Simulating Realistic Rain, Snow, and Fog Variations For Comprehensive Performance Characterization of LiDAR Perception

Sven Teufel[1], Georg Volk[1], Alexander von Bernuth[1] and Oliver Bringmann[1]

***Abstract*—For robust object detection on LiDAR data, neural networks have to be trained on diverse datasets that contain many different environmental influences like rain, snow, or fog. To this date, few datasets, with those features, are available while there exist many datasets recorded under perfect weather conditions. Repurposing those datasets by simulating adverse environmental conditions on top of them and training networks with the resulting enhanced datasets, is intended to lead to more robust neural networks. In the following we propose models to realistically simulate the effects of rain, snow, and fog on LiDAR datasets based on physical and empirical fundamentals. Then we parameterize our simulation to best fit real LiDAR data that was captured in those environments, in order to achieve a highly accurate simulation. Finally, the impact of adverse weather on neural network detection performance is demonstrated.**

I. INTRODUCTION

Establishing trust in autonomous vehicles is becoming a major task for car manufacturers all over the industry. In the light of multiple prominently discussed collisions involving self driving cars, not only regulators and lawmakers are watching with caution—potential customers are also having second thoughts on trusting their well-being on apparently untrustworthy systems.

Important to realize is, that driving to work or to shop is already a relatively dangerous task. In Germany, a country of 83 million inhabitants, 2.7 million road accidents occur every year—injuring over 380,000 people and killing over 3,000 [1]. Many of those deaths are avoidable, as 40% of fatal accidents at least involve alcohol, distraction, drugs, or fatigue [2]; factors that are not immediately relevant when driving an autonomous vehicle. Human error is responsible for over 90% of all crashes. With that in mind, taking drivers out of the equation might make traffic safer overall.

Increasing the miles traveled without incident also increases the trust in autonomous systems [3]. So to regain the trust that has been lost in autonomous vehicles, carmakers have to create safe and well tested products that prove themselves over time. Utilizing established standards for automotive safety processes like ISO 26262 can help achieving this goal. Due to machine learning algorithms being deployed in virtually all autonomous vehicles, verification of those algorithms regarding strict standards is becoming a key task. Neither can the used algorithms be fully explained, nor their behavior fully predicted—especially in adversarial conditions like intense weather or malicious attacks [4].

In this paper, we propose models to generate more diverse LiDAR datasets that can be either used for performance characterization of LiDAR-based object detection or its robustness improvement against environmental influences. Thus, we simulate weather conditions, especially heavy rain, thick snow, and dense fog realistically. This enables the reuse of existing datasets which do not contain adverse weather conditions by extending them with the desired environmental conditions. Therefore, the proposed approach drastically reduces the time to gather diverse datasets for training robust neural networks, as no additional real world test drives are needed to capture this data.

Section II investigates current state of the art for simulating environmental conditions on LiDAR data. Sections III to V address our proposed models for simulating rain, snow, and fog respectively. In Section VI the influence of these adverse weather conditions on LiDAR object detection are evaluated. Section VII concludes this paper and provides an outlook.

II. RELATED WORK

Having robust algorithms which allow an autonomous vehicle to correctly perceive the environment even in adverse environmental conditions is a key requirement to enable fully autonomous vehicles. Especially for the development and verification of such algorithms the used data needs to contain as many different scenarios and environmental conditions as possible to guarantee robustness.

A. Impact of adverse weather on LiDAR

The impact of adverse weather conditions on LiDAR sensor data results from different phenomena that occur because of the interaction between the laser beam and the raindrops, fog droplets, or snowflakes. LiDAR point clouds which were recorded under adverse weather conditions show less points compared to the ones recorded under clear weather conditions in the same scenario. This is due to absorption and scattering of the laser beam at the particles [5], [6].

Another observation is the occurrence of false points in front of the sensor caused by backscattered light from the particles in adverse weather [7], [8]. Moreover, the intensity of the reflected light is affected by the weather condition. On the one hand, the intensity might be increased by an increase in the reflectivity of a surface, for instance by snow covered roads compared to clear roads. On the other hand, the intensity

[1] University of Tübingen, Faculty of Science, Department of Computer Science, Embedded Systems Group {sven.teufel, georg.volk, alexander.von-bernuth, oliver.bringmann} @uni-tuebingen.de

might decrease because of a partial absorption or scattering of the laser beam or the reduction of the surface reflectivity, as it is the case for wet surfaces [9].

B. Datasets

Existing datasets like the well-known KITTI [10] or the more recent Appollocscape dataset [11] offer no environmental variation in LiDAR data. Most data is captured during good weather conditions. Hence, training neural networks with such data does not result in robust algorithms as the performance of neural networks heavily depends on the dataset used for training. Diverse datasets are essential for training neural networks to be robust [12]. Other datasets offer recordings during rainfall (Waymo [13]) or even in harsh conditions like dense fog or heavy snowfall (DENSE dataset [14]). These datasets are the better choice for training neural networks. However, when it comes to evaluating the impact of adverse weather conditions on LiDAR-based object detection it is important to have knowledge about the intensity of the current weather, which is lacking in parts of these datasets. Otherwise, the algorithms' robustness cannot be determined. Additionally, the evaluation is restricted to the scenes and conditions present in these datasets.

