object detection pipelines can be trained on our partially synthetic data to get improved robustness on *real* foggy data. This scheme has already been applied on images for semantic segmentation [32, 31, 13] and we show that it is also successful for LiDAR data and 3D object detection.

For our experiments, we simulate fog on the clear-weather training set of STF [2] and evaluate on their real foggy test set. Fig. 1 shows an example scene from the STF dense fog test set, where the noise introduced by fog is clearly visible in the LiDAR data. The authors of STF [2] used a Velodyne HDL-64E as their main LiDAR sensor. This sensor comes with 64 *channels* and a so-called *dual mode*. In this mode, it can measure not only the *strongest*, but also the *last* return received for each individual emitted light pulse. Even though the *last* signal contains less severe noise, fog still causes a significant amount of spurious returns. Therefore, even in this *dual mode*, the sensor cannot fully “see through fog”.

Fig. 2 shows an interesting characteristic of the noise introduced by fog, namely that it is not uniformly distributed around the sensor. On the contrary, the presence of noise depends on whether there is a target in the line of sight below a certain range from the sensor. If there is a solid object at a moderate range, there are few, if any, spurious returns from the respective pulses. On the other hand, if there is no target in the line of sight below a certain range, there are a lot of spurious returns that are caused by fog. This becomes apparent in the example of Fig. 2, where on the left side of the road there is a hill and on the right side there is open space behind the guardrail. Only in the latter case does the noise caused by fog appear in the measurement. This behavior is explained with our theoretical formulation in Sec. 3.

As a side note, similar sensor noise can also be caused by exhaust smoke, but if the future of transportation goes electric, at least this problem may *vanish into thin air*.

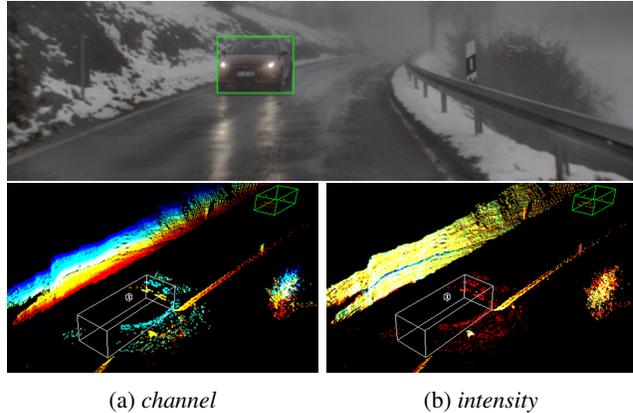


Figure 2: LiDAR returns caused by fog in the (top) scene. Color coded by the LiDAR *channel* in (a) and by the *intensity* in (b). The returns of the ground are removed for better visibility of the points introduced by fog. Best viewed in color, same color coding as in Fig. 1 applies.

## 2. Related Work

### 2.1. Effects of Adverse Weather on LiDAR

Some of the early works include Isaac *et al.* [20]. In 2001, they investigate the influences of fog and haze on optical wireless communications in the NIR spectrum. Then, in 2011, Rasshofer *et al.* [30] investigate the influences of weather phenomena on automotive LiDAR systems. In recent years, adverse weather conditions got a lot more attention and there are many other works worth mentioning that look into the degradation of LiDAR data in different adverse weather conditions [45, 9, 21, 17, 14, 44, 24, 22]. Very recently, in 2020, the authors of LIBRE [4] test several LiDAR sensors in a weather chamber under rain and fog. Thereby they provide great and valuable insights on the robustness of individual sensors of this time on challenging weather conditions.

### 2.2. Adverse Weather and LiDAR Simulation

In the automotive context, artificial fog simulation is so far mostly limited to image based methods. Sakaridis *et al.* [32] e.g. create a foggy version of Cityscapes [6], a dataset for Semantic Segmentation, and Hahner *et al.* [13] a foggy version of the purely synthetic dataset Synscapes [46] by leveraging the depth information given in the original datasets. Sakaridis *et al.* also released ACDC [33], a dataset providing semantic pixel-level annotations for 19 Cityscapes classes in adverse conditions. Only recently, Bijelic *et al.* [2] propose a first order approximation to simulate fog in an automotive LiDAR setting. However, their simulation only aims at reproducing measurements they carried out in a 30m long fog chamber.

Goodin *et al.* [12] develop a model to quantify the performance degradation of LIDAR in rain and incorporate their model into a simulation which they use to test an advanced driver assist system (ADAS). Michaud *et al.* [26] and Tremblay *et al.* [41] propose a method to render rain on images to evaluate and improve the robustness on rainy images.

### 2.3. 3D Object Detection

After the release of many LiDAR datasets [10, 3, 19, 5, 39, 27, 11, 16, 28, 2] over the past few years, 3D object detection is receiving increasing attention in the race towards autonomous driving. While there exist camera based methods such as Simonelli *et al.* [38] and gated camera methods such as Gated3D [18], the top ranks across all dataset leaderboards are typically sorted out among LiDAR based methods.

Seminal work in LiDAR based 3D object detection methods include PointNet [29] and VoxelNet [49]. PointNet [29] is a neural network that directly processes point clouds without quantizing the 3D space into voxels beforehand. Most notably, the network architecture is robust to input perturbation, so the order in which the points get fed into the network do not effect its performance. VoxelNet [49] builds on the idea to quantize the 3D space into equally sized voxels and then leverages PointNet-like layers to process each and every voxel. Due to it's computational intensive 3D convolutions, it is however a rather heavy architecture.

That's where PointPillars [23] comes in: it gets rid of the quantization in the height domain and processes the point cloud instead in a 3D pillar grid. PointPillars [23] is based on the SECOND [47] codebase, but due to the novel pillar idea, it can fall back to much faster 2D convolutions and achieves very competitive results at a much faster speed. Its successor PointPainting [43] further leverages image segmentation results and "paints" the points with a pseudo class label before processing them with the PointPillars [23] architecture.

