공학박사학위논문

악천후상황 자율주행 인지 시스템을 위한 LiDAR 포인트별 객체 검출 기반 잡음 제거 및 유효인지거리 추정

LiDAR Point-wise Segmentation based De-noising and Detectible-range Estimation for Autonomous driving Perception system in Adverse weather

2022년 8월

서울대학교 대학원

기계공학부

이 종 민

악천후상황 자율주행 인지 시스템을 위한 LiDAR 포인트별 객체 검출 기반 잡음 제거 및 유효인지거리 추정

LiDAR Point-wise Segmentation based De-noising and Detectible-range Estimation for Autonomous driving Perception system in Adverse weather

지도교수 이 경 수

이 논문을 공학박사 학위논문으로 제출함

2022 년 4 월

서울대학교 대학원

기계공학부

이 종 민

이종민의 공학박사 학위논문을 인준함

2022 년 6 월

위 원 장 : 박 종 우

부위원장 : 이 경 수

위 원 : 김 아 영

위 원 : 오 광 석

위 원 : 신 동 훈

Abstract

LiDAR Point-wise Segmentation based De-noising and Detectible-range Estimation for Autonomous driving Perception system in Adverse weather

Jongmin Lee

School of Mechanical Engineering

The Graduate School

Seoul National University

Recently, Numerous studies have been conducted on the perception part of automated vehicle research. The LiDAR (Light Detection And Ranging) sensor is a fundamental sensor for Automated vehicles and has been used in many prior studies. LiDAR sensors prove to be advantageous in providing sufficient and precise distance data, complementing cameras, radars, and ultrasonic sensors. They have been widely used in fields such as autonomous driving. However, adverse weather conditions such as rain, fog, or snow have been shown to affect LiDAR data significantly. Many studies have been conducted only for perception algorithms using LiDAR. However, studies on the degradation of LiDAR in adverse weather conditions are still considered to be lacking. In such conditions, particulates in the air such as rain, snow, and fog droplets have been shown to reflect the lasers emitted by the LiDAR sensor, resulting in these particulates being misinterpreted as objects. When such noisy data is given to a perception algorithm, performance degradation of the algorithm occurs.

Therefore, this dissertation focused on developing a point-wise de-noising algorithm based on semantic segmentation. In this proposed de-noising deep neural network, feature aggregation is performed to minimize any information loss and misclassification errors while progressively increasing the receptive field. In front of the de-noising module, a module uses LiDAR Point Cloud Data (PCD) to determine the weather condition and inform the current weather probabilistically to the de-noising module. Moreover, after the de-noising module, points classified as fog are then used by the latter module to quantify fog density through the Occupancy Grid Map (OGM) method. The maximum detection range at the current state is derived within and used as the output of the module, utilizing the correlation between the Maximum Detection Range and the OGM pixel's probability.

Adverse weather data especially fog driving data with LiDAR Point Cloud Data (PCD), is rare. For this dissertation, PCD data in real-life driving in heavy fog were acquired and used in the study. Some research teams obtained LiDAR PCD set from a CEREMA's climate chamber system. However, only data could be acquired in this controlled weather environment with the vehicle stopped. Therefore, in this study, not only has the proposed approach been experimented on and validated through a large dataset obtained in controlled weather environments, but also point cloud data directly obtained by our research team while driving in real-life dense fog scenarios. The suggested module successfully segments and filter out the adverse weather-affected point cloud data, providing safety guaranteed perception information for autonomous driving.

Keywords: Autonomous driving, De-noising, Semantic Segmentation, Adverse weather, Sensor degradation, Fog weather, Point cloud density

Student Number: 2018-39006

List of Figures

Contents

Chapter 1 Introduction

1.1 Background and Motivation

The perception module in autonomous driving system is a key task and a significant factor limiting the availability and performance of the system. Lidar effectively provides high-resolution distance information that cameras, radars, ultrasonic sensors cannot provide, showing excellent performance in recognizing surrounding objects even at night. Developing a fully automated vehicle needs to know the AV system's boundaries without the driver's intervention and drive properly to react to all impairments while driving. According to the definition of levels of automation by the 'Society of Automotive Engineers (SAE), A system with autonomous driving levels 3 and 4 should be found for system failures, and control should be handed over to the driver, or the system should judge and drive safely. [SAE'10].

Lidar effectively provides high-resolution distance information that cameras, radars, ultrasonic sensors cannot provide, showing excellent performance in recognizing surrounding objects even at night. However, due to the operating characteristics of the lidar, the distance information could be distorted under adverse weather conditions because the emitted lasers are reflected on the particles such as raindrops, snow, and fog distributed in the air [Bijelic'18, Charron'18, Djuricic'13, Heinzler'19, Kutila'18, Phillips'17, Yondeda'19]. Studies have been conducted on the measurement results of LiDAR for weather conditions. Autonomous vehicles should be able to drive safely under adverse weather conditions such as heavy snow, heavy fog, and rain. Therefore, the experimental data provided in this situation contributes significantly to algorithm development. In [Cho'14] study, sensor data of autonomous vehicles were acquired in dust, smoke, and rain conditions, and it was observed heavy dust particles in the air were detected by LiDAR and did not reach objects behind dust point cloud. In [Brostow'08] study, Opal 3D Lidar was utilized to examine the penetration performance of dust, fog, white snow, and smoke. Dust and fog detection performance was tested in chamber. The snow and smoke test were conducted in the field. In [Dollar'12], disabilities caused by low external temperatures and exhaust gas were studied. Since quantification of fog density is not possible in a natural environment, [Dollar'12] study tested and concluded in a dedicated indoor environment with a closed-loop control system [Eigen'14]. The effect by rain on LiDAR was examined in [Filgueira'17], [Ryde'09]. In [Filgueira'17], the study presented the task of quantifying the effect of heavy rain on LiDAR in a static scene. It describes the average range, intensity, and several points for a specific object. The effects of dust are analyzed with 2D laser scanner [Ryde'09]. [Phillips'17]. Smoke and rain are further investigated in [Ryde'09], but there is no investigation into the effect of the fog because the utilized chamber could not produce artificial fog.

The performance of the lidar sensor varies greatly depending on environmental conditions. Therefore, to develop a strong perception and the resulting fusion algorithm. Identifying and quantifying the weather's impact on lidar performance is essential. Furthermore, it is clear that the adverse weather effects on the Lidar system are significant [Heinzler'19], [Filgueira'17], [Naboulsi'04], [Hasirlioglu'17], [Ryde'09], [Phillips'17], [Dollar'12], [Eigen'14], [Brostow'08], [Cho'14], [Colomb'04], [Dolgov'10], [Dollar'12]. The video supplementation data conducted in this study describes in detail what type of points LiDAR shows in heavy fog[Lee'22(a), Lee'22(c), Lee'22(b)].

For a perception system using point clouds, such distorted distance information is undesirable noise, which restrains the ability to understand the surrounding scene and results in the degradation of the object detection performance. Mainly, these issues are fatal for lidar perception algorithms involving classical bottom-up approaches. Although many studies have been conducted only for perception algorithms using LiDAR, studies on the degradation of LiDAR in adverse weather conditions are still considered lacking. Thus, the de-noising module was developed in adverse weather situations.

In the past few years, learning-based perception methods have achieved outstanding progress about object detection. By using shape characteristics to recognize objects, learning-based algorithms tend to cope better with noisy data. Nevertheless, these algorithms still remain susceptible to misinterpreting noise points as objects, giving a false positive within the object detection module. The most intuitive and ideal solution is to classify and filter out only these noise points from the scene which can be achieved through a semantic segmentation approach. Numerous works have shown considerable performance on object detection using semantic segmentation, by projecting 3D PCD onto 2D pixel-wise images or voxelization into 3D grids [Graham'18, Chen'19]. However, such methods may result in the loss of geometric details during the processing stage. For example, valid points within the same voxel or grid with noise points are classified as noise points during the semantic segmentation process. These misclassified points are then removed during the de-noising process, resulting in the degradation of overall object detection performance.

1.2 Previous Researches

1.2.1 De-noising methodologies & Semantic Segmentation

A number of studies have been studied for the development of an automated vehicle perception system. Through many previous studies, it was found that adverse weather caused the performance degradation of LiDAR. However, research on responding to performance degradation is still considered insufficient. The studies related to this research can be largely divided into two methodologies. The first is the filter-based methodology, and the second is the Artificial Neural Network-based method.

1) Filter-based de-noising

It was mainly used in camera image data processing with depth information. Regular smoothing filters [Tomasi’98], media filters [Gokturk’04], and bilateral filters [Li’14] are used to filter snow data in 2d lidar [Ronnback’08]. Filters have been proposed to solve problems such as edge loss. A filter for an image was performed using Joint Bilter filter (JBF) [Chaudhary’16], [Le’14], NL-Means [Huhle’10], and [Huhle’08] developed based on a Bilter filter. There has been continuous research on a fatal impairing edge of the 2d depth image, and a method of locally smoothing through segmentation of one image has been proposed [Chen’12]. After that, several types of bilateral filters were applied.

There is a method of filtering in the 3d geometric domain. Although these methodologies are computationally not cheap and challenging to implement in edge hardware, however, 3d raw data could be directly utilized without any pre-processing. The maximum likelihood estimation (MLE) method [Schall'05] made the noisy point cloud smooth and statistical using Bayesian statistics. However, this method is computationally expensive, so real-time operation is impossible. A methodology using two points geographically close by applying a method similar to the bilateral filter described above to the 3d point cloud has been proposed [Schall'08]. Among the methods of directly processing 3D points, reducing the number of points by dividing 3d space into a cube form has been developed. [Connolly'84, Martin'97]. However, the points left in the voxel are not robust to noise points, and they are not represented by the underestimating surface accumulation. A method of determining the filter's Outlier using the number of surrounding points has also been proposed. The method has the advantage of being computationally inexpensive [Rusu'11]. Among the methods of determining the Outlier, the mean and standard deviation of the distance based filtering method was proposed [Hu'13]. The threshold is determined based on the k-nearest neighbor distance. The threshold is calculated as the global mean, standard deviation and filtered based on the reference value [Rusu'11]. In addition to setting the distance as an index, methods such as using the size of the neighborhood or fitting the line of neighborhood points were proposed. However, the more complex the method is the more computational expense increases, making it difficult to guarantee real-time [Weyrich'04]. Similarly, the Radius Outlier removal filter measures the number of neighbors within a predefined radius in the SOR filter. Decide whether to remove or calculate based on a specific value [Rusu’11]. Based on this methodology, [Charron’18] introduces a dynamic radius outlier removal filter. DROR is a dynamically considered filter. These methods also have the disadvantage of erasing meaningful point clouds. Moreover, the test has not yet been conducted at the vehicle level. DROR is a dynamically considered filter. These methods also have the disadvantage of erasing meaningful point clouds. Moreover, the test has not yet been conducted at the vehicle level.

2) Network-based De-noising

If we think about De-noising easily, De-noising is a method of labeling all points, removing points classified as fogs, and leaving points classified as valid points. Therefore, most of the network-based de-noising methods could be belong to the semantic segmentation category.

Heinzler et al[Heinzler'20]. proposed a CNN-based De-noising network in adverse weather. Raw data processing performs 3D PCD to 2d projection. Moreover, it directly applies the CNN method by applying the 2d pixel-wise method. Dialed convolution is performed by applying LiLaBlock. This paper proposes WeatherNet[Heinzler'20] based on LiLaNet, and network depth has the advantage of decreasing. During the test, validation is performed on three data sets by mixing the chamber and rod datasets. A disadvantage of the method is that in the 3d PCD projection step, it inevitably introduces information loss. Furthermore, the disadvantage is that geometrical and structural information cannot be fully utilized. This paper quantitatively compared the DROR [Charron’18] method using the heuristic method. Bijelic et al[Bijelic’20].

firstly proposed a multimodal sensor fusion-based network available in adverse weather situations. In this paper, a new fog model was proposed. The proposed mathematical fog model was utilized in many studies for quantitative comparison. They proposed a multimodal adverse weather dataset with lidar, radar, RGB camera, Gated camera, and FIR camera sensors. The dataset has rare adverse weather scenarios, such as heavy fog situations, heavy snow, heavy rain, etc. Its dataset contains more than 10,000km of driving in Europe. From the point of view of the de-noising part, the 'chamber dataset' has excellent research value for de-noising research. Furthermore, many studies conducted quantitative evaluations using the corresponding dataset. The WeatherNet [Heinzler'20] mentioned above also used the corresponding chamber dataset.

3) Semantic Segmentation

As mentioned above, de-noising networks mostly use semantic segmentation methods. Therefore, below, previous studies investigated point-based semantic segmentation. Although it is a point-based semantic segmentation, it also describes how points can be represented in different formats in many networks. We thought about how to learn and segment a droplet with these point characteristics. It was found that a network-based method should be applied to segmentation by processing a large amount of data in real-time. Deep learning-based point cloud processing has become increasingly popular with various methods to solve different problems in numerous research areas.

Semantic segmentation is also divided into three parts: instance segmentation, part segmentation, and semantic segmentation. Semantic segmentation can be divided into **projection-based, point-based, discretization-based, and hybrid methods**. The projection-based and discretization-based methods transform 3d PCD into other forms through pre-processing. The papers referenced for this dissertation are explained below [Guo'20]. The final network structure could be constructed through the previous research below.

Projection-based Methods

These methods mainly transform 3D shaped point clouds on pixel-wise 2D images. These methods project an unstructured and unordered 3D PCD into regular representation. For example, multi-view images, spherical images, voxel grids, volumetric, hybrid representations, and lattice representations can leverage the well-structured 2d or 3d convolution with this representation. Finally, the pixel-wise or voxel semantic labels are projected back to the raw point clouds.

Lawin et al. [Lawin'17] first proposed projection of a 3d point into a 2d plane with multiple virtual camera views. To predict the pixel-wise score, the multi-stream FCN was utilized. By fusing the re-projected scores from different views, the point-wise labeling result was obtained. Tatarchenko et al. [Tatarchenko'18] proposed the assumption that the point cloud is at the local Euclidean surface. They proposed tangent-based convolution segmentation for densely distributed point cloud. The base concept of this method is project every points on tangent plane. Then the tangential convolution works directly on the surface geometry. This method could handle large scale point clouds. Using multiple camera positions, Boulch et al first generate multiple RGB and depth snapshots of point clouds [Boulch'17]. Then utilized 2D convolutional segmentation network to perform pixel-wise labeling. The inference values from RGB and depth images are fused using residual modification [Audebert'16].

Overall, the multi-view segmentation method is greatly influenced by viewpoint selection and occlusion. On the other hand, these methods could not fully use geometric information. Therefore, it unavoidably leads to an information drop. In the spherical representation part, to segment 3D PCD faster and accurately, Wu et al [Wu'18] introduced an SqueezeNet [Iandola'16] and Conditional Random Field (CRF). SqueezeSegV2 [Wu'19] was proposed to enhance segmentation accuracy. This network also used an unsupervised domain adaptation pipeline to solve the domain shift problem. Milioto et al [Milioto'19] proposed RangeNet++ for semantic segmentation of LiDAR PCD in real-time. The segmented labels in 2D image transfer to the 3D point cloud. The computationally cheap KNN algorithm-based post processing also performed. It could mitigate dimension change(2D to 3D) errors. The spherical projection-based method possesses more information than single projection-based algorithms. Furthermore, it is more suitable for labeling PCD. However, these representation changes could lead to some problems, such as geometric information loss and discretization errors.