C. Artificial weather

Another approach, compared to capturing adverse weather conditions through real world test drives, is to simulate these desired conditions. Zang et al. [7] investigated the influences of adverse weather conditions to different sensors like LiDAR, camera, RADAR, or GPS. They simulated attenuation and backscatter effects to estimate detection ranges at different rain rates for RADAR. Different signal to noise ratios for fog, snow, and rain for LiDAR sensors were evaluated by Hespel et al. [15]. Filgueira et al. [16] focused on investigating the influence of rain on LiDAR performance. Turbulent snow influence on LiDAR was analyzed by Jokela et al. [17] by driving a test vehicle behind another vehicle on snowy roads. Hasirlioglu et al. [18] used an indoor fog simulation facility with different fog layers to analyze the effect on LiDAR sensors. Similarly, Heinzler et al. [5] examined the influence on classification with LiDAR sensors in rain and fog with the CEREMA weather simulation facility and real world recordings. The performance of multiple LiDAR sensors in dense fog was evaluated by Bijelic et al. [6]. They made recordings in the CEREMA facility with different settings for each sensor and at varying fog intensities. All these approaches to evaluate the sensor performance either use theoretical attenuation models, lacking validation against real-world data, or use real-world test drives or an indoor weather simulation facility, which are restricted to the evaluated scenarios.

The CEREMA facility was also used by Li et al. [19] to analyze the effect of fog on LiDAR-based object detection. They tried to predict the object detection quality for given reflectivities and object distances. Influences of rain, snow, and fog on LiDAR were investigated by Rasshofer et al. [8]. They recorded the backscatter signal of a LiDAR target in a foggy environment and replicated it with their electro-optical laser target simulation. Therefore they were able to replay the recorded scenario for different LiDAR sensors. Still, the presented laser target simulator is not as easily adaptable as simulating a LiDAR point cloud in a virtual environment.

Hasirlioglu et al. [9] showed an approach to simulate rain on automotive surround sensors including LiDAR. They used a noise filter of distributed spheres in the sensor's field of view, whose sphere sizes were determined by a raindrop size distribution. Afterwards ray tracing was performed on this noise filter: the beam divergence is approximated by generating multiple rays per point into the point cloud at an offset angle. Points in the point cloud were then modified if enough rays intersect with the noise filter. However, this approach only manipulates the point cloud by moving points closer towards the sensor. For a more realistic simulation points also need to be removed. Additionally to the model presented by Hasirlioglu et al. the simulation approach for rain by Goodin et al. [20] is not only capable of modifying points but also of removing them. Hadj-Bachir and de Souza [21] presented a simulation of rain and fog for virtual environments. However, their method lacks the theoretical basis and similarly to the method of Goodin et al. an optimization of the simulation with real rain is missing.

III. RAIN SIMULATION

For the simulation of rain on point clouds we used the approach introduced by Hasirlioglu et al. [9] with some improvements. By modeling rainfall in the sensor's field of view they generated a noise filter. It consists of spheres, which represent the raindrops. To each drop a three-dimensional position in the sensor's field of view is assigned. The sizes of the raindrops were determined by a raindrop size distribution. This noise filter is then used for ray tracing. In order to take the beam divergence into account there are multiple generated rays per point with an offset to each other. The point cloud is modified by setting scan points closer to the sensor if the hit ratio for a point is above a threshold.

In the same manner as in [9] we used a noise filter consisting of spherical raindrops on which ray tracing is performed. The raindrop positions are uniformly distributed in the sensor's field of view. Instead of the proposed maximum diameter of $8.5\,\mathrm{mm}$ as in [9] we limited the size of each raindrop to $6\,\mathrm{mm}$ since this is the largest diameter for a stable raindrop [22]. Furthermore, we assumed the laser beam of the sensor to be circular instead of rectangular, which corresponds better to the real beam shape [23].

Each ray is calculated by interpreting a scan point P as a vector $\vec{P}$ and rotating $\vec{P}$ around the rotation vector $\vec{R}$. The initial $\vec{R}$ is the cross product of $\vec{P}$ with the Z axis, the offset angle δ is given by

$$\delta = \frac{\phi}{2 \cdot N_r}$$

where ϕ is the beam divergence in degree and N_r the number of rays in the circle's radius. $\vec{R}$ is then rotated around $\vec{P}$, the angle of rotation α is given as $\alpha = 360^\circ / N_c$ where N_c is the

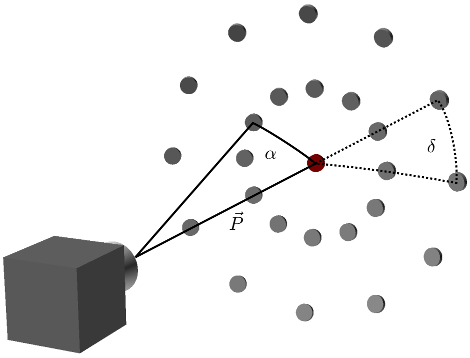


Fig. 1: Beam divergence visualization with $N_r = 2$ and $N_c = 12$. The red center point is the original point from the point cloud, pointed at by $\vec{P}$. The gray dots represent the end points of the rays which were generated.