Shi *et al.* achieved several recent milestones in 3D object detection. PointRCNN [36] is a two-stage architecture, where the first stage generates 3D bounding box proposals from a point cloud in a bottom-up manner and the second stage refines these 3D bounding box proposals in a canonical fashion. Part-A<sup>2</sup> [37] is part-aware in a sense that the network takes into account which part of the object a point belongs to. It leverages these intra-object part locations and can thereby achieve higher results. PV-RCNN [34] and it's successor PV-RCNN++ [35] are the latest of their works that simultaneously process (coarse) voxels and the raw points of the point cloud at the same time.

## 3. Fog Simulation on Real LiDAR Point Clouds

To simulate the effect of fog on real-world LiDAR point clouds that have been recorded in clear weather, we need to resort to the optical system model that underlies the function of the transmitter and receiver of the LiDAR sensor. In particular, we examine a single measurement/point, model the full signal of received power as a function of the range and recover its exact form corresponding to the original clear-weather measurement. This allows us to operate in the signal domain and implement the transformation from clear weather to fog simply by modifying the part of the impulse response that pertains to the optical channel (i.e. the atmosphere). In the remainder of this section, we first provide the background on the LiDAR sensor's optical system and then present our fog simulation algorithm based on this system.

### 3.1. Background on the LiDAR Optical Model

Rasshofer *et al.* [30] introduced a simple linear system model to describe the received signal power at a LiDAR's receiver, which is valid for non-elastic scattering. In particular, the range-dependent received signal power  $P_R$  is modeled as the time-wise convolution between the time-dependent transmitted signal power  $P_T$  and the time-dependent impulse response  $H$  of the environment:

$$P_R(R) = C_A \int_0^{2R/c} P_T(t) H\left(R - \frac{ct}{2}\right) dt. \quad (1)$$

$c$  is the speed of light and  $C_A$  is a system constant which is independent of time and range. For our fog simulation, as we explain in Sec. 3.2,  $C_A$  can be factored out.

We proceed with the description and modeling of the remaining terms in (1). The time signature of the transmit pulse can be modeled for automotive LiDAR sensors [30] by a  $\sin^2$  function:

$$P_T(t) = \begin{cases} P_0 \sin^2\left(\frac{\pi}{2\tau_H}t\right), & 0 \leq t \leq 2\tau_H \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Where  $P_0$  denotes the pulse's peak power and  $\tau_H$  the half-power pulse width. Typical values for  $\tau_H$  lie between 10 and 20 ns [30]. In (2), the time origin is set to the start of the pulse, so in case a LiDAR sensor does not report the range associated with a rising edge, but the maximum of the corresponding peak in the return signal, we can perform the required correction later in our pipeline. Since it is common to report the rising edge in embedded signal processing, we keep this convention throughout all of our equations and show where one can perform such a correction later on.

The spatial impulse response function  $H$  of the environment can be modeled as the product of the individual impulse responses of the optical channel,  $H_C$ , and the target,  $H_T$ :

$$H(R) = H_C(R) H_T(R). \quad (3)$$

The impulse response of the optical channel  $H_C$  is

$$H_C(R) = \frac{T^2(R)}{R^2} \xi(R), \quad (4)$$

where  $T(R)$  stands for the total one-way transmission loss and  $\xi(R)$  denotes the crossover function defining the ratio of the area illuminated by the transmitter and the part of it observed by the receiver, as illustrated in Fig. 3. Because generally the full details of the optical configuration in commercial LiDAR sensors are not publicly available (i.e. the precise values of  $R_1$  and  $R_2$  are unknown),  $\xi(R)$  in our case is a piece-wise linear approximation defined as

$$\xi(R) = \begin{cases} 0, & R \leq R_1 \\ \frac{R}{R_2 - R_1} - \frac{R_1}{R_2 - R_1}, & R_1 < R < R_2 \\ 1, & R_2 \leq R. \end{cases} \quad (5)$$

The total one-way transmission loss  $T(R)$  is defined as

$$T(R) = \exp\left(-\int_{r=0}^R \alpha(r) dr\right), \quad (6)$$

where  $\alpha(r)$  denotes the spatially varying attenuation coefficient. In our simulation, we make the assumption of a *homogeneous* optical medium, i.e.  $\alpha(r) = \alpha$ . As a result, (6) yields

$$T(R) = \exp(-\alpha R). \quad (7)$$

The attenuation coefficient  $\alpha$  depends on the weather at the time of measurement and increases as visibility range decreases. Therefore, for the same 3D scene, the impulse response of the optical channel  $H_C$  varies with visibility.

The last term of the optical system (1) that remains to be modeled is the impulse response of the target,  $H_T$ . However, we need to distinguish cases for  $H_T$  according to the weather condition, as the composition of the target for the same 3D scene is different in fog than in clear weather. We make the contribution of constructing a direct relation between the response  $P_R$  in clear weather and in fog for the same 3D scene and this relation enables us to simulate fog on real clear-weather LiDAR measurements.

### 3.2. Fog Simulation for LiDAR

We now particularize the optical model of Sec. 3.1 for the individual cases of clear weather and fog in terms of the impulse response terms,  $H_C$  and  $H_T$ .

In clear weather, the attenuation coefficient  $\alpha$  is 0, so

$$H_C(R) = \frac{\xi(R)}{R^2}. \quad (8)$$

Moreover, the target in clear weather is comprised only of the solid object on which the LiDAR pulse is reflected. This type of target is called a *hard* target [30].

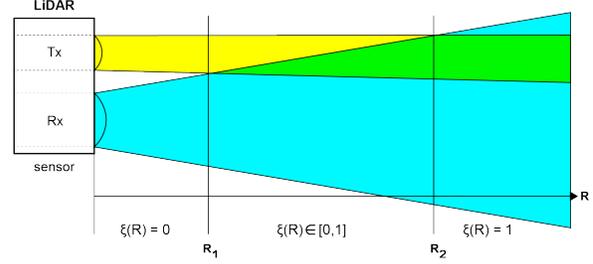


Figure 3: Sketch of a LiDAR sensor where the transmitter Tx and the receiver Rx do not have coaxial optics, but have parallel axes. This is called a bistatic beam configuration. Figure adjusted from [30].