Discretization-based Methods

In the Discretize-based Strategies part, these methods usually transform a PCD into a discrete representation, such as permutohedral lattices and volumetric representation. The initial method voxelizes the point data into a grid form. The conventional 3d convolution was performed. Huang et al.[Huang'16 changed it to occupancy voxel and provided these voxel data information to 3D CNN for segmentation. All voxel grids are segmented by this method. Meng et al.[Meng'19] presented an interpolation variational encoder network structure to extract geometrical information locally per voxel. This model gets a serial representation and not a discrete one. It also saves the distribution of each voxel. They used correspondence CNN to perform powerful feature learning. Tchapmi et al.[Tchapmi'17] suggested that SEG Cloud to supply well grained and poses a globally uniform segmentation network. This approach presents decisive cubic interpolation. It could map the coarse voxel forecast designed by 3D-FCNN [Long'15] shapes back to the point cloud. The Fully Connected CRF has been utilized to enhance the consistency in the space of these deduced point-wise labels. Volumic networks could be trained very openly and validated with different sizes with 3D CNN. In FCPN [Rethage'18], The geometric relations of PCD are abstracted lightly, then the 3D Conv and pooling method extract range dependence and feature information. This method has excellent scalability and could manage large-scale PCD. Dai et al.[Dai'18] presented 3D scan-based voxel-wise segmentation method. It has excellent scalability that could be fed various input-sized data. Since volumetric repression preserves the structure of 3D points, it has the advantage of preserving structural information. Moreover, 3D convolution could be applied directly. However, this method is always accompanied by a computationally expensive problem. The voxelization method includes a problem of information loss during the discretization process and a problem of determining the size of the grid. There is a data sparsity problem with 3d PCD. Non-zero values account for a very small percentage. Thus, applying a deep CNN network to spare data is unsuitable. Choy et al. [Choy'19] proposed a 4D CNN named MinkowskiNet for 3D video recognition. A sparse CNN is used to solve the problem of high-dimensional data efficiently.

Point-based Methods

A network based on a point cloud should consider the characteristics of 3d PCD. The properties of 3D PCD are irregular and unstructured. Thus, it is impossible to use CNN networks. A pioneering work, PointNet [Qi'17] has presented a method that utilizes shared-MLP and symmetric pooling functions to learn point-wise features. Many studies have been derived based on PointNet [Qi'17]. Classifying the methods in [Guo'20], it can be divided into point-wise MLP, point convolution, RNN-based methods, and graph-based methods. In Point-wise MLP methods, The advantage is to increase sMLP efficiency using the essential tool of the network. The disadvantage is that it is impossible to consider the correlation between points that could lose local geometrical information. Point cloud information on a local area is essential, and the degree of distribution of points is also significant. Thus, many methodologies have been proposed for richer learning of local structures. Feature pooling, Attention-based aggregation, and Local-global feature concatenation are the various methodologies that have been suggested in point-wise methods for feature learning.

In Feature pooling, pioneering work PointNet++[Qi'17] uses the sampling method to pack points and learn features from a more extensive scope to a smaller area. To learn local point structure patterns, Learn local features by gradually expanding the area from a small area's point. Scale and resolution's great scalability has solved the non-uniform and varying density problems of 3D PCD. Hu et al.[Hu'20] presented a productive and lightweight network for large point cloud segmentation called RandLA-Net. The novelty of this network is to use the random sampling method to extract large-scale point clouds, also larger the receptive field of a point feature. This network has an encoding and decoding network structure and skip connection to share feature points in the network layer. The feature aggregator module could preserve local feature information. For the decoding process, it used KNN to upsample the extracted points. Engelmann et al. et al.[Engelmann'18] also used KNN and kmean clusters to define feature maps and geometric maps separately. Jiang et al. [Jiang'18] suggested that the PointSIFT network could orientationally feature learning. Eight directions and orientations were encoded through convolution work. Zhao et al.[Zhao'19] designed PointWeb module explore all points locally and build an interconnected web. Feature Adjustment tool could interact with information and improve features. Zhang et al.[Zhang'19] developed ShellConv based on the spherical shell concept. The pooling module was used to gather information in various shells.

In Attention-based aggregation, Yang et al.[Yang'19] suggested modeling point-wise connection and a newly proposed sampling method named GSS. Gumbel Subset Sampling, which is different from the FPS sampling method offered by PointNet[Qi’17]. Zhao et al.[Zhao’19] proposes the post-process step called ASR, which could refine the score of the segmentation result. It has good expandability to merge with the existing network. Chen et al.[Chen’19] proposed 3D PCD's spatial distribution capturing oriented 'spatial awareness' weights-based network. It could easily learn local PCD's structure and spatial information by LSA layer (Local Spatial Aware).

Point Convolution Method

The Point Convolution offers productive convolution for PCD. In [Engelmann’20], Engelmann et al. proposed Dilated Point Convolution (DPC) to aggregate local features broadly instead of KNN. This operator could enlarge the receptive field and also effectively merge with the aggregation module. Engelmann et al. accomplished numerous ablation studies to show the performance of DPC. Wang et al. [Wang’18] proposed a Deep Parametric Continuous Convolutional Neural Network (PCCN) based on parametric constant convolution. The kernel part of this layer is parameterized by the multilayer perceptron and transits constant vector space. Hua et al. [Hua’18] also proposed a point-wise convolution operator. The distinguished point was that its neighboring points are attached to the kernel and then a product with the weight of kernels.

RNN-based Methods.

Ye et al. [Ye’18] presented a Point-wise Pyramid Pooling (3P) module to seize local structure. It used two-direction hierarchical RNN to obtain long-range spatial dependences additionally. The end-to-end learning was performed with RNN. These methods could fail to get geometric features of PCD when aggregating the local features. To help the issues caused by the fixed pooling, Zhao et al. [Zhao’19] suggested a Dynamic Aggregation Network that considers global and local features simultaneously. The medium features from local to global are aggregated using an adjusted receptive field and node weights. Based on PointNet [Qi’17], Engelmann et al. [Engelmann’17] converted points into multi-scale and grid blocks. The block-wise features have been extracted by the PointNet module. Also, it is consecutively provided to Consolidation Units to get output context. Liu et al. [Liu’17] proposed 3D point cloud CNN to DQN and RNN for productive semantic parsing of large-scale point clouds. It learns the 3D PCD's spatial information and RGB features using 3D CNN. The DQN is later used to classify the specific classes. The concatenated feature matrix is used to utilize in RNN to get segmentation results.

Graph-based Methods.

Various methods are studied to capture the local geometric structures of 3d point cloud data. Wang et al. [Wang’19] used Graph Attention Convolution (GAC) to learn the spatial local features of PCD selectively. The attention weights are assigned to local points and features vector according to PCD's spatial information and feature distinctions. In [Landrieu’18], Landrieu et al. believe the PCD is simple interconnected forms and super points. Keep local and global form information using a super point graph. Therefore, PCD segmentation is divided into 3 problems. i.e., geometrically homogeneous divisions, super-point decisions, and context segmentation. Ma et al. [Ma’20] proposed a Point Global Context Reasoning (PointGCR) module to learn global PCD context data with channel dimensions using a graph representation. As mentioned above, the comparative semantic segmentation previous study was referred to [Guo’20] paper.

1.2.2 Adverse weather data in Autonomous Driving

LiDAR and camera data in bad weather conditions are very rare. Many studies lack sharing of raw data. However, the most famous dataset is open source. Through that dataset, many studies on adverse weather have been conducted. A data set is acquired in the weather-controlled chamber when the vehicle or sensor is in a stopped condition. Bijelic et al. proposed the most famous dataset and has been cited in the most relevant research papers [Bijelic'20]. The reason why many studies prefer chamber data sets is because of the problem of point-wise labeling. In the chamber where all objects are stopped, there is no need to put much effort into labeling because the ground truth data of the object is known by users. For example, comparing the data in clear weather and fog and rain with multiple objects in a motionless chamber could easily solve the labeling problem. For that reason, many studies solve the labeling problem by acquiring chamber data. However, the downside is that no team has ever acquired and studied sensors data in adverse weather in situations where autonomous vehicles are in motion.

In the following subsection, there is a detailed description of the advertise weather data proposed by the prior study. And there is an explanation for advertise weather in the driving situation acquired in this study.

Adverse weather data of prior studies. Currently, available datasets about adverse weather are the only [Bijelic'20, Heinzler'20] datasets mentioned above. Other datasets are datasets that have not been disclosed or cannot be used for research due to limited amounts of data.

In [Heinzler'20] dataset, It analyzed the effect of adverse weather on LiDAR data. In [Bijelic'20, Heinzler'20] study, they acquired LiDAR data within the weather chamber to derive results. It shows the results as figure 1.1.

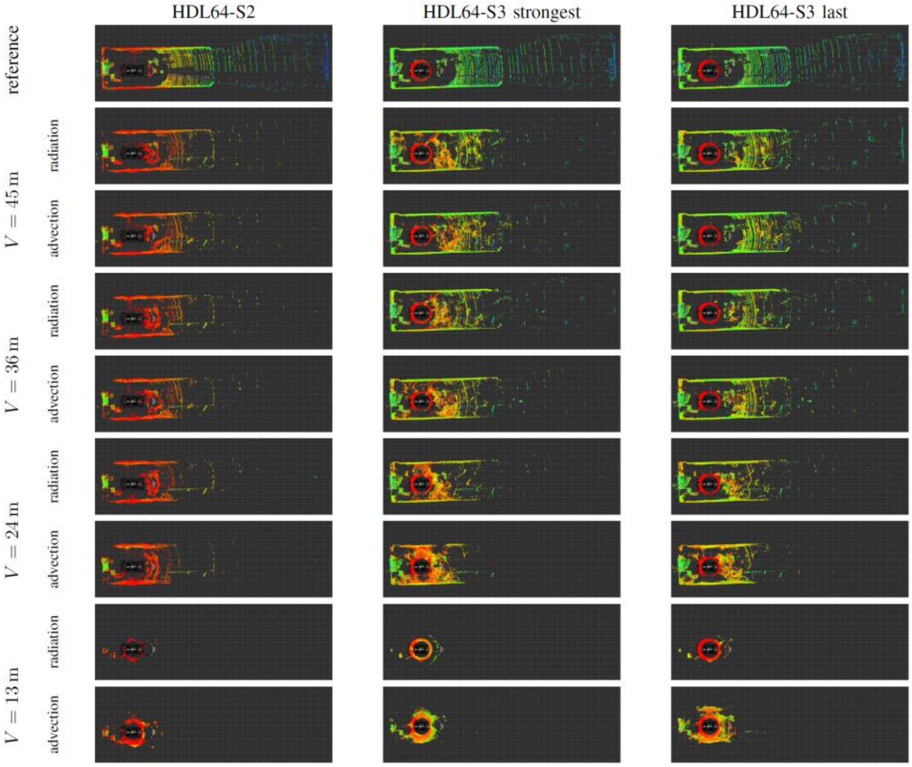
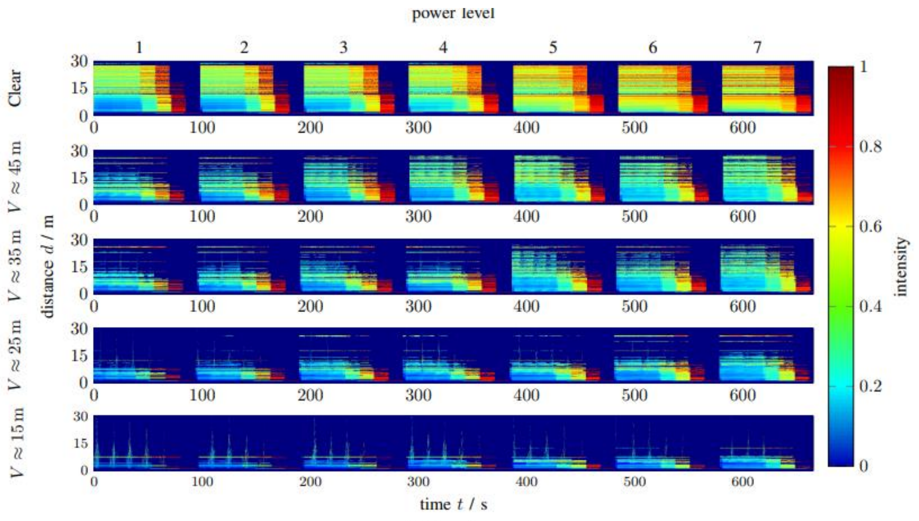


Figure 1.1. Bird view of Lidar sensors for different fog types and fog densities [Heinzler'20]

In figure 1.1, Lidar represents three modes on the longitudinal axis. And V represents the visual distance. This figure shows the shape and intensity of the LiDAR point cloud according to the perceptible distance through the adjustment of fog density. The higher the fog density, the lower the intensity, the lower the circular fog ring around the lidar, and the perception distance decreases. In this thesis, to take the advantage of labeling, all data were obtained when a sensor or vehicle was stopped.

In figure 1.2, the power level and intensity decreases with the fog density.



(b) Intensity

Figure 1.2 Changes in power level and intensity according to fog density[Heinzler'20]

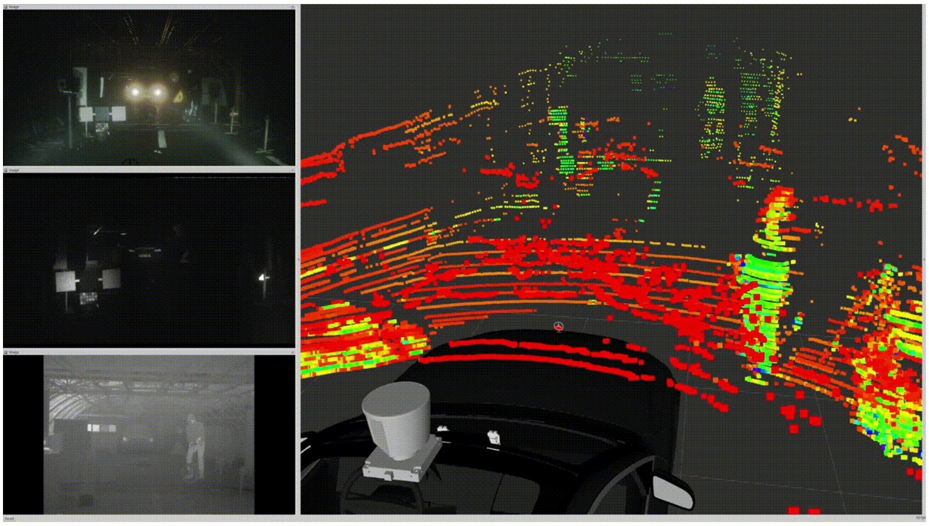


Figure 1.3 Data acquiring scenes in Dense fog [Heinzler'20]

[Carballo'20] paper tested 10 lidars in JARI's weather chamber and compared the dataset. In this thesis, only the results of the accuracy of the cognitive distance according to the density of fog were derived. The fog shape showed a fog ring similar to previous studies. The study also derived results in an unmoving state. The brief test environment and results are shown as figure 1.4.

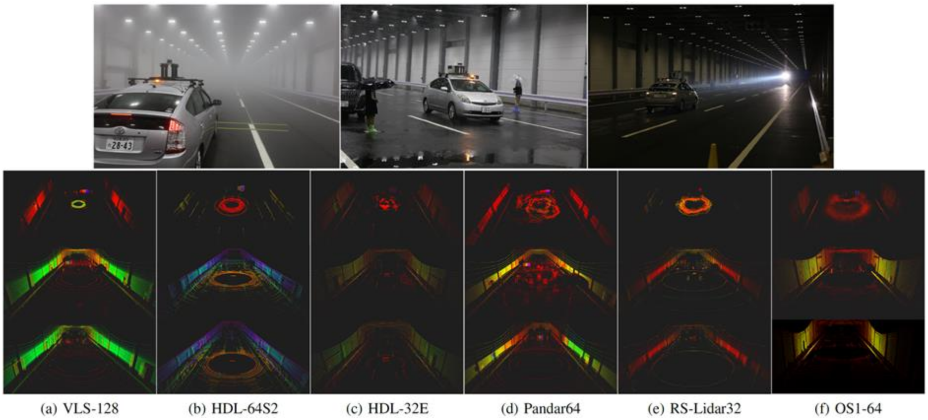


Figure 1.4. Test environment and result of LiDAR shape in dense fog [Carballo'20]

Adverse weather data in real-life driving conditions. Most of the studies related to adverse weather acquire and process sensor data while the vehicle is stopped. However, it is essential to observe changes in sensor raw data due to sensor gradation while driving. The perception system of autonomous vehicles aims to detect objects such as vehicles and people around them while the vehicle is driving. According to the experience of acquiring data directly, it was found that the sensor results during driving and stopping in a heavy fog situation differ a lot, and the characteristics of the fog are also different. The driving fog point cloud characteristics are better seen in video data. The captured picture of the video is described in figures 1.5 to 1.9.