number of lines in the circle starting from the center point. As can be seen in Fig. 1, the ray generation process results in a starlike shape with the original point in the center. Each ray is then checked for potentially hitting a raindrop in the noise filter. As suggested by Hasirlioglu et al. a scan point is modified if the hit ratio R_{all} of the rays from this point with the noise filter is over a certain threshold T_{all}. The hit ratio is calculated as

$$R_{\text{all}} = \frac{N_{\text{intersects}}}{N_{\text{rays}}}$$

where $N_{\text{intersects}}$ is the number of rays that intersect with raindrops and N_{rays} is the total number of rays sent around a single point. In the approach of Hasirlioglu et al. the point is then set to the closest intersection with the noise filter, whereas we introduced another threshold for the ratio R_{most}, defined as

$$R_{\text{most}} = \frac{N_{\text{most}}}{N_{\text{intersects}}}$$

where N_{most} is the number of rays intersecting the raindrop with most intersections. If R_{most} is larger than a certain threshold T_{most} the point is set towards the sensor; if it is lower the scan point is deleted to cover the loss of scan points caused by absorption and scattering. Another improvement was to place the points at the position of the raindrop with most intersections, instead of the closest intersection, since this is the hit with the highest expected intensity.

IV. SNOW SIMULATION

The snow simulation is—like the rain simulation—based on ray tracing. The sizes of the snowflakes were estimated by the distribution from Gunn and Marshall [24] with the optimizations of Sekhon and Srivastava [25]:

$$\begin{aligned} N(D) &= N_0 \cdot e^{-\Lambda \cdot D} \\ N_0 &= 2500 \cdot R^{-0.94}\,\mathrm{m}^{-3}\,\mathrm{mm}^{-1} \\ \Lambda &= 2.29 \cdot R^{-0.48}\,\mathrm{mm}^{-1} \end{aligned}$$

where R is the precipitation rate and determines the snowflake size distribution $N(D)$ of diameter D.

As this distribution describes the molten diameter, a scaling factor was used to get the real diameter of the snowflakes. This factor depends on the shape of the snowflakes and ranges between 1 and 5 [26]. A low scaling factor should be used for snow with high water content, whereas a high scaling factor should be used for dry snow, especially for dendrites or aggregated dendrites. The number of snowflakes can be calculated with the mass concentration M_s [27]:

$$M_s = \begin{cases} 0.30 \cdot R & \text{dense snowfall,} \\ 0.47 \cdot R & \text{light snowfall.} \end{cases}$$

Further, to calculate the number of snowflakes N_s per unit volume, the mass concentration was used [28]:

$$N_s = \frac{M_s}{m_d},$$

where m_d is the average mass of the snowflakes.

From observations of the DENSE dataset we found that the intensities of false points resulting from snowflakes are lognormally distributed. The probability distribution function, which was obtained by fitting the false point intensities in the dataset, was defined as follows:

$$I(x) = \frac{1}{(x-\Theta)\sigma\sqrt{2\pi}} \cdot \exp\left(\frac{-\left(\ln \frac{x-\Theta}{m}\right)^2}{2\sigma^2} \right),$$

where Θ denotes the shift of the distribution, σ is the standard deviation and m is a scaling parameter. These three parameters are determined numerically; more on that in VI-A2. The intensity of the remaining scan points was increased by 25% for snowy surfaces or reduced by 10% for wet surfaces.

V. FOG SIMULATION

In fog, the number of droplets per unit volume is very high compared to rain, being in the order of $100\,\mathrm{cm}^{-3}$. This is 10^5 times the number of drops in rain which typically is at $0.001\,\mathrm{cm}^{-3}$ [8]. Because of this huge amount of droplets in fog a ray tracing approach to simulate the effects on LiDAR is not practical due to the enormous computational effort. Instead, we propose an efficient probabilistic model for simulating fog effects on LiDAR, assuming evenly dense fog. We used different probabilities and distributions to modify point clouds which were recorded in clear weather, so that they matched the characteristics of fog influence on LiDAR.

First we selected points for modification. To calculate the probability that a point is modified we used an exponential function depending on the distance of the point from the sensor d, since the attenuation in fog is increasing exponentially with the distance traveled [29]:

$$p_{\text{modify}}(d) = 1 - e^{-d \cdot \epsilon}.$$

Here, ϵ is a scaling parameter, whose value is explained in section VI.

If a point in the point cloud was chosen for modification, this point was either deleted to fit the reduction of points in adverse weather, or the point was moved towards the sensor to fit the occurrence of false points due to back scattering. The probability that a point was deleted was determined by

$$p_{\text{delete}} = a \cdot e^{b \cdot V} + 1,$$

where a and b are parameters and V is the meteorological visibility in m. If a point was chosen to be modified but not chosen to be deleted, it was moved.