The impulse response  $H_T$  of a hard target at range  $R_0$  is a Dirac delta function of the form

$$H_T(R) = H_T^{\text{hard}}(R) = \beta_0 \delta(R - R_0), \quad (9)$$

where  $\beta_0$  denotes the differential reflectivity of the target. If we only consider diffuse reflections (Lambertian surfaces),  $\beta_0$  is given by

$$\beta_0 = \frac{\Gamma}{\pi}, \quad 0 < \Gamma \leq 1. \quad (10)$$

Consequently, plugging in (8) and (9) into (3), in clear weather the total impulse response function  $H(R)$  can be expressed as

$$H(R) = \frac{\xi(R_0)}{R_0^2} \beta_0 \delta(R - R_0), \quad (11)$$

where we have used the property  $f(x)\delta(x - x_0) = f(x_0)\delta(x - x_0)$ . Since in practice  $R_2$  is less than two meters [42], we can safely assume  $R_2 \ll R_0$ , so  $\xi(R_0) = 1$ . Thus, starting from (1) and given the range measurement  $R_0$  for the original clear-weather LiDAR point, we compute the received signal power in closed form as

$$\begin{aligned} P_{R,\text{clear}}(R) &= C_A \int_0^{2\tau_H} P_0 \sin^2\left(\frac{\pi}{2\tau_H} t\right) \frac{1}{R_0^2} \beta_0 \delta\left(R - \frac{ct}{2} - R_0\right) dt \\ &= \begin{cases} \frac{C_A P_0 \beta_0}{R_0^2} \sin^2\left(\frac{\pi(R-R_0)}{c\tau_H}\right), & R_0 \leq R \leq R_0 + c\tau_H \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (12)$$

The received signal power attains its maximum value at  $R_0 + \frac{c\tau_H}{2}$ . So as we mentioned in Sec. 3.1, here one can simply shift the response  $P_{R,\text{clear}}(R)$  by  $-\frac{c\tau_H}{2}$  if necessary.

We now establish the transformation of  $P_{R,\text{clear}}(R)$  to  $P_{R,\text{fog}}(R)$  under fog. While the same hard target still exists since the 3D scene is the same, there is now an additional contribution from fog—which constitutes a *soft* target [30] that provides distributed scattering—to the impulse response  $H_T$ .

The impulse response of this soft fog target,  $H_T^{\text{soft}}$ , is a Heaviside function of the form

$$H_T^{\text{soft}}(R) = \beta U(R_0 - R), \quad (13)$$

where  $\beta$  denotes the backscattering coefficient, which is constant under our homogeneity assumption, and  $U$  is the Heaviside function.

The co-existence of a hard target and a soft target can be modeled by taking the *superposition* of the respective impulse responses:

$$\begin{aligned} H_T(R) &= H_T^{\text{soft}}(R) + H_T^{\text{hard}}(R) \\ &= \beta U(R_0 - R) + \beta_0 \delta(R - R_0). \end{aligned} \quad (14)$$

Consequently, plugging in (7) into (4) and then (4) and (14) into (3), in fog the total impulse response function  $H(R)$  can be expressed as

$$\begin{aligned} H(R) &= \frac{\exp(-2\alpha R)\xi(R)}{R^2} \\ &\times (\beta U(R_0 - R) + \beta_0 \delta(R - R_0)). \end{aligned} \quad (15)$$

We observe that compared to clear weather, the spatial impulse response in fog is more involved, but it can still be decomposed into two terms, corresponding to the hard and the soft target respectively, leading to a respective decomposition of the received response as

$$P_{R,\text{fog}}(R) = P_{R,\text{fog}}^{\text{hard}}(R) + P_{R,\text{fog}}^{\text{soft}}(R). \quad (16)$$

Focusing on the hard target term, using (1) to calculate the corresponding term of the received response  $P_{R,\text{fog}}^{\text{hard}}$  and leveraging again the assumption that  $R_2 \ll R_0$ , we obtain

$$\begin{aligned} P_{R,\text{fog}}^{\text{hard}}(R) &= \\ &= C_A \frac{\exp(-2\alpha R_0)}{R_0^2} \int_0^{2\tau_H} P_0 \sin^2\left(\frac{\pi}{2\tau_H}t\right) \beta_0 \delta\left(R - \frac{ct}{2} - R_0\right) dt \\ &= \begin{cases} \frac{C_A P_0 \beta_0 \exp(-2\alpha R_0)}{R_0^2} \sin^2\left(\frac{\pi(R-R_0)}{c\tau_H}\right), & R_0 \leq R \leq R_0 + c\tau_H \\ 0 & \text{otherwise.} \end{cases} \\ &= \exp(-2\alpha R_0) P_{R,\text{clear}}(R). \end{aligned} \quad (17)$$

In other words, the hard target term of the response in fog is an attenuated version of the original clear-weather response  $P_{R,\text{clear}}$ . On the other hand, the soft target term is

$$\begin{aligned} P_{R,\text{fog}}^{\text{soft}}(R) &= C_A P_0 \beta \int_0^{2\tau_H} \sin^2\left(\frac{\pi}{2\tau_H}t\right) \times \\ &\times \frac{\exp\left(-2\alpha\left(R - \frac{ct}{2}\right)\right)}{\left(R - \frac{ct}{2}\right)^2} \xi\left(R - \frac{ct}{2}\right) U\left(R_0 - R + \frac{ct}{2}\right) dt \end{aligned} \quad (18)$$

and does not possess a closed-form expression, as the respective integral  $I(R, R_0, \alpha, \tau_H)$  on the right-hand side of (18) cannot be calculated analytically.