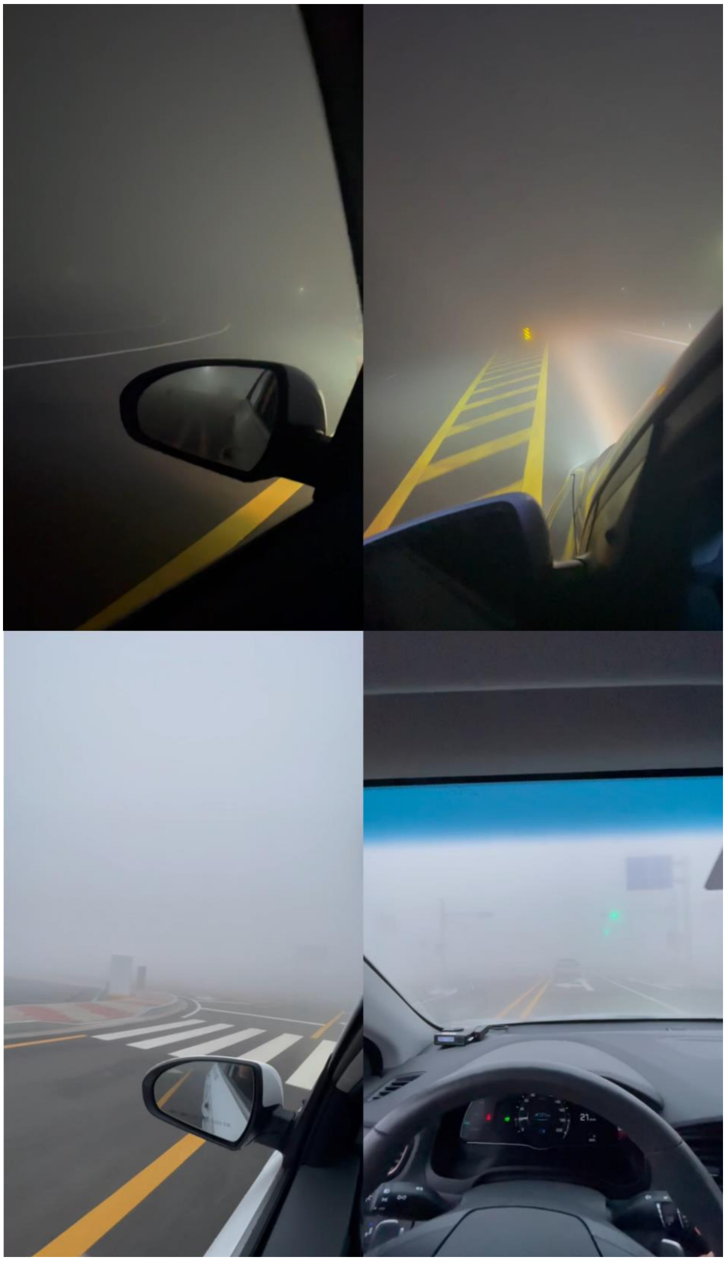


Figure 1.5. Dense Fog Video capture (Driving visibility is less than 20 to 35 meters, Korea Meteorological Administration officially announced on November 20, 2021) [Lee’22(d)]

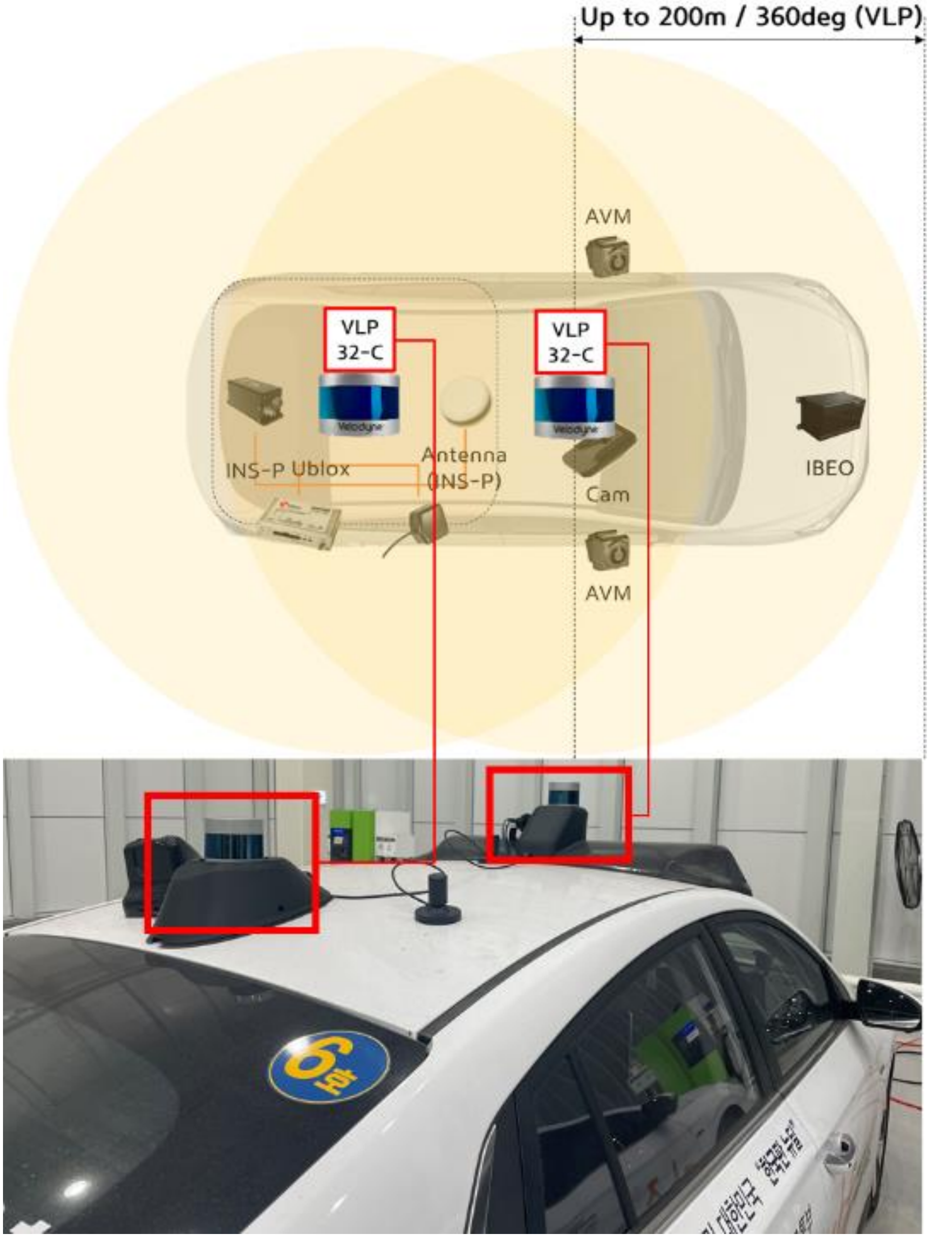


Figure 1.6. Sensor configuration

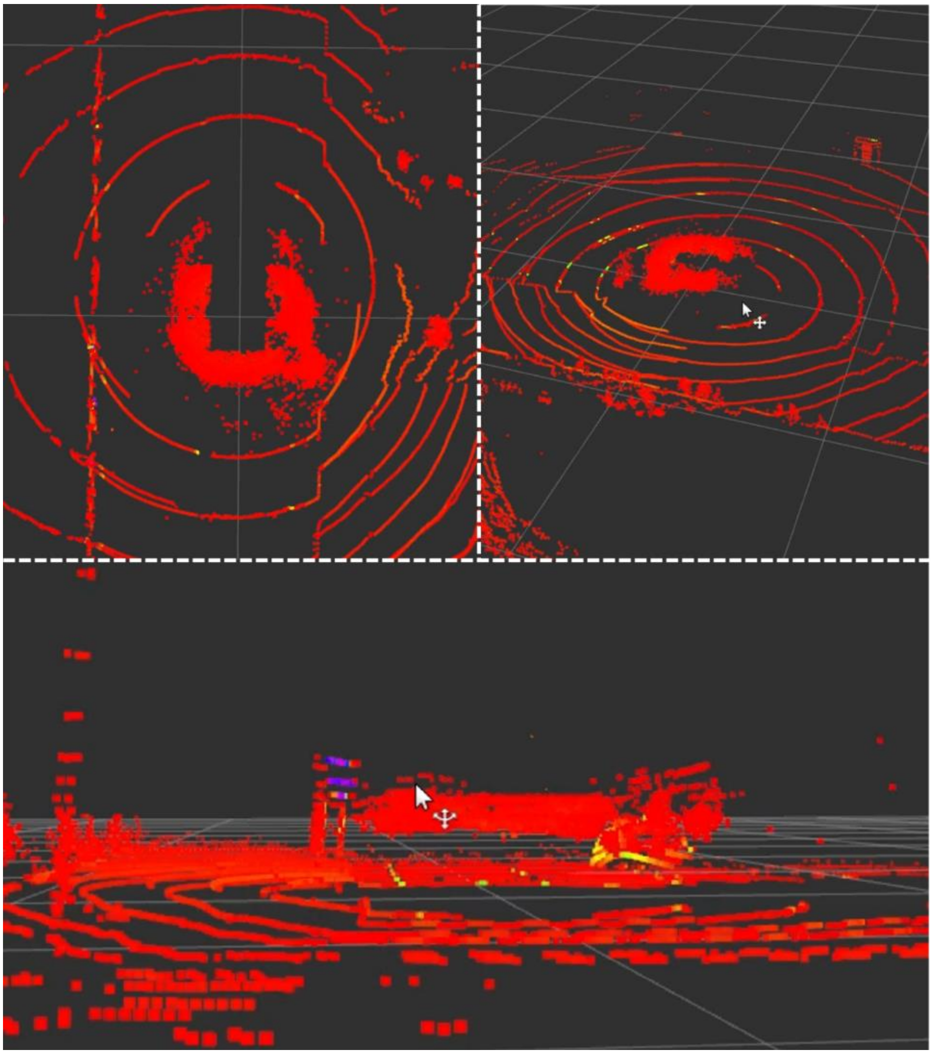


Figure 1.7. Fog shape view with three different views, Top view(left), 45-degree view (right), Front view (down), Point cloud colour setting: intensity-based configuration (ROS rviz rainbow setting, intensity value is grayscale red ≈ 0, purple ≈ 255) (Grey colour Grid size is 10m. When data is acquired, PCD(Point Cloud Data) as much as the vehicle size is deleted, so there is no data in a rectangular shape; however, it reflects the characteristics of fog)

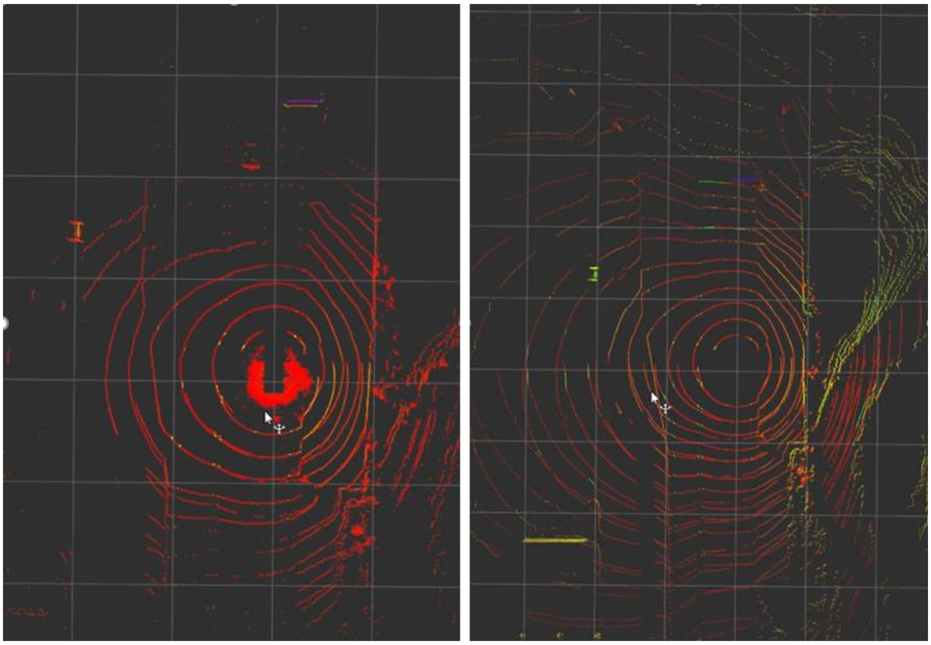


Figure 1.8. Comparison capture of LiDAR raw data acquired in the same vehicle, same place, and same time zone.

Although it is the same place and the same time zone, it can be seen that the LiDAR point’s reach distance and total intensity are very different. As shown in Figure 1.9, Heavy snow shows an unusual appearance of scattering.

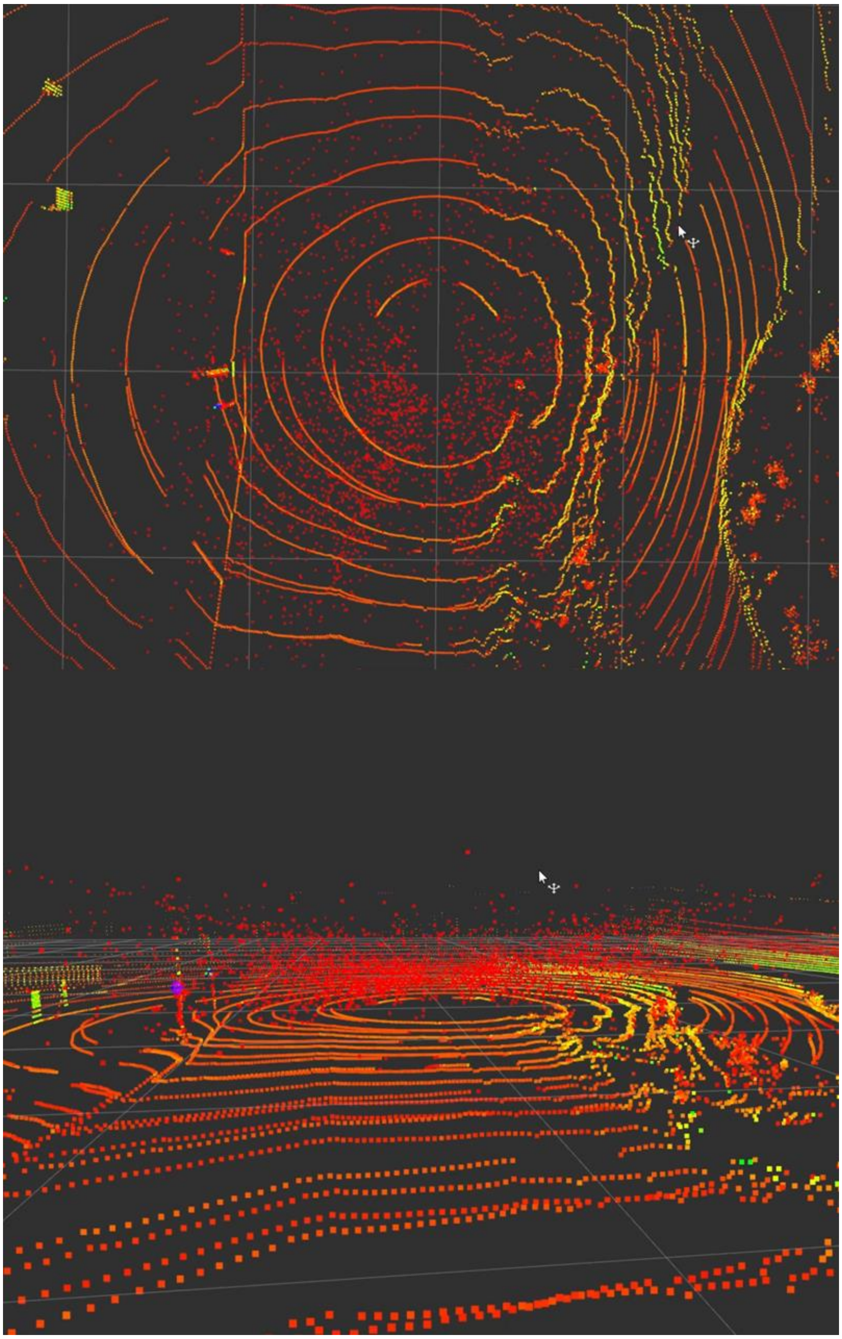


Figure 1.9. Heavy snow view with two different view

1.3 Thesis Objectives

From a considerable amount of literature, adverse weather causes performance degradation of the LiDAR sensor. Moreover, it was found through experiments that mis/un-recognition of the perception system also occurs in the fog situation among the performance degradation problems.