In the event that the point was moved towards the sensor, the new distance of the point was determined by the probability distribution function

$$f(x) = \frac{1}{\lambda} \cdot \exp\left(-\frac{1}{\lambda} \cdot x\right),$$

where λ is an empirically determined parameter explained in section VI. To make sure no point was set within the minimum distance d_{min} the sensor can measure, d_{min} was added to the new point distance.

The intensity of the moved points was randomly set to 0–32% of the maximum intensity, while the intensity of the unaltered points was calculated with the law of Beer and Lambert as

$$I(d) = I_0 \cdot e^{-\gamma d},$$

where I_0 is the original intensity, d is the traveled distance in fog and γ is the extinction coefficient from [30]

$$\gamma = \frac{-\ln(0.05)}{V}.$$

Since LiDAR is an active sensor, the traveled distance is twice the distance to the scan point.

The parameters ϵ, a, b, λ for the equations used in this section were obtained by modifying point clouds captured in clear weather until they best match point clouds in foggy weather with known intensity. For this we used the recordings from Bijelic et al. [6] in the CEREMA fog chamber. To compare the simulated point clouds to the real ones we used two different metrics. One is the widely known chamfer distance

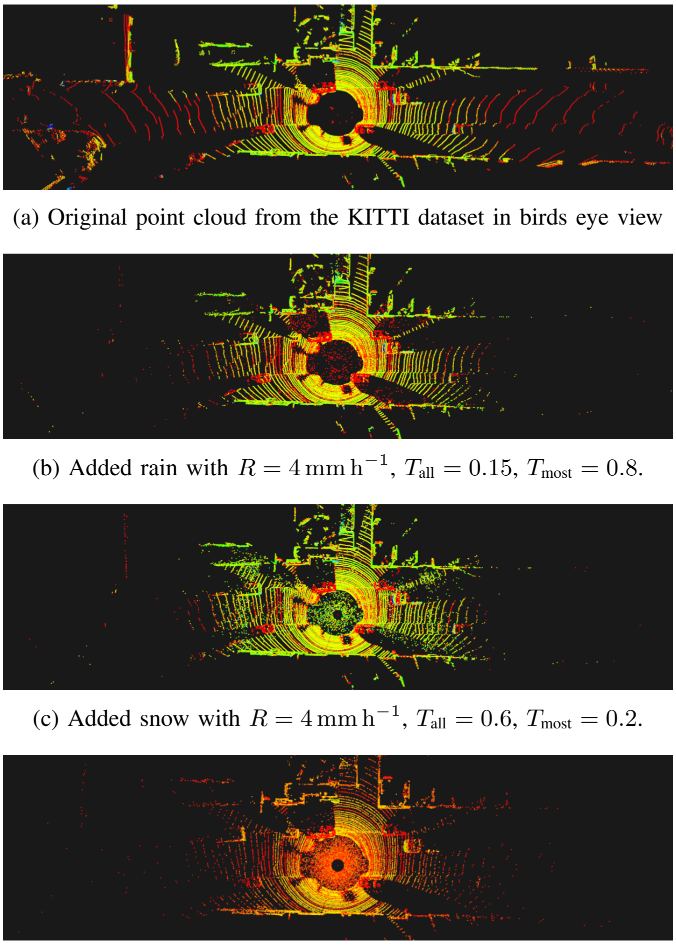
$$\text{CD}(P, Q) = \sum_{p \in P} \min_{q \in Q} ||p - q||_2^2 + \sum_{q \in Q} \min_{p \in P} ||p - q||_2^2$$

where P and Q are two point clouds [31]. Additionally, we used a metric which compares the point density by distance in the point cloud. For this metric the distance distribution for both point clouds is calculated, then those distributions are compared by the Wasserstein metric [32].

The simulation parameters were adjusted to minimize the distance between the simulated point clouds and the real point clouds for each metric separately.

VI. Results

In this section we first present the parameterization of our weather simulations to match real adverse weather conditions. In addition to the final model parameterization, the influence of adverse weather conditions on LiDAR-based object detection, including the influence on different object classes, will be evaluated.



(a) Original point cloud from the KITTI dataset in birds eye view

(b) Added rain with $R = 4\,\mathrm{mm\,h^{-1}}$, $T_{\text{all}} = 0.15$, $T_{\text{most}} = 0.8$.

(c) Added snow with $R = 4\,\mathrm{mm\,h^{-1}}$, $T_{\text{all}} = 0.6$, $T_{\text{most}} = 0.2$.

(d) Added fog with $V = 80\,\mathrm{m}$.

Fig. 2: Influence of rain, snow, and fog on a point cloud. The color of the points indicates the intensity.

A. Model parameterization

In order to generate realistic weather conditions we used the DENSE dataset by Bijelic et al. [14] to parameterize our proposed weather simulations. The DENSE dataset is an adverse weather dataset containing real world test drives and fog chamber recordings. In contrast to other available datasets, this dataset also includes heavy rain, snow, and fog. We adjusted the parameters in our models to best fit the recordings under these conditions. For snow we used the available real world test drives and for rain and fog the captures of the CEREMA weather simulation facility.