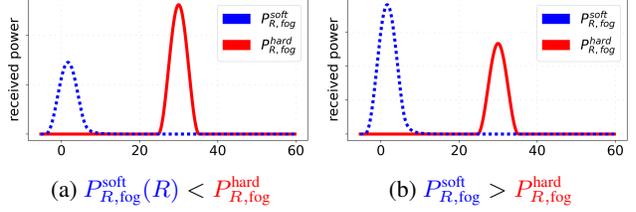


Figure 4: The two terms of the received signal power  $P_{R,\text{fog}}$  from a single LiDAR pulse, associated to the solid object that reflects the pulse ( $P_{R,\text{fog}}^{\text{hard}}$ ) and the soft fog target ( $P_{R,\text{fog}}^{\text{soft}}$ ), plotted across the range domain. While in (a) the fog is not thick enough to yield a return, in (b) it is thick enough to yield a return that overshadows the solid object at  $R_0 = 30\text{m}$ .

However, for given  $\tau_H$  and  $\alpha$ ,  $I(R, R_0, \alpha, \tau_H)$  can be computed numerically for fixed values of  $R$ . We use Simpson's  $\frac{1}{3}$  rule for numerical integration and provide indicative examples of the profile of  $P_{R,\text{fog}}^{\text{soft}}(R)$  in Fig. 4. Depending on the distance of the hard target from the sensor, the soft target term of the response may exhibit a larger maximum value than the hard target term, which implies that *the measured range changes due to the presence of fog* and becomes equal to the point of maximum of the soft-target term.

The formulation that we have developed affords a simple algorithm for fog simulation on clear-weather point clouds. The input parameters to the algorithm are  $\alpha$ ,  $\beta$ ,  $\beta_0$  and  $\tau_H$ . The main input of the algorithm is a clear-weather point cloud, where each point  $\mathbf{p} \in \mathbb{R}^3$  has a measured intensity  $i$ . We make the assumption that the intensity readings of the sensor are a linear function of the maxima of the received signal power  $P_{R,\text{clear}}$  corresponding to each measurement. The procedure for each point  $\mathbf{p}$  is given in Algorithm 1. Note, that we add some noise to the distance of  $P_{R,\text{fog}}^{\text{soft}}$  (line 14-15), otherwise all points introduced by  $P_{R,\text{fog}}^{\text{soft}}$  would lie precisely on a circle around the LiDAR sensor.

## 4. Results

### 4.1. Fog Simulation

A qualitative comparison between our fog simulation and the fog simulation in [2] can be found in Fig. 5. We can see that in contrast to the fog simulation in [2] where the response of soft target is only modeled heuristically, our fog simulation models  $P_{R,\text{fog}}^{\text{soft}}$  in a physically sound way. To highlight this difference, we specifically picked a clear weather scene with a similar layout to the real foggy scene depicted in Fig. 2. Only in our fog simulation (best visible in the bottom right visualization of Fig. 5), a similar half circle of fog noise gets simulated. In the Supplementary Materials we show a comparison with additional  $\alpha$  values.

---

**Algorithm 1** LiDAR fog simulation
 

---

```

1: procedure FOGGIFY( $\mathbf{p}, i, \alpha, \beta, \beta_0, \tau_H$ )
2:    $R_0 \leftarrow \|\mathbf{p}\|$ 
3:    $x, y, z \leftarrow \mathbf{p}$  ▷  $i = P_{R, \text{clear}}$ 
4:    $C_{AP_0} \leftarrow i \frac{R_0^2}{\beta_0}$  ▷ follows from Eq. (12)
5:    $i_{\text{hard}} \leftarrow i \times \exp(-2\alpha R_0)$  ▷ see Eq. (17)
6:   for  $R$  in  $(0, 0.1, \dots, R_0)$  do ▷ 10cm accuracy
7:      $I_R \leftarrow \text{SIMPSON}(I(R, R_0, \alpha, \tau_H))$  ▷ see Eq. (18)
8:   end for
9:    $i_{\text{tmp}} \leftarrow \max(I_R)$ 
10:   $R_{\text{tmp}} \leftarrow \arg \max(I_R)$ 
11:   $i_{\text{soft}} \leftarrow C_{AP_0} \beta \times i_{\text{tmp}}$  ▷ see again Eq. (18)
12:  if  $i_{\text{soft}} > i_{\text{hard}}$  then
13:     $s \leftarrow \frac{R_{\text{tmp}}}{R_0}$  ▷ scaling factor  $s$ 
14:     $p \leftarrow \text{RANDOM\_UNIFORM\_FLOAT}(-1, 1)$ 
15:     $n \leftarrow 2^p$  ▷ noise factor  $n \in (\frac{1}{2}, 2)$ 
16:     $x \leftarrow s \times n \times x$ 
17:     $y \leftarrow s \times n \times y$ 
18:     $z \leftarrow s \times n \times z$ 
19:     $i \leftarrow i_{\text{soft}}$ 
20:  else ▷ keep original location
21:     $i \leftarrow i_{\text{hard}}$  ▷ only modify intensity
22:  end if
23:  return  $x, y, z, i$ 
24: end procedure

```

---

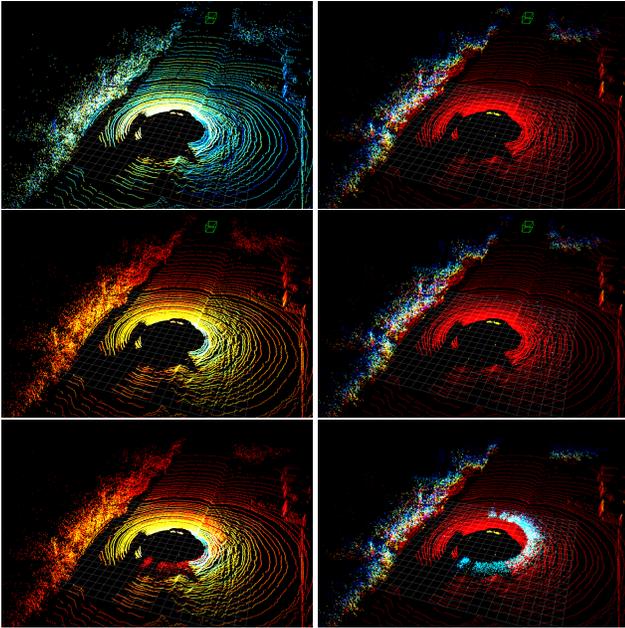


Figure 5: Comparison of our fog simulation (bottom) to the fog simulation in [2] (middle) with  $\alpha$  set to 0.06, which corresponds to a meteorological optical range (MOR)  $\approx 50\text{m}$ . In the left column, the point cloud is color coded by the *intensity* and in the right column it is color coded by the *height* ( $z$  value). The top row shows the original point cloud.