The figures below show that mis/un-recognition occurs using Point Pillar, a reference detector, in the real-driving fog data obtained directly. A misrecognition phenomenon occurs with fog points as a vehicle in various directions. And in the case of Un-recognition, when the vehicle passes next to the fog, the points reflected by the vehicle and the fog points overlap, so the detector fails to detect the vehicle. It can be seen that these phenomena often occur in heavy fog situations. This mis/un-recognition will be described in more detail in Chapter 3. In addition, even in the situation using a fog model rather than a real fog, the same mis/un-recognition situation occurred, and the contents were treated as subsections in Chapter 3, and various situations were described.

Thus, the main objective in this paper is to derive a classified point cloud to eliminate the Mis/unrecognition situation for the perception system is the contention of the de-noising module.

In order to automatically turn on/off the de-noising module, a Weather decision module is developed in front of the de-noising module so that the de-noising module operates only in fog weather. In addition, we developed a Max detectible-range estimation module that operates on the rear end of the de-noising module. The module transmits the detectible range in metric units required by the decision & control part of the autonomous vehicle system in adverse weather conditions. In a heavy fog situation, the density of the fog is quantified, and the distance recognizable by the reference detector is transmitted to the decision & control part.

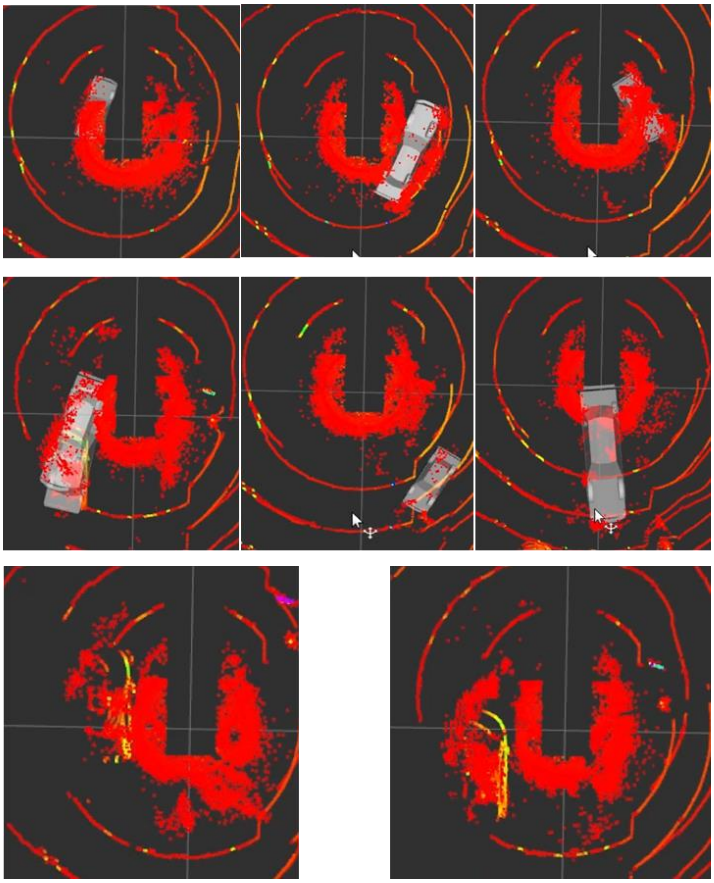


Figure 1.10 Misrecognition results are depicted in upper six figures, Un-recognitions results are last two figures. These results are derived from reference detector PointPillar(detailed detector description is in chapter 3)

In addition, through the acquisition of real driving fog data, the physical characteristics of LiDAR in the heavy fog were also founded. From the obtained real-life fog driving data, the three characteristics of heavy fog are defined as follows:

1. Intensity of Points - the average intensity of total points is lower than that during clear weather

2. LiDAR's laser reaching distance - the maximum reaching distance of total points is reduced.

3. Abnormal Point Cloud - A 'circular point group' is formed around the sensor (fog points).

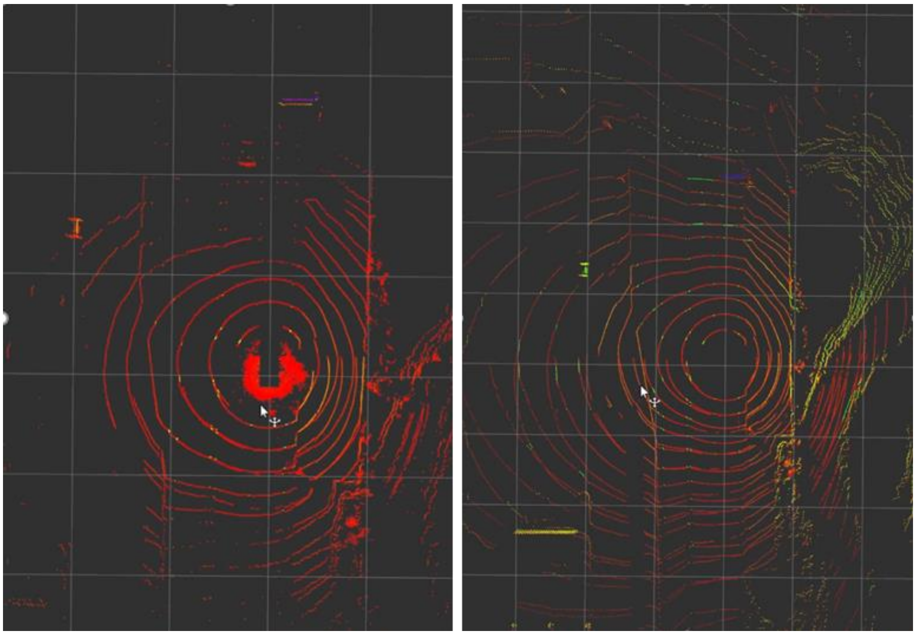


Figure 1.11 In heavy Fog(left) & Clear(right) at the same time zone in the same place(Intensity different – point color indicates intensity of each point, Laser reaching distance different, Fog points detected)

A detailed description is provided in Chapter 2. Considering the physical characteristics mentioned above, a correlation between the density of the fog points generated around the LiDAR, and the maximum detection distance of the LiDAR has been studied. The detailed contents are introduced in the 'max detectible-range estimation module' section.

The contributions could be summarized as follows:

1) the point-wise approach to segmentation of LiDAR point clouds, producing the noise filter with minimal loss of geometrical information. In the validation part, we demonstrate that the proposed method outperforms the conventional one in denoising performance when tested on large datasets obtained in controlled weather environments. Furthermore, qualitative evaluation was carried out on data acquired from driving in actual heavy fog scenarios, which is rare in this research area.

2) After the de-noising module, considering the physical characteristics of LiDAR, a correlation between the density of the fog points generated around the LiDAR and the maximum detectible range of the LiDAR has been studied. This thesis introduces a unique methodology to derive the correlation between the density of the local area's fog points and the maximum detectible range.

3) Before the de-noising module, we also developed a weather decision module based on the point cloud data. The de-noising module operating at all times could be computationally intensive and increase the probability of de-noising meaningful point cloud data. The module obtains point cloud data from several adverse weathers and determines the weather through data clustering

methods. Through weather data observation, the physical characteristics of lidar in adverse weather were found. Fog, snow, and clear weather are classified, and the weather decision result value based on probability is delivered to the denoising module.

1.4 Thesis Outline

The remainder of this dissertation is divided into six sections and organized as follows. The weather decision algorithm with LiDAR point is introduced in Chapter 2. Using the lidar point, the clustering-based method distinguishes snow, fog, and clear weather. Detailed explanations of the proposed De-noising algorithm is in Chapter 3. Evaluation results of the algorithm is described in Chapter 4. Verification data sets are divided into controlled weather chamber data sets and real-driving data sets. Quantitative and qualitative evaluations were also conducted. In Chapter 4, the max detectible-range estimation algorithm is proposed. In chapter 5, the characteristic of lidar sensors in adverse weather conditions is introduced. Finally, the conclusion and future works are provided in Chapter 6.

Chapter 2 Lidar-based Weather Decision Module

2.1 Overview of De-noising modules

The modules proposed in this thesis can be largely divided into three categories. Furthermore, the detailed figure is described in Figure 2.1. The first module is the Lidar point-based weather decision module. The module receives the original lidar raw point cloud as input. It distinguishes three types of weather: fog, snow, and clear. The module's output is whether the weather probability and the de-noising module are operated. As a result of analyzing the directly acquired Real driving data, an index was set for classifying the weather. There are a total of three indexes. The first is the Total Lidar point cloud number, the second is the point cloud's intensity average, and finally, the local area (3.5m) points number.

The second module is the Point-wise de-noising module. This module performs point-wise segmentation to distinguish between fog and non-Fog points. Many works [Graham'18, Chen'19] achieve great semantic segmentation and object detection results by projecting the 3D PCD onto 2D pixels or voxelized to 3D voxels. However, geometric details may be lost during the processing. For example, the points in the same voxel or grid are classified as the same class in the semantic segmentation process. These misclassified points are removed during the de-noising step, resulting in degradation of overall object detection performance. This de-noising module aims to design a weather noise filtering semantic segmentation network that can directly process the raw 3D PCD in one process without pre/post-processing stages. We randomly sample the key points at each layer to efficiently process large-scale point clouds. According to [Hu'20], It is essential to preserve detailed local information. Essential points may be lost in the decoding process through random sampling. To handle this problem, we propose an attention feature extraction to learn local relational feature characteristics based on self-attention. Through encoding layers, we extract the local feature vectors of key points via KNN algorithm. Then this operation continues at each layer to enlarge the receptive field. In decoding layers, the up-sampled feature maps are concatenated with the intermediate feature maps produced by encoding layers by skip connections. The network adopts the commonly used encoder-decoder structure. The network contains four Encode and Decode layers, which are used to train features of points. The later part of the network includes fully-connected layers and a dropout layer to predict the label per point. More details are covered in Chapter 3. The input of this module is the output of the weather decision module, and the output of this module is the point cloud classified with filtered non-fog.

Finally, the third module is named as perception algorithm max detectible-range estimation module. The module quantifies the density of the fog point cloud. The quantification method makes the 3d point cloud a 2d bird eye view and then uses the 2d occupancy grid map(OGM) method. The OGM method was used to quantify the probability of existence for each grid in the previous step and the current step. Since the OGM method is a well-known method, detailed methodologies are omitted. When 3D OGM is used, the real-time performance of the algorithm deteriorates so in this section we used 2d OGM method.

We try to derive the perception range according to the current density of fog in the heavy fog situation. A perception algorithm that becomes a reference is needed to derive a perception range in the process. Thus, this thesis uses a LiDAR point cloud Network-based perception algorithm called Point Pillars[Lang'19] as a reference perception algorithm. This algorithm ranked high detection performance (KITTI object detection task 5th). It is an open-source code and has real-time performance. Fog density quantification simultaneously stores the results of the detectible range of nearby vehicles in that situation. The detailed process and result are described in Chapter 5.

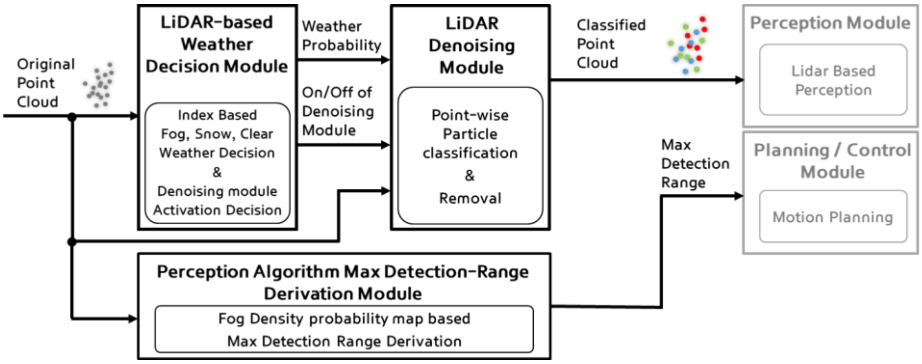

Original Point Cloud
LiDAR-based Weather Decision Module
Index Based Fog, Snow, Clear Weather Decision & Denoising module Activation Decision
Weather Probability
On/Off of Denoising Module
LiDAR Denoising Module
Point-wise Particle classification & Removal
Classified Point Cloud
Perception Module
Lidar Based Perception
Max Detection Range
Planning / Control Module
Motion Planning
Perception Algorithm Max Detection-Range Derivation Module
Fog Density probability map based Max Detection Range Derivation

Figure 2.1. Overall structure of Weather decision / De-noising / Max detectible range estimation Modules.

2.2 Lidar-based Weather Decision Module

This Weather Decision Module is a pre-stage of the Denoising algorithm, as shown in Figure 2.1. The LiDAR PCD is received to determine the current weather situation probabilistically and decide whether to execute the Denoising Module.

The most important part of the module's development was observing the point cloud's driving data in heavy fog and heavy snow situations. The description of the module concept is divided into three parts as follows and depicted in Figure 2.2:

STEP 1: Weather Data acquiring & Index map Derivation

STEP 2: Cluster-based Weather's Centroid point Derivation

STEP 3: Centroid points-based Weather's probability Derivation]

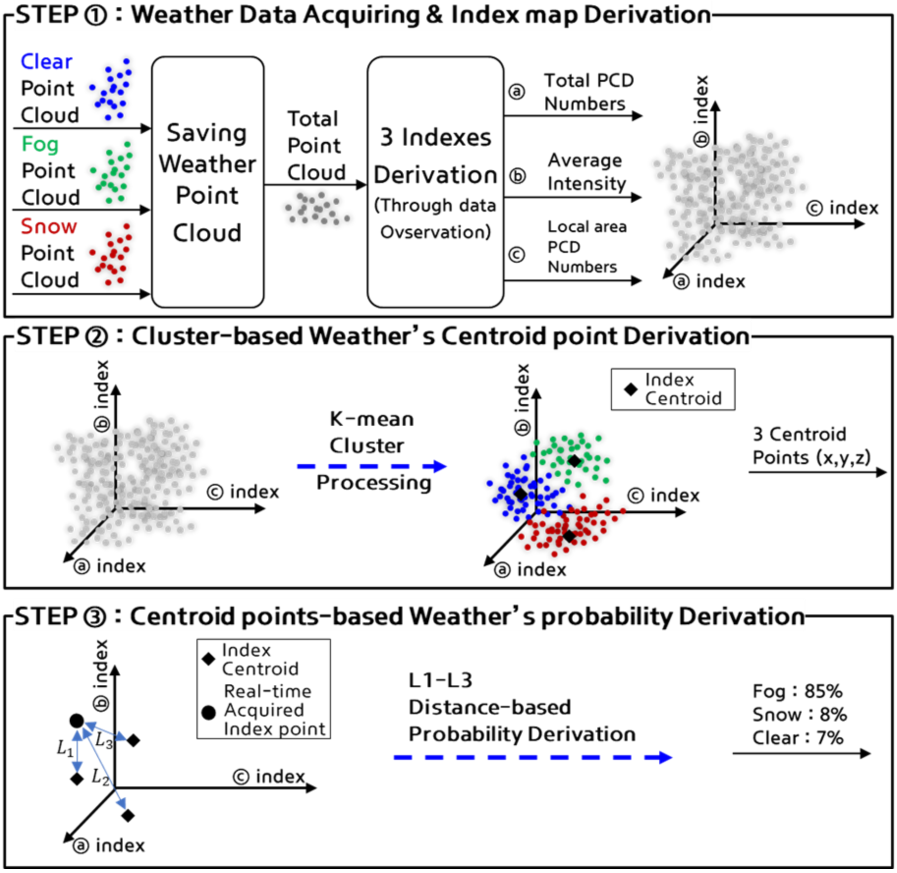

STEP ① : Weather Data Acquiring & Index map Derivation
Clear Point Cloud
Fog Point Cloud
Snow Point Cloud
Saving Weather Point Cloud
Total Point Cloud
3 Indexes Derivation (Through data Ovservation)
ⓐ Total PCD Numbers
ⓑ Average Intensity
ⓒ Local area PCD Numbers
ⓑ index
ⓒ index
ⓐ index
STEP ② : Cluster-based Weather's Centroid point Derivation
ⓑ index
ⓒ index
ⓐ index
K-mean Cluster Processing
◆ Index Centroid
ⓑ index
ⓒ index
ⓐ index
3 Centroid Points (x,y,z)
STEP ③ : Centroid points-based Weather's probability Derivation
ⓑ index
L_1
L_2
L_3
ⓒ index
ⓐ index
◆ Index Centroid
● Real-time Acquired Index point
L1-L3 Distance-based Probability Derivation
Fog : 85%
Snow : 8%
Clear : 7%

Figure 2.2. Three steps for Weather decision module process

Regarding STEP 1, data of various weather data are collected first, and three indexes are derived through data observation. Create a 3d-index map of the index and scatter the entire point over the 3d-index map. The explanation of the essential points when creating an index map is described as follows:

1) Index variables that are not related to the surrounding environment: For example, Lidar's point interval distance may be high in fog situations when there are no vehicles around, but it may be low when there are many vehicles/or

when driving on narrow and high roads. Thus, the impact of this surrounding environment should be excluded.