1) Rain: To achieve a realistic modification of the intensity values of the LiDAR point cloud during our rain simulation we analyzed the present intensity values of the DENSE dataset. The parameters and thresholds were then set based on these empirical findings. Since the intensity of false points is under 0.5% of its maximum for 99.3% of the points, we set the intensity from 0–0.5% with a uniform distribution. The remaining scan points were set to 90% of their original intensity because of the reduced reflectivity of wet surfaces as suggested by Hasirlioglu et al. [9]. Furthermore we parameterized the thresholds for the rain simulation with $T_{\text{all}} = 0.15$ and $T_{\text{most}} = 0.8$. For the parameters of the beam divergence approximation $N_r = 5$, $N_c = 20$, and $\phi = 0.1146°$ was used. The resulting plot for the rain simulation on a point cloud from the KITTI dataset is shown in Fig. 2b. The raindrop sizes were determined by the Feingold-Levin lognormal distribution [33]. Apart from this distribution one could also use other drop size distributions such as the Marshall-Palmer [34] or the Deirmendjian distribution [35] which we compared in Fig. 3.

2) Snow: From examinations of the DENSE dataset we found the probability distribution function for false point intensities in Section IV. The parameters for this lognormal distribution were:

$$\sigma = 0.649,\ \Theta = 0.105,\ m = 0.204.$$

For the purpose of finding the intensity of the remaining points we compared recordings on roads covered in snow with clear road recordings. The intensity on snowy roads was about 25% higher on average than on clear roads. The thresholds for snow simulation were set to $T_{\text{all}} = 0.6$ and $T_{\text{most}} = 0.2$. The impact of $4\,\text{mm}\,\text{h}^{-1}$ snow is shown in Fig. 2c. In this plot the light snow variant with a size scaling of 2 and an average snowflake mass of $2\,\text{mg}$ was used. The parameters for the beam divergence approximation were the same as for rain.

3) Fog: For parameterizing the fog we used the two metrics presented in Section V to fit our simulation with the recordings of the DENSE dataset. The parameters we obtained from the two metrics are shown below. First the scaling parameter ϵ for the point modification probability was obtained:

$$\epsilon = \begin{cases} 0.23 \cdot e^{-0.0082V} & \text{chamfer metric,} \\ 0.32 \cdot e^{-0.0220V} & \text{distance metric,} \end{cases}$$

where V is the meteorological visibility in m. Second we found the λ parameter for the probability distribution function used to determine the new distance of a modified point:

$$\lambda = \begin{cases} -0.00600 \cdot V + 2.31 & \text{chamfer metric,} \\ -0.00846 \cdot V + 2.29 & \text{distance metric.} \end{cases}$$

Lastly the parameters for the probability to delete points from the point cloud were fitted:

$$(a, b) = \begin{cases} (-0.70, -0.024) & \text{chamfer metric,} \\ (-0.63, -0.020) & \text{distance metric.} \end{cases}$$

The intensity of scan points due to back scattering in the evaluated dataset is below 32% of the maximum intensity for 95.7% of the points, so we set the intensity of false points from 0–32% with a uniform distribution. A point cloud after the fog simulation with $80\,\text{m}$ visibility and the parameters from the chamfer optimization is shown in Fig. 2d.

B. Impact on LiDAR-based object detection

To evaluate the influence of rain, snow, and fog on LiDAR-based object detection the state of the art CNN PointPillars [36] implemented by [38] was used. The network was trained on the KITTI 3D object detection dataset without weather simulations [37]. As dataset KITTI's raw data recordings [10] were chosen. The evaluation was restricted to only those recordings of the KITTI raw dataset which had a duration of at least 30 seconds and contained at least 10 labeled objects. Additionally we used the first 300 frames of the KITTI 3D object detection dataset [37] as it incorporates more pedestrians and cyclists. This allowed us to analyze the impact on different object classes.

Two different metrics were used for performance assesment. The first one was average precision (AP) as presented by Everingham et al. [39]. For a detection to count as true positive we required a bird's eye view overlap threshold of 50%. Furthermore we averaged the AP over all present object classes for the overall evaluation on the KITTI raw dataset resulting in the mean average precision (mAP). We parameterized our weather simulation according to the parameters presented in Section VI-A.

The second evaluation metric we used is the comprehensive safety metric (CSM) presented by Volk et al. [40]. This metric allows a safety assessment of perception systems and is capable of evaluating object detection as well as object tracking. It outputs a single, easy-to-compare number as a safety assessment. Since we want to directly evaluate the influence of adverse weather on object detection methods, the processing pipeline has to be as small as possible. With this in mind, no object tracking was employed and the safety metric was parameterized to only take object detection performance into account.