## 4.2. 3D Object Detection in Fog

Our experimental setup codebase is forked from OpenPCDet [40]. It comes with implementations of the 3D Object Detection methods PV-RCNN [34], PointRCNN [36], SECOND [47], Part-A<sup>2</sup> [37] and PointPillars [23]. For our experiments we train all of these methods from scratch for 80 epochs with their provided standard training policies on the STF [2] dataset. We also tried to fine-tune from KITTI [10] weights (which uses the same LiDAR sensor), but besides the networks converging faster, we did not see any benefit, so all the numbers you see in section 4.2 are trained from scratch on the STF [2] clear weather training set that consist of 3469 scenes. The STF [2] clear weather validation and testing set consists of 781 and 1847 scenes respectively. However, the main benefit of using STF [2] for our experiments is because it comes with test sets for different adverse weather conditions. In particular, it comes with a light fog test set of 946 scenes and a dense fog test set with 786 scenes. This allows us to test the effectiveness of our fog simulation pipeline on real foggy data.

Regarding our fog simulation, we assumed the half-power pulse width  $\tau_H$  of the Velodyne HDL-64E sensor to be 20ns and set  $\beta$  to  $\frac{0.046}{\text{MOR}}$  as in Rasshofer *et al.* [30]. We empirically set  $\beta_0$  to  $\frac{1 \times 10^{-6}}{\pi}$  for all points to get a similar intensity distribution as we can observe in the real foggy point clouds of STF [2]. Since the Velodyne HDL-64E uses some unknown internal dynamic gain mechanism, it delivers at each and every time step intensity values in the full value range  $[0, 255]$ . To mimic this behaviour and also cover the full value range again we linearly scale up the intensity values after they have been modified by Algorithm 1.

### 4.2.1 Quantitative Results

For the numbers we report, we select the snapshot with the best performance on the clear weather validation set and test it on the aforementioned test splits. In Table 1 we report the 3D average precision (AP) on the STF [2] dense fog test split for the classes *Car*, *Cyclist* and *Pedestrian* as well as the 3D mean average precision (mAP) over those three classes. Note, it is always one model that predicts all three classes and *not* one model per class. All AP and mAP numbers reported in this paper are being calculated using 40 recall positions as suggested in [38]. We can see that in mAP over all classes and on the major *Car* class, the training runs of all methods using our fog simulation outperforms the clear weather baseline and the training runs using the fog simulation in [2].

As a second baseline, we evaluate the clear weather model after applying an additional preprocessing step at test time, where we only feed those points to the network that are present in both, the *strongest* and *last* measurement of the same scene. We dub this filter “*strongest*  $\cap$  *last filter*”.

Method	$\alpha$	Car AP@.5IoU			Cyclist AP@.25IoU			Pedestrian AP@.25IoU			mAP over classes		
		easy	mod.	hard	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard
PV-RCNN [34] †	0	45.03	46.00	45.08	24.33	24.63	24.63	43.96	41.92	40.09	37.77	37.51	36.60
PV-RCNN [34] ‡	0	45.24	46.18	45.25	24.38	24.67	24.67	<b>44.81</b>	<b>43.09</b>	<b>40.98</b>	38.15	37.98	36.97
fog simulation in [2]	*	45.60	46.60	45.60	26.42	26.93	27.80	42.95	40.89	39.09	38.32	38.14	37.50
our fog simulation	*	<b>46.69</b>	<b>47.38</b>	<b>46.51</b>	<b>27.89</b>	<b>27.89</b>	<b>29.29</b>	42.38	40.65	39.20	<b>38.99</b>	<b>38.64</b>	<b>38.33</b>
PointRCNN [36] †	0	44.00	45.03	43.73	22.99	22.99	24.23	41.73	38.38	35.71	36.24	35.47	34.56
PointRCNN [36] ‡	0	44.40	45.25	44.15	<b>23.52</b>	<b>23.52</b>	<b>25.62</b>	43.23	40.16	37.05	37.05	36.31	35.61
fog simulation in [2]	*	46.08	47.02	45.85	20.36	20.36	20.36	43.03	41.60	39.95	36.49	36.33	35.39
our fog simulation	*	<b>47.81</b>	<b>47.99</b>	<b>46.68</b>	22.88	22.88	25.18	<b>45.79</b>	<b>43.47</b>	<b>41.33</b>	<b>38.83</b>	<b>38.11</b>	<b>37.73</b>
SECOND [47] †	0	42.36	42.99	41.99	24.03	25.21	25.21	36.72	35.37	33.84	34.37	34.52	33.68
SECOND [47] ‡	0	42.78	43.47	42.42	22.32	23.69	23.69	37.06	36.14	34.14	34.05	34.43	33.42
fog simulation in [2]	*	42.67	43.58	42.77	26.28	27.11	27.11	37.89	36.54	35.38	35.61	35.74	35.09
our fog simulation	*	<b>43.47</b>	<b>44.01</b>	<b>43.20</b>	<b>26.85</b>	<b>27.21</b>	<b>27.46</b>	<b>38.41</b>	<b>37.06</b>	<b>35.87</b>	<b>36.24</b>	<b>36.09</b>	<b>35.51</b>
Part-A <sup>2</sup> [37] †	0	37.60	38.15	37.76	24.51	25.59	25.59	<b>41.03</b>	<b>39.29</b>	<b>37.59</b>	34.38	34.34	33.65
Part-A <sup>2</sup> [37] ‡	0	38.04	38.73	38.30	24.37	25.45	25.45	40.36	38.55	36.65	34.26	34.25	33.47
fog simulation in [2]	*	41.07	41.63	40.81	21.12	21.12	21.12	38.83	37.57	34.94	33.67	33.44	32.29
our fog simulation	*	<b>42.16</b>	<b>42.75</b>	<b>41.70</b>	<b>25.13</b>	<b>25.72</b>	<b>26.22</b>	39.19	38.29	36.29	<b>35.49</b>	<b>35.59</b>	<b>34.74</b>
PointPillars [23] †	0	34.30	35.23	35.00	23.05	23.26	25.50	26.43	25.35	24.17	27.93	27.95	28.22
PointPillars [23] ‡	0	34.89	35.84	35.47	<b>24.14</b>	<b>24.36</b>	<b>25.38</b>	27.17	26.04	24.85	28.74	28.75	28.57
fog simulation in [2]	*	37.02	38.16	37.88	21.68	21.68	23.33	28.84	28.25	26.95	29.18	29.37	29.39
our fog simulation	*	<b>38.31</b>	<b>39.14</b>	<b>38.91</b>	23.40	23.40	25.37	<b>30.50</b>	<b>29.51</b>	<b>27.91</b>	<b>30.73</b>	<b>30.68</b>	<b>30.73</b>