In Figure 2.3, the left side collects the point cloud passing through a single lane within the test track. On the right is the situation when driving with surrounding cars on an urban road. What can be seen from the shape of the point is that variables unrelated to the surrounding environment should be selected as an index. Thus, the entire point cloud's arrival distance should not be selected as an index.

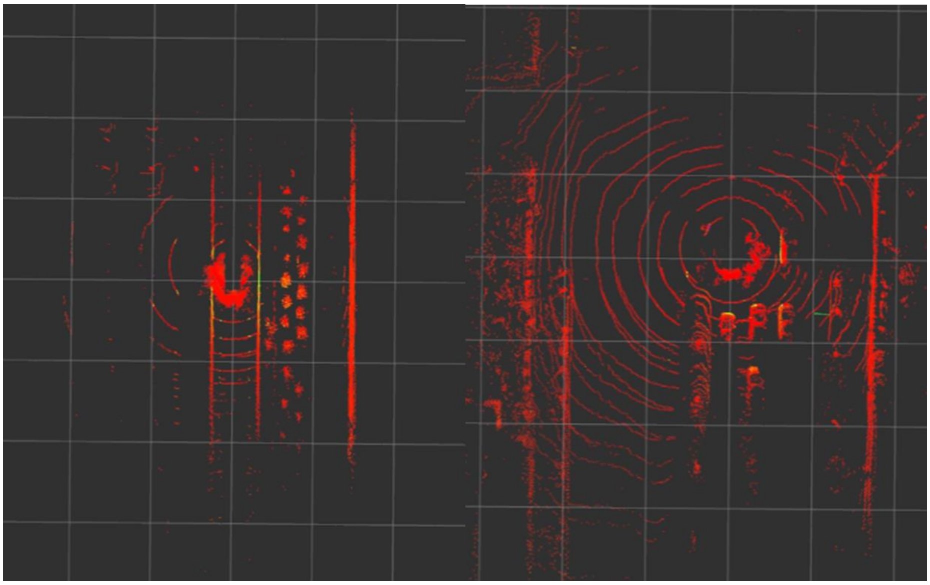


Figure 2.3. Point cloud shape in Heavy fog (Left: driving in test track,Right: driving in urban road)

2) Selection of variables considering the physical properties of LiDAR: heavy fog or heavy snow days, there are situations in which the shot light disappears and cannot return. Thus, the intensity value of the entire point, which is an index that indicates the situation well, is used.

Based on many previous studies and acquired data, the intensity of the point cloud in the [Carballo'20, Heinzler'20, Bijelic'20, Lee'22(d)] heavy fog situation is very low. The directly acquired data could be more clearly identified by directly analyzing the intensity. Details of this are described in the section on the singularity of the physical properties of Chapter 5 and LiDAR. Therefore, it was used to develop an index by taking advantage of the low intensity.

3) Total point number changed: The number of points received may be different for each step in which LiDAR is operated. However, in adverse weather situations, the differential value of lidar total points is very big. The considerable differential value means that the emitted light did not return due to the water droplets generated by the weather. Thus, the total point value is derived as one index.

4) Local area's point number: In the Fog/Snow situation, the number of points in the local area (within 3.5m) and shape singularity are selected as index. In the Snow and Fog situations, it was found that the movement was different from the shape of the point occurring in the surrounding area. Furthermore, it can be seen that the number of points is very different from the clear situation. Thus, we made such a singularity index.

In conclusion, there were a total of three indexes. The first is the 1. LiDAR PCD Points Number, the second is the 2. Intensity average, and finally, the 3. Local area (3.5m) Points number. Three indexes have been determined as above, and the determined index is delivered for use in STEP 2.

The following process was scattering on a 3d index map based on three indexes of points for three kinds of weather on the index map.

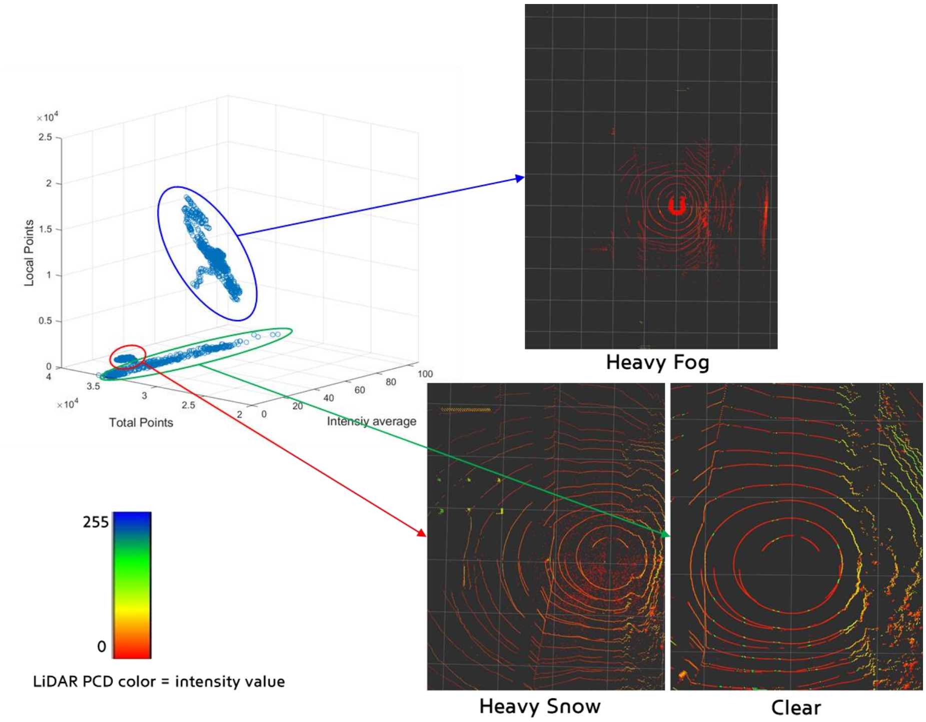


Figure 2.4. weather point scattered results in 3d index map

One point in the 3d index map is generated for each sensor operating step. Thus, weather-specific scattering is performed on the 3d index map. The dataset used to generate an index map used data in situations where only weather variables were changed in the same place.

Figure 2.5 shows the result of scattering on the 3d map, and Figure 2.6 shows the result of viewing 3d in three view directions.

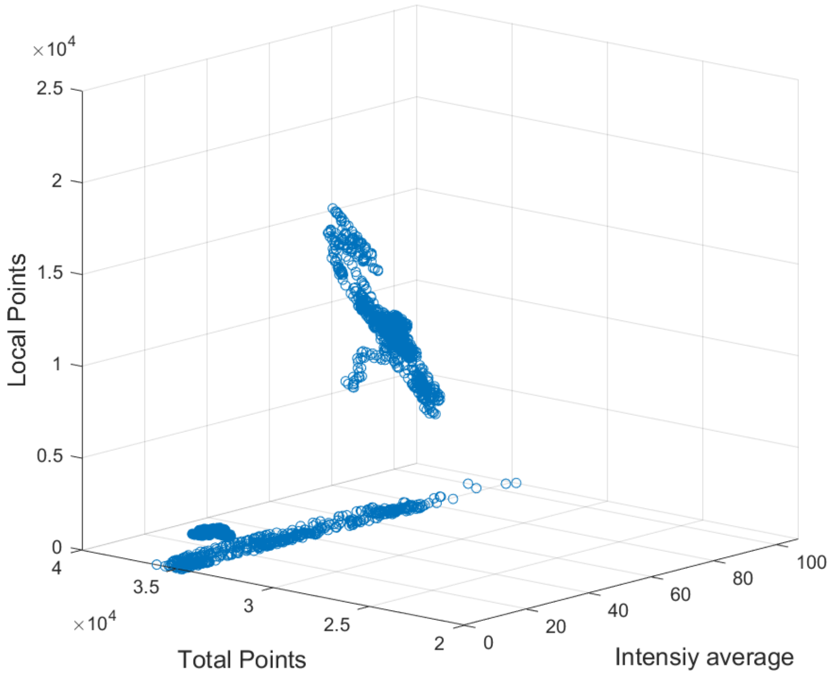


Figure 2.5. Fog, Snow, Clear weather points in 3d index map

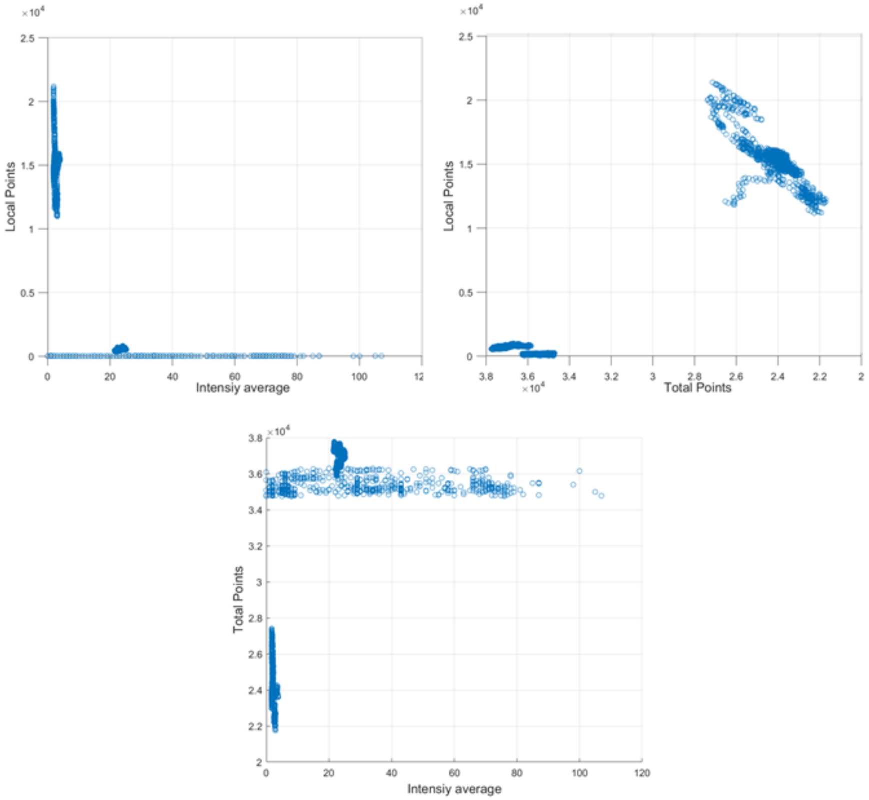


Figure 2.6 A plan view looking at the 3d index map from 3 directions.

In STEP 2 and 3, three types of weather in the index map are first classified through the K-mean cluster. Then, the centroid of each weather group is derived, and the 3d coordinates(x,y,z) of the centroid point are derived. The value of the coordinate point is used when the real-time operation is performed. In real-time, it is possible to probabilistically display the current weather by scattering one point, which means the current weather in the 3d index. Through the 3d index map acquired offline, the probability of the current weather is displayed through the Centroids point by weather and the 3d index distance at the current time. As shown in the figure below, the result value of K-mean clustering shows that all points are labeled with excellent performance without error.

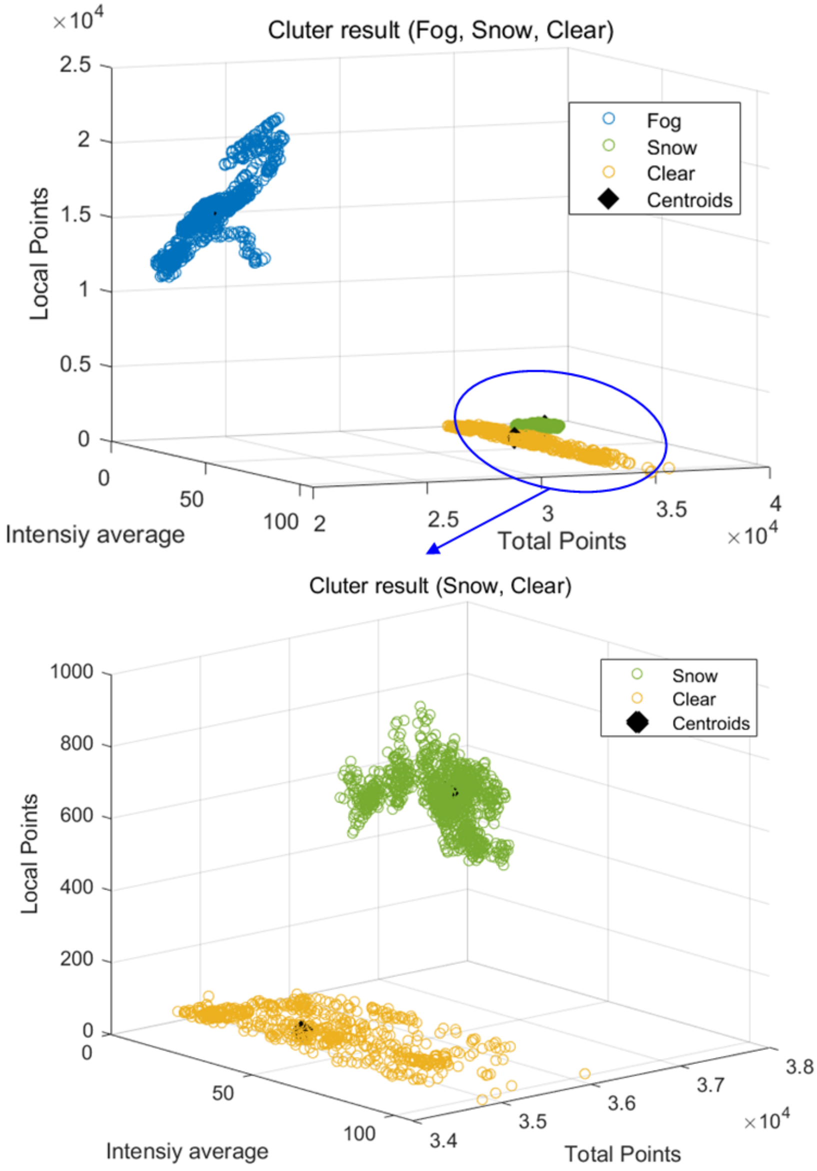


Figure 2.7. K-mean cluster result

Table 1. 3d index map based weather percentage result in various scenes

Various Scenes			Fog Percent Ave (%)	Fog percent (%) Min/Max/STD	Snow Percent Ave (%)	Snow Percent (%) Min/Max/STD	Clear Percent Ave (%)	Clear Percent (%) Min/Max/STD
Fog	In test track	Scene1(heavy fog)	**89.54**	57.24 / 99.32 / 5.06	5.14	0.07 / 20.94 / 5.06	5.32	0.08 / 21.82 / 5.25
		scene2(light fog)	**87.15**	51.24 / 98.82 / 5.24	6.22	0.12 / 22.09 /5.31	6.63	0.13 / 21.07 / 5.07
	Urban road Driving (low velocity)	Scene1	**83.08**	59.24 / 97.91 / 5.02	8.26	0.04 / 19.84 / 5.20	8.66	0.10 / 20.96 / 5.11
		Scene2	**84.29**	43.71 / 99.03 / 4.52	7.51	0.15 / 19.10 / 2.04	8.20	0.28 / 14.58 / 4.49
	Urban road Driving (high velocity)	Scene1	**82.34**	58.89 / 99.27 / 5.17	8.73	0.23 / 21.10 / 3.27	8.93	0.09 / 20.91 / 3.50
		Scene2	**88.04**	55.13 / 98.08 / 5.11	6.51	0.09 / 18.73 / 5.47	5.45	0.10 / 19.77 / 5.11
Snow	In test track	Scene1(heavy snow)	0.56	0.26 / 1.37 / 0.30	**86.72**	70.03 / 99.41 / 7.50	12.72	0.56 / 28.90 / 7.21
		Scene2(light snow)	0.37	0.13 / 0.81 / 0.27	**81.07**	77.28 / 97.71 / 8.28	18.56	0.75 / 30.44 / 8.35
clear	In test track	Scene1	0.11	0.32 / 1.01 / 0.37	9.19	0.73 / 26.45 / 6.48	**90.70**	72.54 / 99.24 / 6.73
	Urban road	Scene2	1.01	0.71 / 1.57 / 0.42	10.37	1.07 / 30.54 / 7.13	**88.62**	68.49 / 97.60 / 7.23

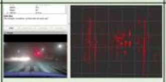
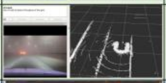
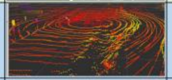
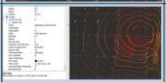
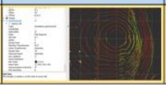
In Table 1, point scattering is performed based on the 3d map index. Furthermore, this is the result of classifying based on the k mean cluster and classifying the weather based on the distance of the cluster's centroid.