At first we evaluated the influence of fog, rain, and snow on the detection of cars, cyclists, and pedestrians as a whole on the KITTI raw dataset. We also evaluated the two different parameter optimizations for fog, three different raindrop distributions and two different snow densities. The results are presented in the first two rows in Fig. 3. For mAP as well as for CSM, fog represents the most severe weather condition. The dashed lines represent the baseline detection without simulating any weather conditions.

Rain and snow have shown to affect mAP less compared to fog. Compared to the baseline, for rain the mAP dropped by $32.45\,\text{p.p.}$ with the Deirmendjian raindrop distribution and by $32.39\,\text{p.p.}$ not only with the Feingold-Levin but also with the the Marshall-Palmer distribution. This shows that using different rain distributions only yields in small variations. For snow the detection drop varied from $25.68\,\text{p.p.}$ to $28.11\,\text{p.p.}$ for light and dense snow respectively. The mAP for fog dropped by $37.42\,\text{p.p.}$ for distance optimization and $40.25\,\text{p.p.}$ for chamfer optimization. Even though the resulting detection drop between the chamfer and distance optimization is almost the same, the mAP curve is quite different. The chamfer optimization resulted in bad mAP rates even at higher visibility compared to the distance optimization.

When it comes to safety evaluation, it can be seen that the perception safety is not as affected by different weather conditions as the mAP. The worst CSM drop of the investigated weather simulations was $7.27\,\text{p.p.}$ for fog, $3.34\,\text{p.p.}$ for rain, and $3.99\,\text{p.p.}$ for snow. However, this finding is also due to

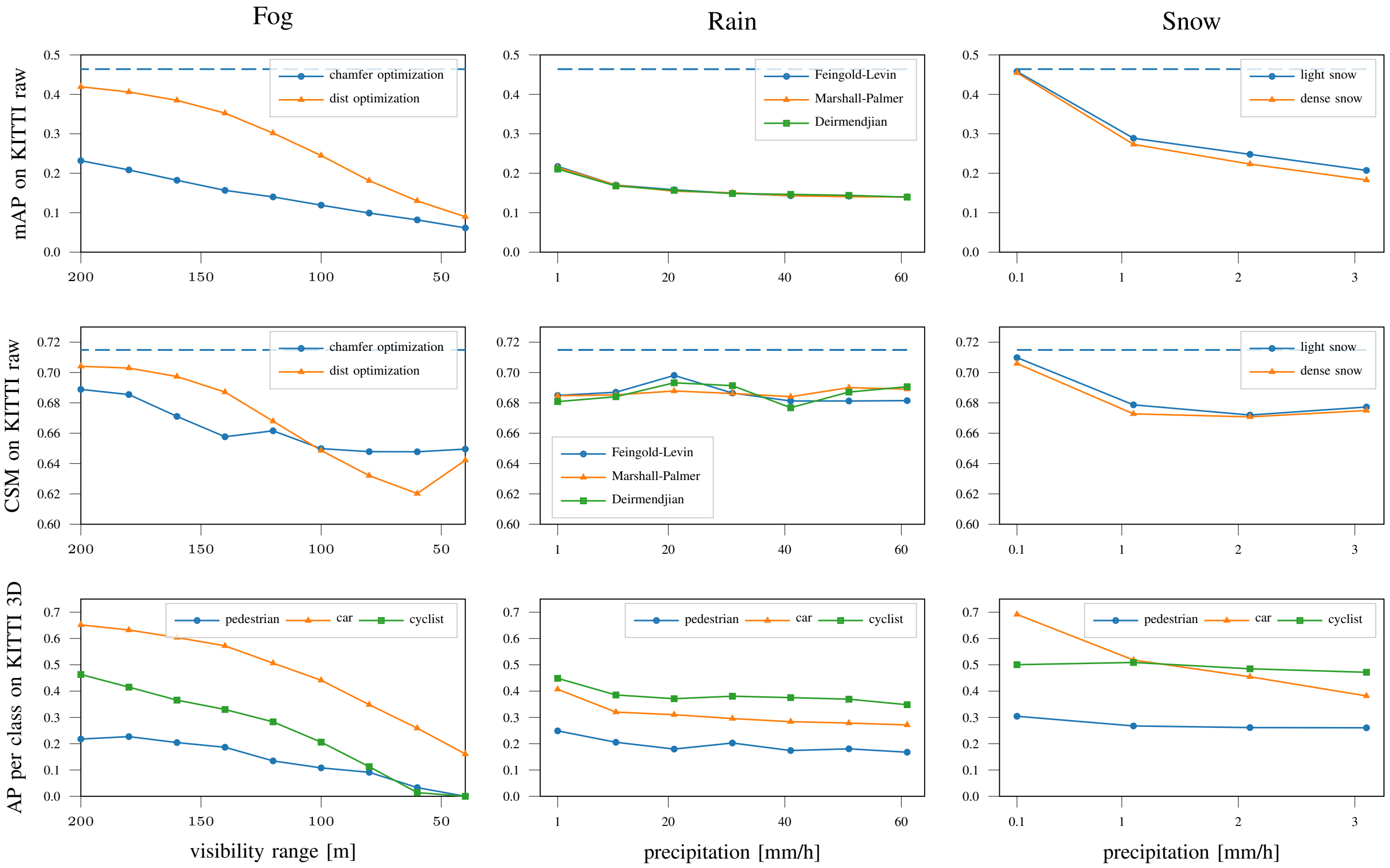


Fig. 3: Impact of fog, rain, and snow on the perception performance of PointPillars [36]. The first two rows evaluate the mAP and CSM metric on the KITTI raw dataset [10]. The dashed line represents the baseline detection. The last row shows the impact on the AP of different object classes on the KITTI 3D object detection dataset [37].