Table 1: 3D average precision (AP) results on the STF [2] dense fog test split.

† clear weather baseline ‡ clear weather baseline (same model as †) with strongest  $\cap$  last filter applied at test time

\* fog simulation gets applied to every training example with  $\alpha$  uniformly sampled from [0, 0.005, 0.01, 0.02, 0.03, 0.06]

Method	$\alpha$	dense fog			light fog			clear			mAP over conditions		
		easy	mod.	hard	easy	mod.	hard	easy	mod.	hard	easy	mod.	hard
PV-RCNN [34] †	0	45.03	46.00	45.08	69.55	70.17	68.44	79.61	77.05	71.03	64.73	64.41	61.52
PV-RCNN [34] ‡	0	45.24	46.18	45.25	69.64	70.30	68.42	<b>79.80</b>	77.16	71.08	64.89	64.55	61.58
fog simulation in [2]	*	45.60	46.60	45.60	70.02	70.56	<b>69.37</b>	79.63	<b>77.48</b>	<b>72.48</b>	65.08	64.88	<b>62.48</b>
our fog simulation	*	<b>46.69</b>	<b>47.38</b>	<b>46.51</b>	<b>71.42</b>	<b>70.96</b>	69.03	79.27	76.75	71.80	<b>65.79</b>	<b>65.03</b>	62.45
PointRCNN [36] †	0	44.00	45.03	43.73	71.30	<b>71.48</b>	<b>68.31</b>	80.05	76.52	<b>70.80</b>	65.12	64.34	60.95
PointRCNN [36] ‡	0	44.40	45.25	44.15	<b>71.36</b>	70.45	68.28	79.96	76.37	70.59	65.24	64.02	61.01
fog simulation in [2]	*	46.08	47.02	45.85	70.80	70.27	67.66	79.90	76.16	69.18	65.59	64.48	60.90
our fog simulation	*	<b>47.81</b>	<b>47.99</b>	<b>46.68</b>	70.74	70.84	67.65	<b>80.41</b>	<b>76.58</b>	69.68	<b>66.32</b>	<b>65.14</b>	<b>61.34</b>
SECOND [47] †	0	42.36	42.99	41.99	<b>70.51</b>	70.07	68.60	78.67	75.20	70.67	63.85	62.75	60.42
SECOND [47] ‡	0	42.78	43.47	42.42	70.50	<b>70.08</b>	<b>68.63</b>	78.53	75.08	69.91	63.93	62.87	60.32
fog simulation in [2]	*	42.67	43.58	42.77	69.67	69.89	68.45	79.23	<b>76.61</b>	71.89	63.85	<b>63.36</b>	61.04
our fog simulation	*	<b>43.47</b>	<b>44.01</b>	<b>43.20</b>	69.55	69.63	68.49	<b>79.44</b>	75.95	<b>71.94</b>	<b>64.15</b>	63.20	<b>61.21</b>
Part-A <sup>2</sup> [37] †	0	37.60	38.15	37.76	65.29	65.88	64.31	76.38	73.79	68.56	59.76	59.27	56.88
Part-A <sup>2</sup> [37] ‡	0	38.04	38.73	38.30	65.98	66.41	64.62	76.43	<b>73.86</b>	<b>68.57</b>	60.15	59.67	57.16
fog simulation in [2]	*	41.07	41.63	40.81	65.91	65.84	63.91	76.61	73.84	68.31	61.20	60.44	57.68
our fog simulation	*	<b>42.16</b>	<b>42.75</b>	<b>41.70</b>	<b>68.12</b>	<b>67.76</b>	<b>65.19</b>	<b>76.64</b>	<b>73.86</b>	68.05	<b>62.31</b>	<b>61.46</b>	<b>58.32</b>
PointPillars [23] †	0	34.30	35.23	35.00	67.92	68.47	66.73	77.20	74.64	<b>69.63</b>	59.81	59.45	57.12
PointPillars [23] ‡	0	34.89	35.84	35.47	67.97	68.52	66.76	77.27	<b>74.66</b>	69.59	60.04	59.67	57.27
fog simulation in [2]	*	37.02	38.16	37.88	68.18	68.24	67.10	76.33	73.91	69.03	60.51	60.10	58.00
our fog simulation	*	<b>38.31</b>	<b>39.14</b>	<b>38.91</b>	<b>68.31</b>	<b>68.95</b>	<b>67.18</b>	<b>77.42</b>	74.56	69.55	<b>61.34</b>	<b>60.88</b>	<b>58.55</b>

Table 2: Car 3D AP@.5IoU results on all relevant STF [2] test splits.