For Fog weather, On the test track, the Urban road driving with low speed and Urban road driving with high speed has an average accuracy of 88.35, 83.69, and 85.20%, respectively. In Snow weather, it has an average accuracy of 83.90% through the index. In clear weather, it has an average accuracy of 89.70%.

Point cloud data acquisition location and details

As shown in the figures below, the total amount of data is about 300 minutes and about 280GB of data. Data were acquired in several scenarios, mainly from inside the test track and Urban Road. The Figure 2.8 shows the test track details as follows. Moreover, Figure 2.9 is an overview of the road from which the data was acquired. It was mainly acquired from roads in Siheung-si and bridges to Daebudo Island. The in-vehicle test result video capture is depicted in the Conclusion part with de-noising module results.

Table 2. Data acquiring details

Scenarios		Representative Figures	Data Amout(Gb)	Data Time (Minutes)	Reference Cluster Data length (figure2.5)
Fog	Heavy Fog (Track)		51.8	61.9	15.7Gb
	Light Fog (Track)		18.6	21.5	X
	Light Fog (Low Vel, Urban)		94.2	109.2	X
	Heavy Fog (High Vel, Urban)		56.5	65.5	X
Snow	Heavy Snow (Track)		12.1	14.0	11.7Gb
	Light Snow (Track)		16.5	19.1	X
Clear	Clear		28.3	32.8	20.6Gb

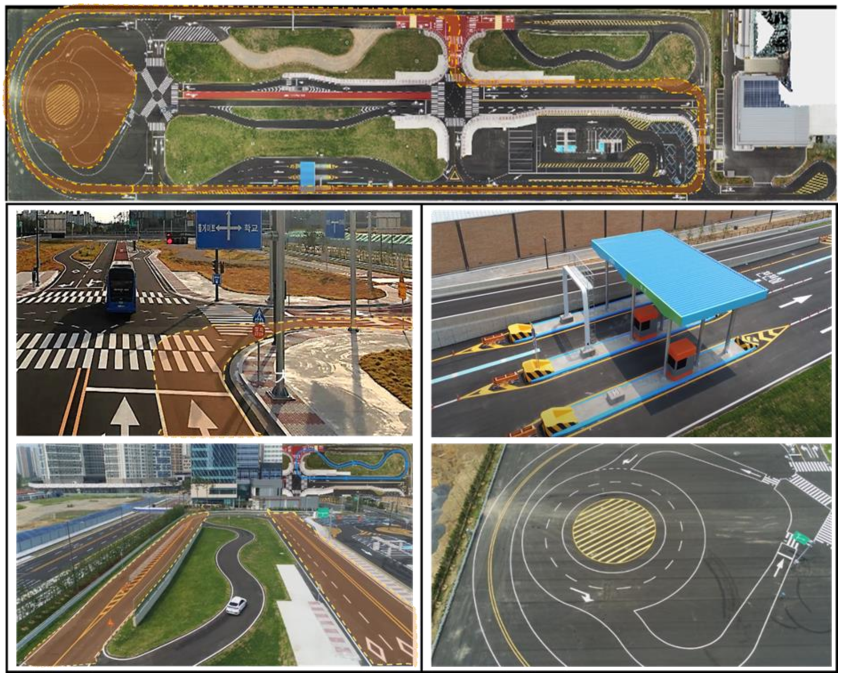


Figure 2.8 Test track detail figures

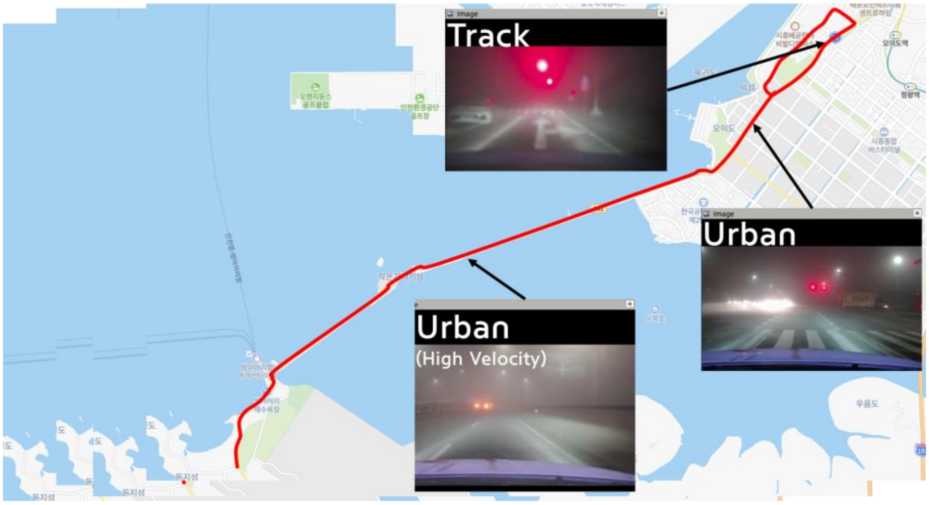


Figure 2.9. Data acquiring spots

Clustering details

In this part, we will explain in technical detail how Point cloud data were clustered, the evaluation environment, and how the results were derived.

As for the data, a large amount of data was used, as shown in Table 2. The k value in the K-mean cluster is 3. In other words, the three weather categories are classified. Explaining the creation of a reference cluster for fog, the data is extracted in the densest situation of fog. In the same way, three weather reference weather clusters are created. The result is drawn on the index map as shown in Figure 2.5 above. Reference cluster data is not used for validation. After that, when validation was performed as table 1, validation was performed using the data from the scene corresponding to several scenarios, and quantitative cluster results were derived.

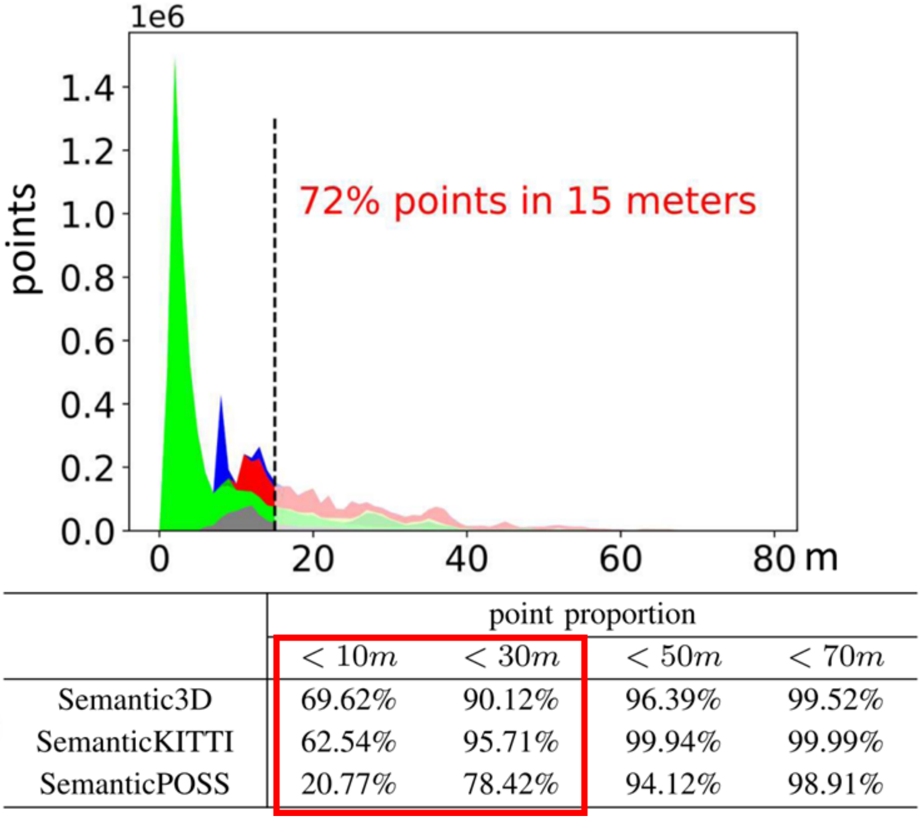
Chapter 3 De-noising Module

This chapter will be developed in the following order. The introduction part deals with concerns in the process of developing deep learning architecture and dealing with fog points. Explain key points in network design and explain how to respond. In the related work part, previous deep learning architectures are introduced about point cloud segmentation. The methods part describes how to solve the various challenges mentioned above. Network architecture describes the final network configuration. Fog model analysis explains the necessity of a fog model. The most commonly used fog models are described. Last part of this chapter, we introduce a phenomenon that perception module detection error occurs because of fog points.

3.1 Network Design Considerations

Typical convolutional architectures use regular input data format to optimize better using voxel or image pixel. However, the point cloud is not a regular format. Different from 2d regular grids, 3d point cloud data is irregularly sampled. Therefore, it is impossible to apply standard CNN directly to unstructured data. In addition, point cloud data is also non-uniform structured. The point cloud density is usually dependent on the LiDAR sensor’s distance. Furthermore, the raw point cloud data with various LiDAR channels could acquire large-scale and immense data in one sensor frame rate. Moreover, for this reason, we focus on directly processing 3d point cloud data without converting, discretization, and projection to other representations. This architecture directly takes point clouds as input and outputs class labels for the input data.

Directly process 3D-Point Cloud Data(PCD). Referring to the [Gao'21] paper, 72 percent of the points have their own characteristics within 15 meters. Furthermore, voxelized those points, the point/voxel production value remains very constant, as shown in the Figure 3.1 and 3.2, as the distance from the sensor increases. Since PCD has these characteristics, if the original raw PCD is changed by voxelization, many points will be lost. In particular, regardless of the sensor, most of the fog points appear within 15m of the sensor. Figure 3.3 shows that the shape of fog appeared within 15m with various LiDAR sensors. Thus, it is a big reason to use PCD directly to maximize point characteristics.



	point proportion			
	$< 10m$	$< 30m$	$< 50m$	$< 70m$
Semantic3D	69.62%	90.12%	96.39%	99.52%
SemanticKITTI	62.54%	95.71%	99.94%	99.99%
SemanticPOSS	20.77%	78.42%	94.12%	98.91%

Figure 3.1. Points number change with distance [Gao'21]

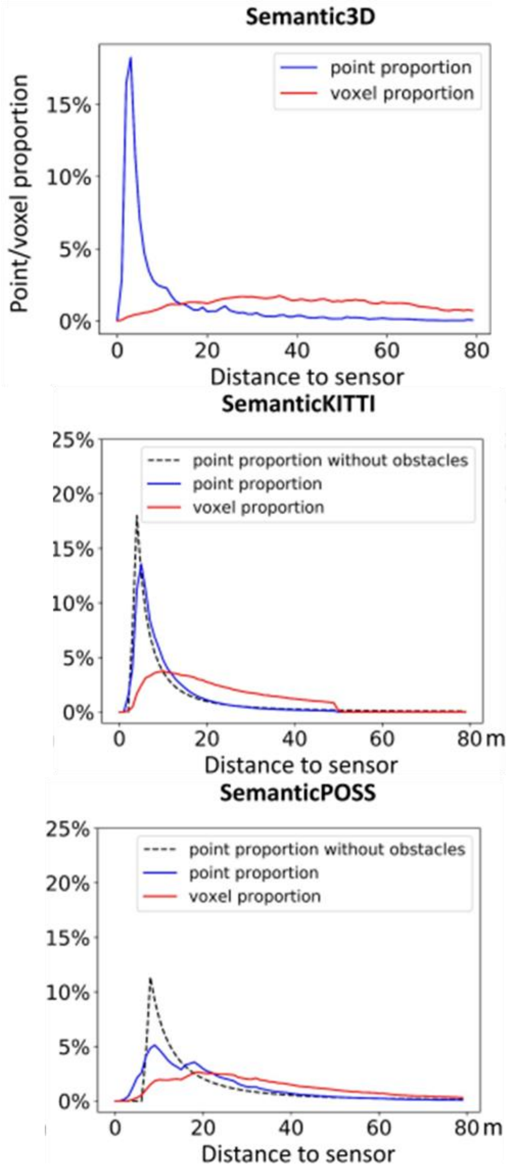


Figure 3.2. Raw 3d points & voxelized points proportion changes with distance[Gao'21]

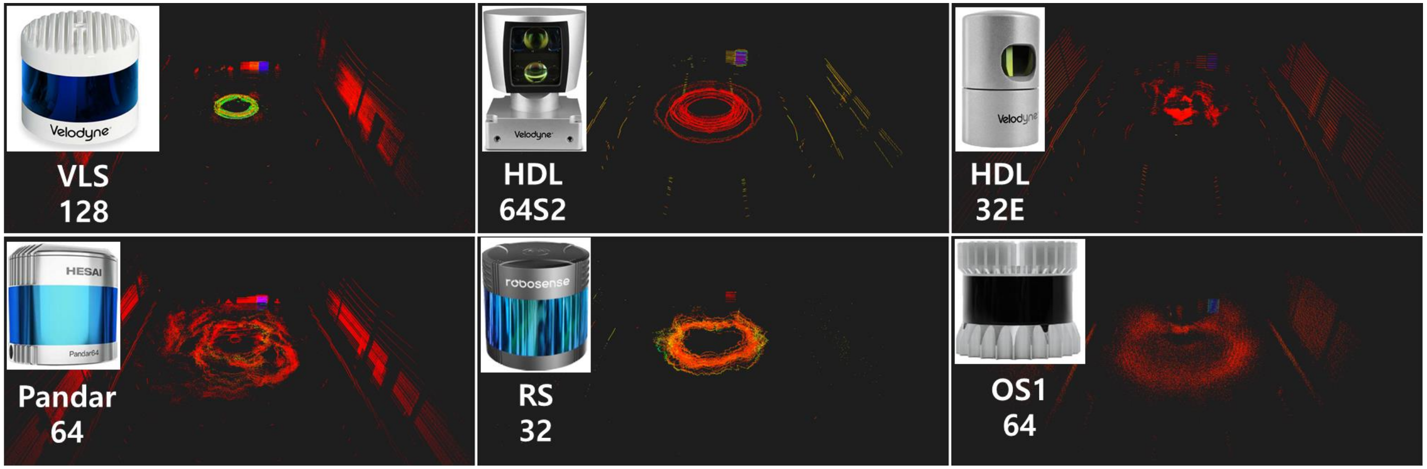


Figure 3.3. Results of different types of lidar sensors PCD scrattering [Carballo'20]

Referring to the paper of [Carballo'20], the appearance of fog is scattered according to the type of sensors, as shown in Figure 3.3. When looking at the scattered results of the point cloud, it can be seen that most points are distributed within 15m. It is challenging to know the exact data, but it can be inferred through the size of the experimental environment through figures. In other words, it can be a reasonable basis for the reason for using raw 3d points directly, rather than proceeding with discretization, voxelization, and projection. Since the feature of fog points is used as much as possible and has intensity characteristics, it is considered that the direct processing of 3D points is the best solution to extract the feature of fog point cloud.