the structure of the KITTI raw dataset. The relatively low ego speed results in short braking times. These short braking times combined with few cutting paths of the ego vehicle and other objects result in almost no safety critical objects. Hence, this dataset is not sufficient to evaluate safety and only allows for an estimation. For the evaluation of the different object classes, we used the KITTI 3D object detection dataset. Furthermore we had to choose a weather parameterization. For fog we used the distance optimization as it does not drop as drastically as the optimization with chamfer distance. For rain we used the Feingold-Levin distribution and for snow we used the dense snow variant. The results of this experiment are presented in the last row of Fig. 3. An observation, not to be overlooked, is that the different object classes get affected differently by the weather variants. Vulnerable road users get detected much worse compared to cars, even without the influence of weather. Analogue to fog representing the most severe weather condition, the detection of cyclists and pedestrians even drops to 0% for fog. Interestingly, for pedestrians slight increases of the AP could be observed, even with increasing rain intensity. Due to the increased noise induced by rain, less false positive objects were detected by the investigated neural network, while maintaining the true positive detections, resulting in a higher AP value. The AP curve for cars follows the mAP rates in the previous evaluation, as seen in Fig. 3, although a different KITTI dataset was used. This is mostly due to cars being the dominant class among the whole KITTI dataset.

VII. CONCLUSION & OUTLOOK

In this work we presented two approaches to realistically simulate weather conditions for LiDAR sensors. The presented approach for rain and snow is based on physical fundamentals realized with ray tracing, while the approach for fog simulation is based on empirical findings. All approaches have been parameterized and optimized according to the DENSE dataset to best fit the effects of real rain, snow, and fog. Firstly we evaluated the influence of adverse weather conditions on the state of the art neural network PointPillars for object detection on LiDAR data. Secondly we showed that rain, snow, and fog drastically reduce the perception capabilities of neural networks which were not trained on a diverse dataset. Lastly we could show that not all object classes are equally affected by adverse weather. Especially vulnerable road users are more prone to not be correctly detected under these environmental conditions.

To summarize, in addition to validating the robustness of existing detection algorithms, the proposed approach may also be used for creating diverse datasets, which include adverse weather conditions, for the training of neural networks.

In conclusion, by providing the ability to generate the desired environmental conditions, our approach helps to develop more robust object detection algorithms.

ACKNOWLEDGMENT

This work has been partially funded by the German Research Foundation (DFG) in the priority program 1835 under grant BR2321/5-2.

We gratefully appreciate the help of Mario Bijelic who supported us with additional data from the DENSE dataset.

REFERENCES

[1] Statistisches Bundesamt, *Verkehrsunfälle 2019*, ser. Verkehr, Jul 2020, no. 8–7, https://www.destatis.de/DE/Themen/Gesellschaft-Umwelt/Verkehrsunfaelle/Publikationen/Downloads-Verkehrsunfaelle/verkehrsunfaelle-jahr-2080700197004.pdf.

[2] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167—-181, Jul 2015.

[3] V. V. Dixit, S. Chand, and D. J. Nair, "Autonomous vehicles: Disengagements, accidents and reaction times," *PLOS ONE*, vol. 11, no. 12, Dec 2016.

[4] N. Rajabli, F. Flammini, R. Nardone, and V. Vittorini, "Software verification and validation of safe autonomous cars: A systematic literature review," *IEEE Access*, vol. 9, pp. 4797—4819, 2021.

[5] 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, pp. 1527–1534.

[6] M. Bijelic, T. Gruber, and W. Ritter, "A benchmark for lidar sensors in fog: Is detection breaking down?" in *IEEE Intelligent Vehicles Symposium (IV)*, 2018, pp. 760–767.

[7] S. Zang, M. Ding, D. Smith, P. Tyler, T. Rakotoarivelo, and M. A. Kaafar, "The Impact of Adverse Weather Conditions on Autonomous Vehicles: How Rain, Snow, Fog, and Hail Affect the Performance of a Self-Driving Car," *IEEE Vehicular Technology Magazine*, vol. 14, no. 2, pp. 103–111, Jun. 2019, conference Name: IEEE Vehicular Technology Magazine.

[8] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Advances in Radio Science: ARS*, vol. 9, p. 49, 2011.

[9] S. Hasirlioglu and A. Riener, "A model-based approach to simulate rain effects on automotive surround sensor data," in *IEEE ITSC*, 2018, pp. 2609–2615.

[10] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets Robotics: The KITTI Dataset," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, pp. 1231 – 1237, 2013.