† clear weather baseline ‡ clear weather baseline (same model as †) with strongest  $\cap$  last filter applied at test time

\* fog simulation gets applied to every training example with  $\alpha$  uniformly sampled from [0, 0.005, 0.01, 0.02, 0.03, 0.06]

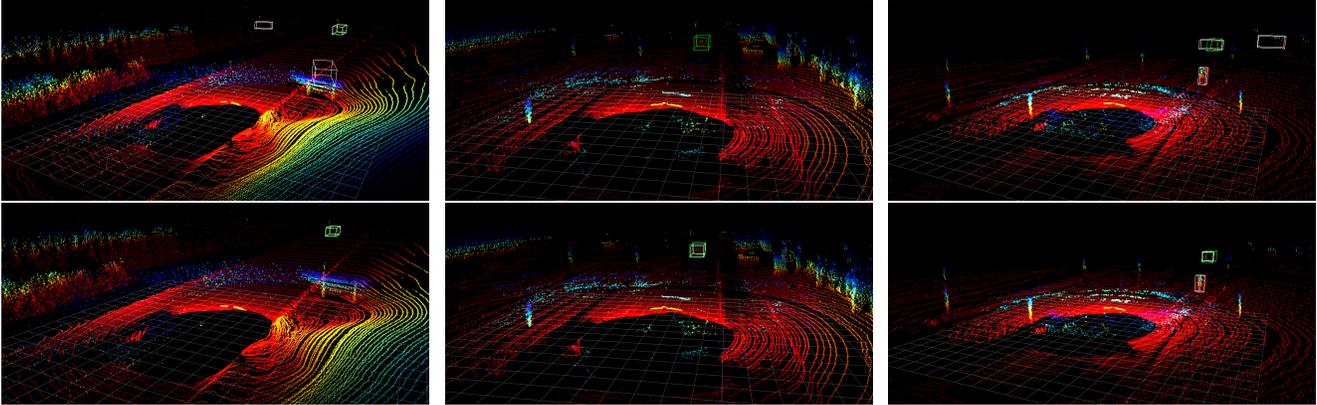


Figure 6: The (top) row shows predictions by PV-RCNN [34] trained on the original clear weather data (first row in tables above), the (bottom) row shows predictions by PV-RCNN [34] trained on a mix of clear weather and simulated foggy data (fourth row in tables above) on three example scenes from the STF [2] dense fog test split. Ground truth boxes in color, predictions of the model in white. Best viewed on a screen (and zoomed in).

The idea for this filter stems from the fact, that all the points that get discarded by this filter, must be noise (most likely introduced by fog in the scene) and can not be from a physical object of interest. We can see that this filter most of the time boosts the performance of the clear weather model but also does not surpass any fog simulation runs for the majority of cases. One might also notice, that the performance on the *Cyclist* class is generally lower than on the other two classes. We attribute this to the fact, that the *Cyclist* class is fairly underrepresented compared to the other two classes in the STF [2] dataset (e.g. 28 cyclists vs. 490 pedestrians and 1186 cars in the dense fog test split). For the *Pedestrian* (and *Cyclist*) class we still achieve three out of five times state-of-the-art performance. For the training runs using either our fog simulation or the fog simulation in [2], we uniformly sample for each training example  $\alpha$  from [0, 0.005, 0.01, 0.02, 0.03, 0.06] which corresponds to a MOR of approximately [ $\infty$ , 600, 300, 150, 100, 50]m respectively. Trying out more sophisticated techniques like curriculum learning [1] is kept for future work.

In Table 2 we present the 3D AP of the major *Car* class on the dense fog, light fog and clear test set as well as the mAP over those three weather conditions. We can see that in dense fog the training runs of all methods using our fog simulation outperforms all other training runs, which is exactly what we aimed for with our physically accurate fog simulation. We can further see that mixing in our fog simulation in training does not hurt the performance in clear weather too much, hence we also achieve state-of-the-art for most cases in mAP over all three weather conditions.

In the Supplementary Materials, we discuss why we chose to focus on relaxed intersection over union (IoU) thresholds and present results using the official KITTI [10] evaluation strictness. Additionally, we present 2D and Birds

Eye View (BEV) results, and further details on the STF [2] dataset.

#### 4.2.2 Qualitative Results

In Fig. 6 we showcase three examples where we clearly outperform the clear weather baseline. We can examine that the model that sees our simulated fog in training, has less false positives (left), more true positives (middle) and overall more accurate predictions (right), each time applying the same confidence threshold for a fair comparison.

## 5. Conclusion

In this work we introduce a physically accurate way to convert real-world clear weather point clouds into *foggy* point clouds. In this process we have full control over all parameters involved in the physical equations. This not only allows us to realistically simulate any density of fog, but also allows us to simulate the influence of fog on basically any LiDAR sensor currently available on the market.

We show that by using this physically accurate fog simulation, we can improve the performance of several state of the art 3D object detection methods on point clouds that have been collected in real-world dense fog. We expect that our fog simulation can lead to even greater performance boosts if the LiDAR data is annotated in 360° and not just in the field of view of a single forward facing camera, but no such dataset is publicly available yet to test this hypothesis.

We believe that our physically accurate fog simulation is not just applicable to the task of 3D object detection. So we hope that our fog simulation also finds its way into many other tasks and works.