It can be verified through the experimental data that the real driving fog data obtained directly is also distributed within 10-15m, as shown in Figure 3.4.

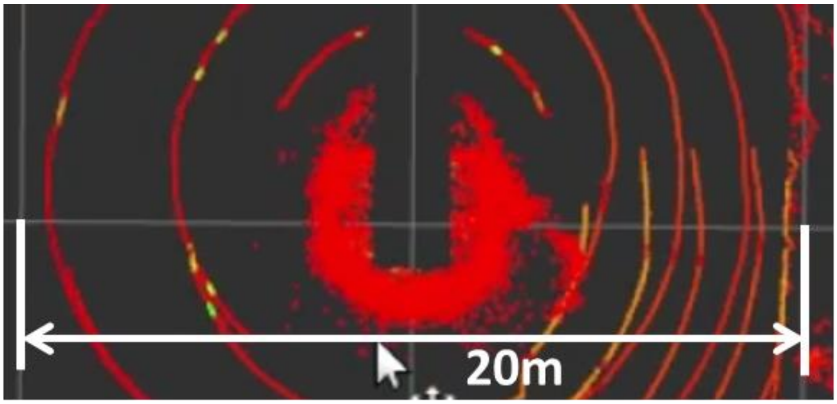


Figure 3.4 Real driving fog data shape[J.Lee(a)]

We proceeded with the network configuration by adhering to the following main points when we designed the network.

The key points of this chapter are as follows:

- Network inference speed should be proper to use for an autonomous vehicle in real-time
- Local Geometrical information preserves motivated Network
- Without Pre/Post-processing, directly process 3D Point cloud data without projection and discretization
- In the validation part, a performance comparison was conducted with previous research in controlled weather environment data. Furthermore, fog model-based de-noising performance was also evaluated.
- Real driving data in the fog was acquired for this study and utilized in Learning and validation.

Our network frameworks are directly used unordered point clouds as inputs. A point cloud vector of (x,y,z, i) coordinates. i coordinate in point cloud vector means intensity channels. The intensity of the point cloud, which appears as a fog droplet, has a very low intensity that is very close to zero.

LiDAR point clouds sparsity

There is a significant difference between the data format of the LiDAR point cloud and the image data format. RGB data is 2d data in pixels. Moreover, each pixel is filled with a meaningful value. Networks based on 2d and pixel-wise

RGB data have been developed for a long time. Thus, many studies have considered how to learn by changing point cloud like an image. The voxelization method is the most frequently used method of changing the PCD to image. The most vulnerable point of the Vexelization method is the physical characteristics of sparse PCD data. Voxelize PCD causes information loss. Furthermore, through voxelized, there are many voxels with empty values. The reason is due to the characteristics of PCD in which 72% of PCD described above is distributed within 15m.

3.2 Methods

As described in the previous chapter, the shape of lidar point cloud data in heavy fog situations has the following characteristics.

1. As there are more lidar channels and lidar with more points, a circular droplet cloud is created around the sensor.

2. The intensity of the points is close to 0

3. Point cloud continuously moves

Many of the studies mentioned in previous sub-chapter have been studied for the segmentation of point clouds. The problem to be solved in this dissertation is to “de-noise” the fog point cloud around the sensor in a heavy fog situation. Many previous studies were referenced to find a suitable method for this problem.

The purpose of this problem can be divided into two parts, as follows:

1. Development of a de-noising algorithm could perform within the sensor frame rate (10~20Hz).

2. The performance of semantic segmentation is that there is no misrecognition phenomenon in heavy fog scenes.

The following 3 contents are the characteristics of the fog point cloud that should be considered when researching a method.

a) Statistically, about 72% of the point cloud is distributed within 15m [Gao'21]

b) In real-life driving fog data, most of the fog points are clustered within 7m to constitute the fog shape.

c) The intensity is very low and suitable for use as a point feature in the Feature Aggregator.

Therefore, methods to achieve the above objectives are listed as follows:

1) Algorithm operates within the frame rate (10~20Hz) of the LiDAR sensor

- Large-scale point cloud must be processed in a single pass without pre/post processing
- 4 pairs of Encoding, Decoding Layers
- Sampling method (Random sampling method based down-sampling)
- KNN based Key point selection
- Computationally efficient MLPs (In up-sampling decoding part, feature learner MLPs are computationally cheap but have a disadvantage in capturing broader geometric information for each point.)

2) Minimize the loss of geometric information

- In the down-sampling process in encoding layers, the feature aggregator diverse intensity point feature numbers to enlarge receptive field size. Even if meaningful points are dropped during down-sampling, geometric details of fog clouds could be preserved.

The detailed methods are explained in follow

Differences between Sampling Method and De-noising. The sampling method is the most necessary method for point-wise segmentation. It is essential to process 40,000 to 100,000 point data within the sensor frame. However, real-time processing is difficult if many PCDs are simultaneously fed into the network at once. Thus, the work of compressing and leaving a meaningful PCD through the encoding layer is carried out through sampling. In this paper, sampling means this. Thus, the sampling method is explained a lot in chapter 1.2.1.

Sampling methods. Unlike previous works using a pseudo-image, which is easy to apply convolutional architecture, such as voxelization, it is important how to extract the features from a local area and gradually dilate the receptive field to get more global feature. Previous works used sampling approaches [Yang'19, Qi'17, Li'18, Groh'18, Dovrat'19, Abid'19, Wu'18] is divided into conventional and learning-based approaches.

The most commonly used conventional sampling is Farthest Point Sampling(FPS) and Random Sampling(RS). FPS is famous because [Qi'17, Li'18] is used for sampling 3d point cloud data before being fed into the network.

The best pros of this method are that it has significant convergence. However, computational complexity is O(N^2). Therefore, it could adopt in large-scale point cloud data. RS selects determined parameter points K from original raw points. It has O(1) computational efficiency; it is proper to process large-scale data. From[Hu’20], it could process 106points in a 0.004s. Commonly used learning-based sampling is Generator-based Sampling (GS) and Continuous Relaxation based Sampling (CRS). These methods are too computationally expensive to use in our research. Random Sampling(RS) could efficiently decrease total input points and be memory-efficient for computation. Therefore, we adopt RS to our point processing for large-scale point data. RS also has the disadvantage that it could easily drop useful point cloud; however, in the following contents, we introduce the feature aggregator to enlarge the receptive field to preserve the point intensity feature.

Encoder and Decoder. The encoder consists of four layers, and they used KNN to reduce the number of PCD, simultaneously increasing the feature dimension. Encoding layer has a KNN-based RS(random sampling) and Feature Aggregators. Passing through each layer, only 25% points are remained, i.e., N → N/4 → N/16 → N/64 → N/256. Meanwhile, the intensity point feature dimension is gradually increased with each layer , i.e., 4 → 16 → 64 → 128 → 256. The decoding layer is configured by being connected after the encoding layer. For each decoding layer, we defined an important, meaningful query point through KNN. The feature set of the point is upsampled through KNN interpolation. Through skip connection, the feature map generated in the encoding layer and the upsampled feature results are integrated together. The sMLPs is applied to the concatenated fature maps. These skip connection methods are utilized from autoencoders for reconstruction images due to the information loss. Residual networks comprising skip connections are a known solution to this problem. Skip connections can be added from the encoder to the decoder to improve the autoencoder performance. These additional connections can directly send the feature maps from the earlier layer of the encoder to a later layer of the decoder. It could help the decoder structure describe decompressions of the input image more clearly[Olaf'15, Gao'17]. Overall encoder and decoder structure is depicted in network structure at Figure 3.5.

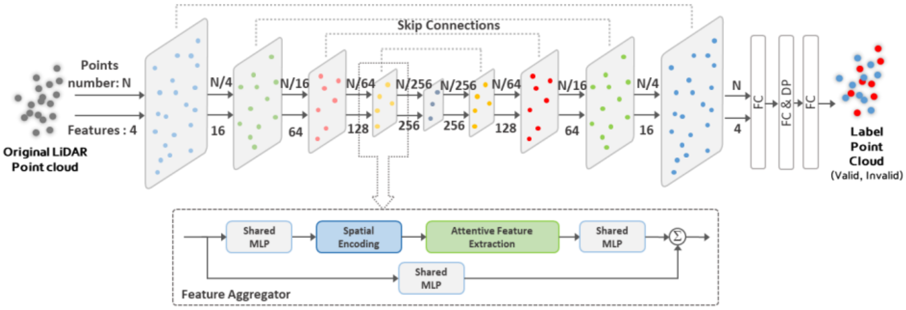


Figure 3.5. Network architecture details. Input number : N, Features : 4, FC : Fully Connected, DP : Dropout

Feature Aggregator After the feature aggregator module receives a large-scale point cloud as an input, the encoder performs down-sampling. The input has a matrix with x-y-z coordinate information and a 1byN intensity feature matrix with point-wise intensity. We used these two coordinate and feature vectors to down-sample the points and thought of a way to accommodate the information about intensity features better. To do this, we first use an algorithm called K-Nearest Neighboring (KNN). This method has already been used in many papers and is considered a meaningful and fast algorithm. The KNN algorithm is based on the point-wise Euclidean distance, and the distance information is also used in the process. Calculate the relative positions of neighboring points. To a points P_i , K nearest points are defined as $\{P_i^1, P_i^2, \cdots, P_i^k\}$, and these points are used to encode relative Euclidian distance to get R_i^k. Many studies used KNN to perform neighbor point gathering [Fan'21, Qi'17, Hu'20, Englemann'20, Zhao'19]. Encoding for position allows the neural unit to become more aware of the local structure of fog point cloud data. In the KNN algorithm, the K value is the threshold value of the number of capturing neighboring points. In this thesis, a value 16 is selected. The KNN algorithm obtains relative spatial position information and intensity feature matrix. A net set of neighboring K features through the concatenate of the two matrixes.

The encode method of relative position is defined as follows:

$$R_i^k = MLP((P_i \oplus P_i^k) \oplus (P_i - P_i^k) \oplus \left\| P_i \oplus P_i^k \right\|) \tag{3.1}$$

After encoding, the feature matrix (intensity matrix) and the feature matrix's corresponding relative position matrix are concatenated, and the dimension is doubled.

Most intensity features have values close to zero in foggy situations. [Liu'19] proposed RS-CNN using a new point relationship. It has more advantages than KNN in terms of accuracy. The overall design of feature aggregator is depicted in Figure 3.6. and Figure. 3.7.

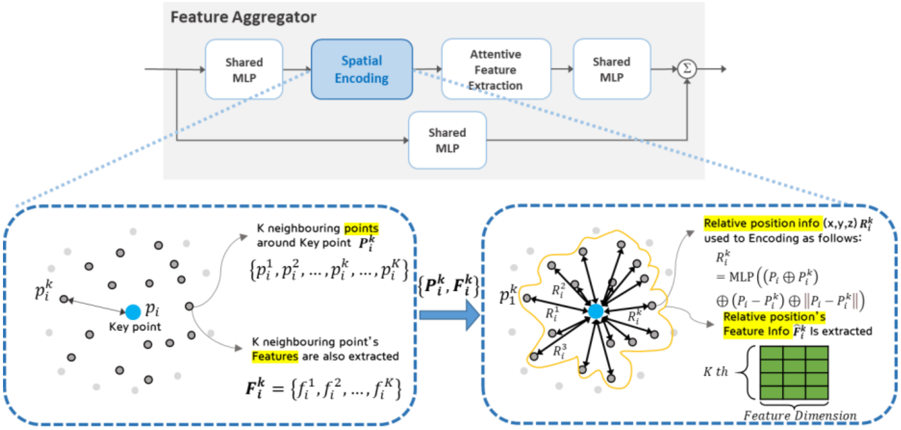

Feature Aggregator
Shared MLP
Spatial Encoding
Attentive Feature Extraction
Shared MLP
Shared MLP
Σ
p_i^k
p_i
Key point
K neighbouring points around Key point P_i^k
$\{p_i^1, p_i^2, \ldots, p_i^k, \ldots, p_i^K\}$
K neighbouring point's Features are also extracted
$F_i^k = \{f_i^1, f_i^2, \ldots, f_i^K\}$
$\{P_i^k, F_i^k\}$
p_1^k
R_i^1
R_i^2
R_i^3
R_i^k
Relative position info (x,y,z) R_i^k used to Encoding as follows:
$R_i^k = \mathrm{MLP}\big((P_i \oplus P_i^k) \oplus (P_i - P_i^k) \oplus \|P_i - P_i^k\|\big)$
Relative position's Feature Info $\hat{F}_i^k$ Is extracted
K th
Feature Dimension

Figure 3.6. KNN and Position encoding in spatial encoding

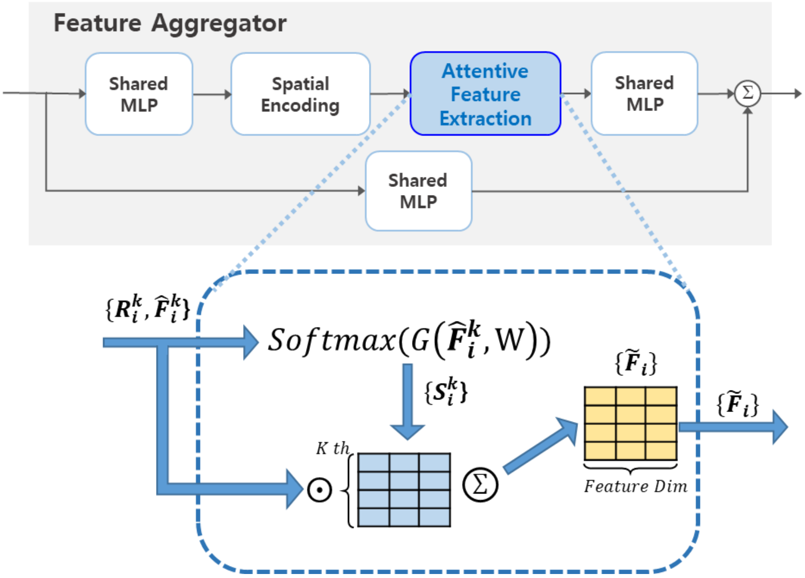


Figure 3.7. AFE in Feature Aggregator

In attention feature extraction (AFE) modules, a form of self-attention is adopted to extract feature similarities of the $\mathrm{i^{th}}$ points, and aggregate their features to represent the structural information regarding the $\mathrm{i^{th}}$ point. AFE uses only the neighboring points around each point and the encoding of positional information. The AFE used is based on vector self-attention and uses subtraction for relation.

A Typical Attentive mechanism is used in the AFE module. $\{\boldsymbol{R_i^k}, \boldsymbol{\hat{F}_i^k}\}$ feature set product with After softmax activated function matrix. After that, sum up the weighted Vectors. Computing attention scores; Given a local feature set $\boldsymbol{\hat{F}_i^k}$, we design a shared function G() to learn a unique attentive score for each feature. The function G() consists of sMLPs and softmax, where W is the learnable weight of sMLP. In weighted sum par, the attentively learned scores are considered a soft mask that automatically selects essential features. It means that essential feature data with high weight is calculated through $\boldsymbol{\hat{F}_i^k}$ and calculated $\boldsymbol{S_i^k}$ through the weighted sum; After that, the mapped point data can be calculated.

3.3 Fog model Analysis

In this chapter, we explain the necessity of the fog model, the previous research related to the fog model, and check what it looks like by running the fog model. It cannot be said that many studies on point cloud de-noising have been done yet. However, some studies conducted research on point-wise fog denoising, and research on the fog model was conducted in the process of denoising.

The reason the fog model is needed is that it is difficult to obtain data with fog labeling. Thus, stationary data acquisition is carried out in a weather controlled chamber for point-wise labeling of fog. Acquire ground truth point data for an object through lidar stopped in a chamber with clear weather conditions. And if the lidar is not moving and the chamber weather is changed to fog or rain, the rest of the point data is droplet data except for the ground truth point. Therefore, many studies have obtained adverse weather point data through this method

However, all point data acquired by the above method are data in a stopped state. Therefore, adverse weather point data is a very important part for autonomous driving in real driving data situations. Thus, several studies have created virtual fog weather point data using mathematical fog models on clear weather data[Li'20, Heinzler'20]. Fog models have been used for many purposes in this field. It is okay to consider it as a method of data augmentation. Recently, many point cloud datasets have been released and available. However, most of the data are data obtained under favorable weather conditions. Furthermore, even if adverse weather data is acquired, the data is precious and is not open for free. In order to utilize the data acquired in such favorable weather conditions, the fog model is used for semantic weather segmentation.