[11] P. Wang, X. Huang, 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*, 2019.

[12] G. Volk, S. Müller, A. v. Bernuth, D. Hospach, and O. Bringmann, "Towards Robust CNN-based Object Detection through Augmentation with Synthetic Rain Variations," in *IEEE ITSC*, 2019, pp. 285–292.

[13] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[14] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2020.

[15] L. Hespel, N. Riviere, T. Huet, B. Tanguy, and R. Ceolato, "Performance evaluation of laser scanners through the atmosphere with adverse condition," *Proc SPIE*, vol. 8186, Oct. 2011.

[16] A. Filgueira, H. González-Jorge, S. Lagüela, S. Lagüela, L. Díaz-Vilariño, L. Díaz-Vilariño, and P. Arias, "Quantifying the influence of rain in LiDAR performance," vol. 95, pp. 143–148, 2017.

[17] M. Jokela, M. Kutila, and P. Pyykönen, "Testing and validation of automotive point-cloud sensors in adverse weather conditions," *Applied Sciences*, vol. 9, no. 11, p. 2341, 2019.

[18] S. Hasirlioglu, I. Doric, A. Kamann, and A. Riener, "Reproducible Fog Simulation for Testing Automotive Surround Sensors," in *IEEE Vehicular Technology Conference*, Sydney, NSW, Jun. 2017, pp. 1–7.

[19] Y. Li, P. Duthon, M. Colomb, and J. Ibanez-Guzman, "What Happens for a ToF LiDAR in Fog?" *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[20] C. Goodin, D. Carruth, M. Doude, and C. Hudson, "Predicting the Influence of Rain on LIDAR in ADAS," *Electronics*, vol. 8, no. 1, p. 89, Jan. 2019, number: 1 Publisher: Multidisciplinary Digital Publishing Institute.

[21] M. Hadj-Bachir and P. de Souza, "LIDAR sensor simulation in adverse weather condition for driving assistance development," Jan. 2019, working paper or preprint.

[22] E. Villermaux and B. Bossa, "Single-drop fragmentation determines size distribution of raindrops," *Nature Physics*, vol. 5, no. 9, pp. 697–702, 2009.

[23] Y. Sheng, "Quantifying the size of a lidar footprint: A set of generalized equations," *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 3, pp. 419–422, 2008.

[24] K. Gunn and J. Marshall, "The distribution with size of aggregate snowflakes," *Journal of Atmospheric Sciences*, vol. 15, no. 5, pp. 452–461, 1958.

[25] R. Sekhon and R. Srivastava, "Snow size spectra and radar reflectivity," *Journal of the Atmospheric Sciences*, vol. 27, no. 2, pp. 299–307, 1970.

[26] T. Oguchi, "Electromagnetic wave propagation and scattering in rain and other hydrometeors," *Proceedings of the IEEE*, vol. 71, no. 9, pp. 1029–1078, 1983.

[27] G. Koh, J. Lacombe, and D. L. Hutt, "Snow mass concentration and precipitation rate," *Cold Regions Science and Technology*, vol. 15, no. 1, pp. 89–92, 1988.

[28] A. Von Bernuth, G. Volk, and O. Bringmann, "Simulating photo-realistic snow and fog on existing images for enhanced cnn training and evaluation," in *IEEE ITSC*, 2019, pp. 41–46.

[29] S. Hasirlioglu and A. Riener, "Introduction to rain and fog attenuation on automotive surround sensors," in *IEEE ITSC*, 2017, pp. 1–7.

[30] P. Duthon, M. Colomb, and F. Bernardin, "Light transmission in fog: the influence of wavelength on the extinction coefficient," *Applied Sciences*, vol. 9, no. 14, p. 2843, 2019.

[31] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[32] S. Vallender, "Calculation of the wasserstein distance between probability distributions on the line," *Theory of Probability & Its Applications*, vol. 18, no. 4, pp. 784–786, 1974.

[33] G. Feingold and Z. Levin, "The lognormal fit to raindrop spectra from frontal convective clouds in israel," *Journal of climate and applied meteorology*, vol. 25, no. 10, pp. 1346–1363, 1986.

[34] J. S. Marshall and W. M. K. Palmer, "The distribution of raindrops with size," *Journal of meteorology*, vol. 5, no. 4, pp. 165–166, 1948.

[35] D. Deirmendjian, "Electromagnetic scattering on spherical polydispersions," Rand Corp Santa Monica CA, Tech. Rep., 1969.

[36] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection From Point Clouds," in *2019 IEEE/CVF CVPR*, Jun. 2019, pp. 12 689–12 697.

[37] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[38] O. D. Team, "Openpcdet: An open-source toolbox for 3d object detection from point clouds," https://github.com/open-mmlab/OpenPCDet, 2020.

[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[40] G. Volk, J. Gamerdinger, A. v. Bernuth, and O. Bringmann, "A Comprehensive Safety Metric to Evaluate Perception in Autonomous Systems," in *IEEE ITSC*, 2020, pp. 1–8.