**Acknowledgements:** This work was funded by Toyota Motor Europe via the research project TRACE Zurich.

## References

- [1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, 2009.
- [2] Mario Bijelic, Tobias Gruber, Fahim Mannan, Florian Kraus, Werner Ritter, Klaus Dietmayer, and Felix Heide. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [3] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint 1903.11027*, 2019.
- [4] A. Carballo, J. Lambert, A. Monrroy, D. Wong, P. Narksri, Y. Kitsukawa, E. Takeuchi, S. Kato, and K. Takeda. Libre: The multiple 3d lidar dataset. In *IEEE Intelligent Vehicles Symposium (IV)*, 2020.
- [5] M. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3d tracking and forecasting with rich maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [7] A. Dewan, T. Caselitz, G. D. Tipaldi, and W. Burgard. Motion-based detection and tracking in 3d lidar scans. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [8] M. Elhousni and X. Huang. A survey on 3d lidar localization for autonomous vehicles. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020.
- [9] A. Filgueira, Higinio Gonzalez, Susana Lagüela, Lucia Díaz Vilarinho, and Pedro Arias. Quantifying the influence of rain in lidar performance. *Measurement*, 95, 2016.
- [10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [11] Jakob Geyer, Yohannes Kassahun, Mentar Mahmudi, Xavier Ricou, Rupesh Durgesh, Andrew S. Chung, Lorenz Hauswald, Viet Hoang Pham, Maximilian Mühlegg, Sebastian Dorn, Tiffany Fernandez, Martin Jänicke, Sudesh Mirashi, Chiragkumar Savani, Martin Sturm, Oleksandr Vorobiov, Martin Oelker, Sebastian Garreis, and Peter Schuberth. A2d2: Audi autonomous driving dataset. *arXiv preprint 2004.06320*, 2020.
- [12] Christopher Goodin, Daniel Carruth, Matthew Doude, and Christopher Hudson. Predicting the influence of rain on lidar in adas. *Electronics*, 8, 2019.
- [13] Martin Hahner, Dengxin Dai, Christos Sakaridis, Jan-Nico Zaeck, and Luc Van Gool. Semantic understanding of foggy scenes with purely synthetic data. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2019.
- [14] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, and W. Stork. Weather influence and classification with automotive lidar sensors. In *IEEE Intelligent Vehicles Symposium (IV)*, 2019.
- [15] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.
- [16] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 42, 2020.
- [17] Maria Jokela, Matti Kutila, and Pasi Pyykönen. Testing and validation of automotive point-cloud sensors in adverse weather conditions. *Applied Sciences*, 9, 2019.
- [18] Frank D. Julca-Aguilar, Jason Taylor, Mario Bijelic, Fahim Mannan, Ethan Tseng, and Felix Heide. Gated3d: Monocular 3d object detection from temporal illumination cues. *arXiv preprint 2102.03602*, 2021.
- [19] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 perception dataset 2020. <https://level5.lyft.com/dataset>, 2019.
- [20] Isaac I. Kim, Bruce McArthur, and Eric J. Korevaar. Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications. In *Optical Wireless Communications III*, volume 4214. International Society for Optics and Photonics, SPIE, 2001.
- [21] M. Kutila, P. Pyykönen, H. Holzhüter, M. Colomb, and P. Duthon. Automotive lidar performance verification in fog and rain. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2018.
- [22] M. Kutila, P. Pyykönen, W. Ritter, O. Sawade, and B. Schäufele. Automotive lidar sensor development scenarios for harsh weather conditions. In *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2016.
- [23] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [24] Y. Li, P. Duthon, M. Colomb, and J. Ibanez-Guzman. What happens for a tof lidar in fog? *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [25] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song. L3-net: Towards learning based lidar localization for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [26] S. Michaud, Jean-François Lalonde, and P. Giguère. Towards characterizing the behavior of lidars in snowy conditions. In *International Conference on Intelligent Robots and Systems (IROS)*, 2015.
- [27] Abhishek Patil, Srikanth Malla, Haiming Gang, and Yi-Ting Chen. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. *arXiv preprint 1903.01568*, 2019.

- [28] Matthew Pitropov, Danson Garcia, Jason Rebello, Michael Smart, Carlos Wang, Krzysztof Czarnecki, and Steven Waslander. Canadian adverse driving conditions dataset. *arXiv preprint 2001.10117*, 2020.
- [29] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [30] R. Rasshofer, M. Spies, and H. Spies. Influences of weather phenomena on automotive laser radar systems. *Advances in Radio Science*, 9, 2011.
- [31] Christos Sakaridis, Dengxin Dai, Simon Hecker, and Luc Van Gool. Model adaptation with synthetic and real data for semantic dense foggy scene understanding. In *European Conference on Computer Vision (ECCV)*, 2018.
- [32] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126, 2018.
- [33] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *IEEE International Conference on Computer Vision (ICCV)*, 2021.
- [34] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [35] Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, and Hongsheng Li. PV-RCNN+: point-voxel feature set abstraction with local vector representation for 3d object detection. *arXiv preprint 2102.00463*, 2021.
- [36] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [37] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.
- [38] Andrea Simonelli, Samuel Rota Buló, Lorenzo Porzi, Manuel Lopez-Antequera, and Peter Kotschieder. Disentangling monocular 3d object detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.
- [39] Pei Sun, Henrik Kretschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [40] OpenPCDet Development Team. Openpcdet: An open-source toolbox for 3d object detection from point clouds. <https://github.com/open-mmlab/OpenPCDet>, 2020.
- [41] Maxime Tremblay, Shirsendu S. Halder, Raoul de Charette, and Jean-François Lalonde. Rain rendering for evaluating and improving robustness to bad weather. *International Journal of Computer Vision (IJCV)*, 126, 2021.
- [42] Velodyne. Hdl-64e user’s manual, 2007.
- [43] Sourabh Vora, Alex H. Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [44] A. M. Wallace, A. Halimi, and G. S. Buller. Full waveform lidar for adverse weather conditions. *IEEE Transactions on Vehicular Technology*, 69, 2020.
- [45] Jacek Wojtanowski, Marek Zygmunt, Mirosława Kaszczuk, Z. Mierczyk, and Michał Muzal. Comparison of 905 nm and 1550 nm semiconductor laser rangefinders’ performance deterioration due to adverse environmental conditions. *Opto-Electronics Review*, 22, 2014.
- [46] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *arXiv preprint 1810.08705*, 2018.
- [47] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18, 2018.
- [48] Ji Zhang and Sanjiv Singh. Loam : Lidar odometry and mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, 2014.
- [49] Y. Zhou and O. Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
- [50] Claudia Álvarez Aparicio, Ángel Manuel Guerrero-Higueras, Francisco Javier Rodríguez-Lera, Jonatan Ginés Clavero, Francisco Martín Rico, and Vicente Matellán. People detection and tracking using lidar sensors. *Robotics*, 8, 2019.