In this study, we use the model that is used the most and can quantitatively evaluate the de-noising algorithm developed in this dissertation.

Sensing range and Intensity decrease considered model firstly proposed in [Heinzler'20] and, it is the most widely used fog model. Through this model, we could compare the de-noising qualitative result with other studies. This model considered the max sensing range, atmospheric extinction coefficient(fog density level parameter), and Lidar sensor parameter. The maximum sensor range is the half of the maximum viewing distance and results in :

$$d_{\max} = \frac{-\ln(\frac{n}{L_{fog} + g})}{2 \cdot \beta} \tag{3.2}$$

The sensor threshold is function of laser intensity L_{fog}, laser gain g and the detectable noise floor n. For n and g, the Velodyne 32 puck-c model's parameter were used in this dissertation. The atmospheric extinction coefficient β is an adjustable fog density. d_{max} is a parameter that affects the density of fog by reducing the maximum distance. In this study, a β value of 0.01 ~ 0.07 was used.

Max sensing range d_{max} is determined by 1) fog density level parameter β 2) Lidar sensor parameter n, g 3) Point Intensity L_{fog}.

New distance d_{new} is proposed to decide fog cloud position as equation (3.2). P_{lost} is lost probability of point cloud and function of β and d_{max}. As the product of β and d_{max} increases, P_{lost} decreases. After generate fog point by fog model, Intensities of original points are renewed by β and d_{max} as equation 3.4. Fog points generated in designed distance with designed probability

$$d_{\text{new}} = \frac{-\ln(0.5)}{\beta} \tag{3.3}$$

$$P_{\text{lost}} = 1 - e^{-(\beta \cdot d_{\max})} \tag{3.4}$$

$$Int_{new} = Int_{origin} \cdot (-e^{-(\beta \cdot d_{\max})}) \tag{3.5}$$

$$P_{fog} = (1 - P_{lost}) \cap (d < d_{\max}) \tag{3.6}$$

The fog model is summarized as follows.

1) Set the maximum sensing distance in real-time (determined by the degree of foggy, lidar type, and intensity of all points)

2) Empirically determine the location of the new fog spray (determined by the degree of foggy)

3) Determine the probability of annihilation of points by distance (determined by the degree of foggy and the maximum sensing distance)

4) After fog is created, the intensity of the existing points is reflected in part affected by the fog (determined by the intensity of the existing point, the degree of foggy, and the maximum sensing distance)

5) Model imitating the shape of fog and snow

The following figures shows the results of checking the shape of the fog model by scattering it in the simulator or actual driving situation using the fog model.

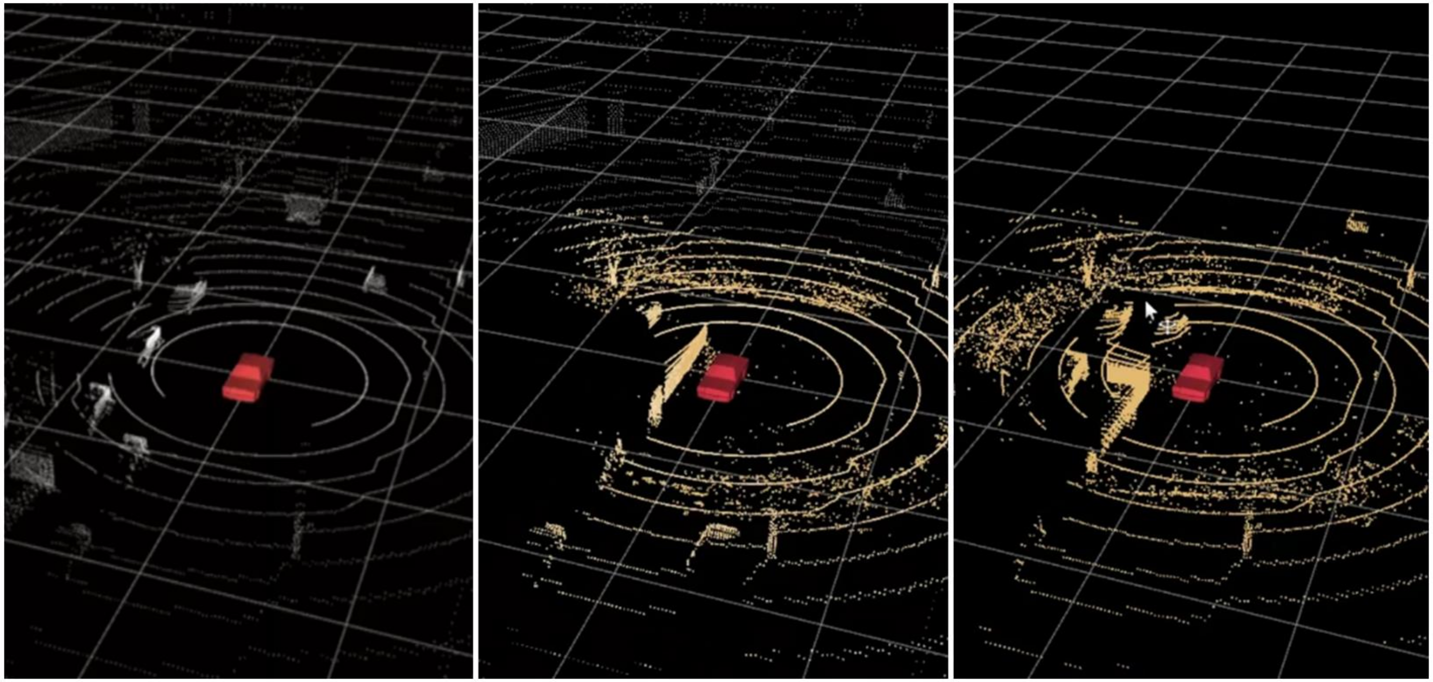


Figure 3.8. Fog model operation in Simulator Beta = 0.05, White : Original points, Yellow : Fog points & Fog affected point

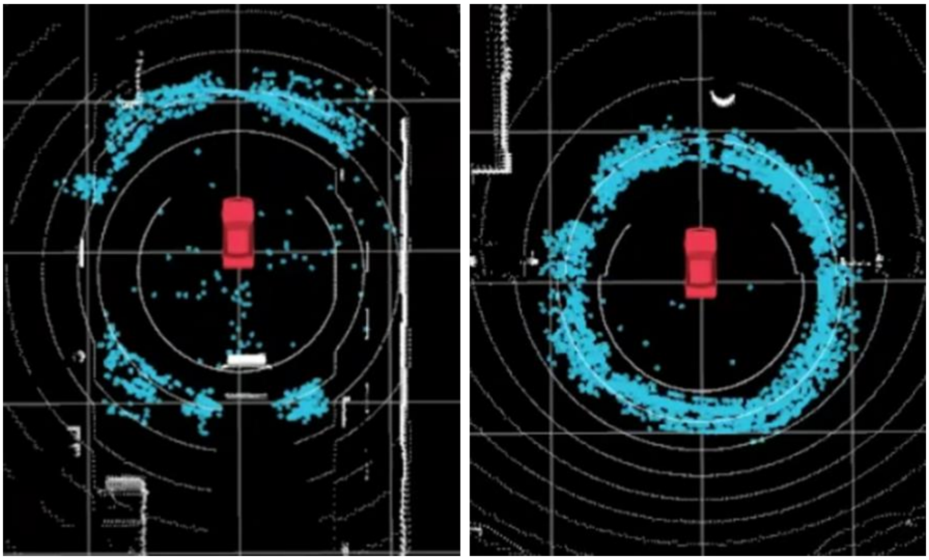


Figure 3.9. Fog model operation in Simulator with Bird-eye View (Left : Beta = 0.04, Right : Beta = 0.07)

Figure 3.8, and Figure 3.9 shows that with the change of β value, fog shape, fog affected point's sensing distance, max sensing distance changed.

The following shows the results of scattering the fog points using the fog model same as in the simulator. This is because if the fog shape in the actual driving situation is different from the fog in the simulator, the fog model could be suspected of its performance. Thus, the actual LiDAR sensor (Robosense 32) was mounted on the vehicle to obtain data. Then, the data was scattered to the fog model offline to see the result. Figure 3.10 is data acquiring vehicle with Robosense LiDAR.

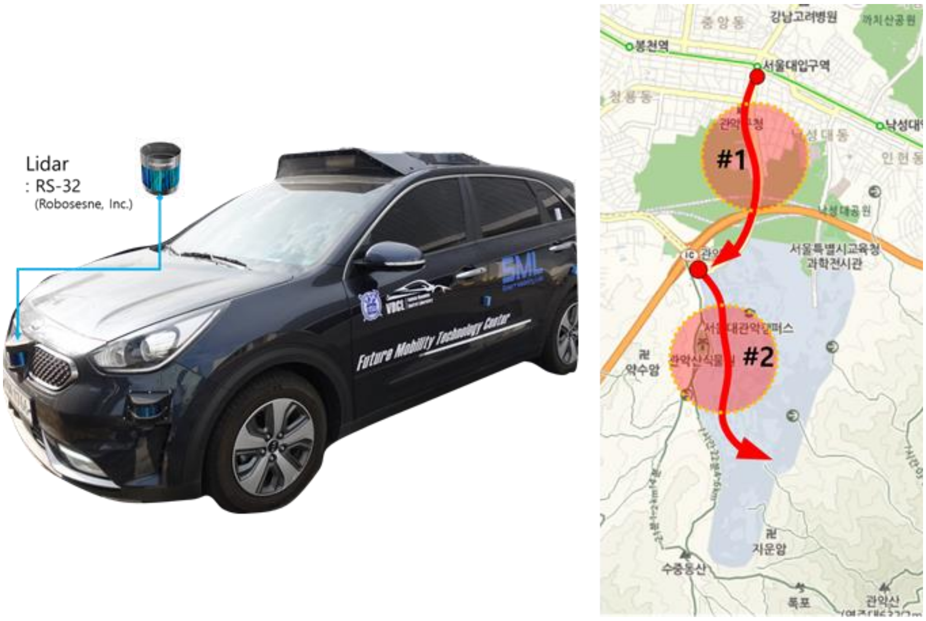


Figure 3.10. Test vehicle and Experimental setup and Data acquired area

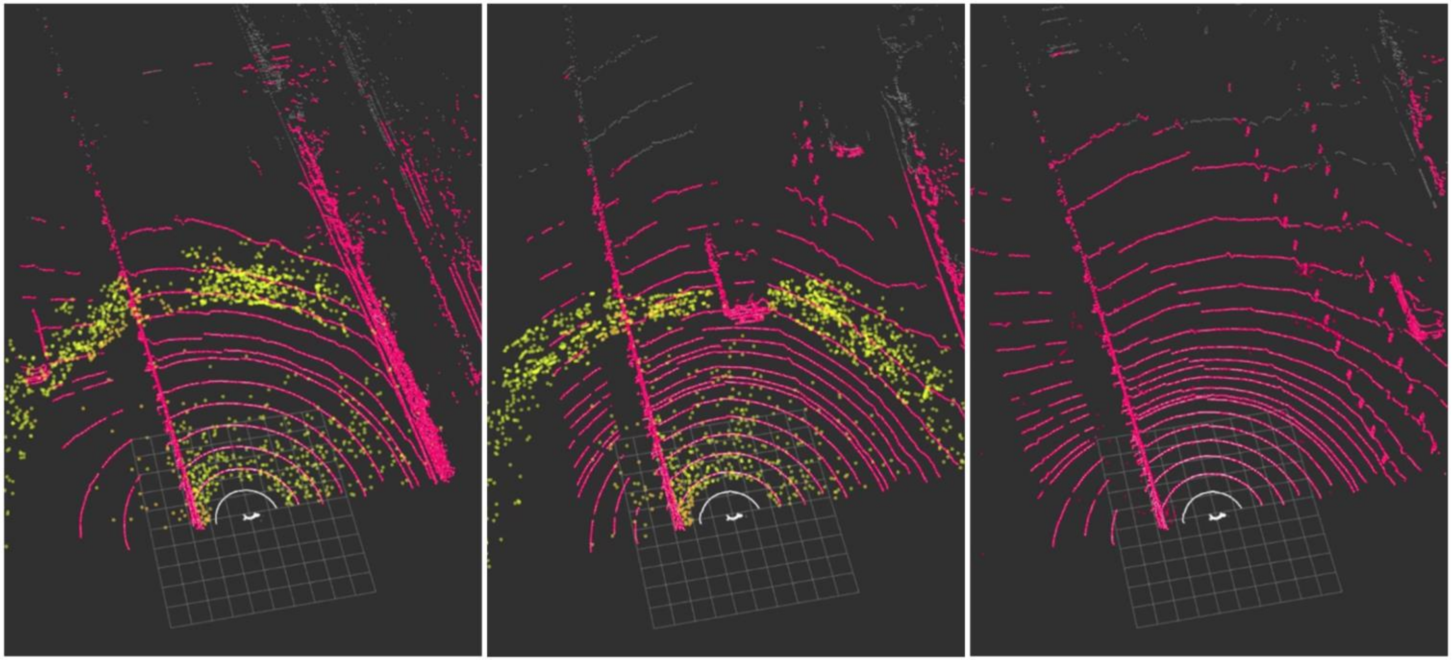


Figure 3.11. Fog model operation in Real-Driving Beta = 0.03, White : Original points, Yellow : Fog points , Magenta : Fog affected points / By comparing Figure 3.8 and Figure 3.9, scattering the fog model in the simulator and real driving data is similar to the fog shape.

3.4 Perception error analysis due to Fog points

This sub-chapter studied how the perception system is affected by fog. If the perception performance in fog conditions is the same as in clear weather, fog de-noising should not be developed anymore. Therefore, we started testing with curiosity about this part.

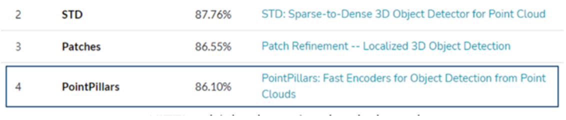
The test for this part is divided into two parts.

1. Testing the use of the fog model in a simulation environment
2. Test in a real-life fog situation

In order to know whether the perception system is operating normally, it is necessary to apply the perception algorithm used as a reference. Therefore, in this test, a widely used algorithm that guarantees performance and real-time was used. Object detection was carried out by applying a reference algorithm called PointPillar [Lang'19], and it has the following characteristics.

- Detect objects in point clouds through sequential point-net, CNN, and SSD neural networks
- High detection performance (KITTI object detection task 5th place)
- Real-time inference (~62Hz)
- Open Source code

Moreover, the performance is as shown in the figures below.



2	**STD**	87.76%	STD: Sparse-to-Dense 3D Object Detector for Point Cloud
3	**Patches**	86.55%	Patch Refinement -- Localized 3D Object Detection
4	**PointPillars**	86.10%	PointPillars: Fast Encoders for Object Detection from Point Clouds

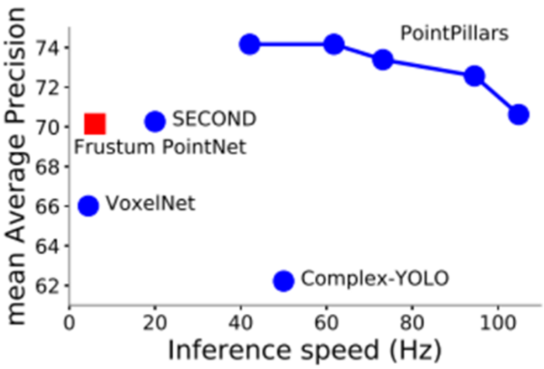


Figure 3.12. PointPillar KITTI vehicle detection leaderboard, and performance speed and mAP

These performance is proper to evaluate the 3d point based object detection algorithm.

Un/Mis-recognition cases in Real-driving Fog. This situation is a dataset acquired while driving around by placing a vehicle in front in an actual heavy fog situation. Detailed video content has been uploaded to Youtube[Lee’22(f)]. In this subchapter, the situation is explained by attaching a snapshot of videos.

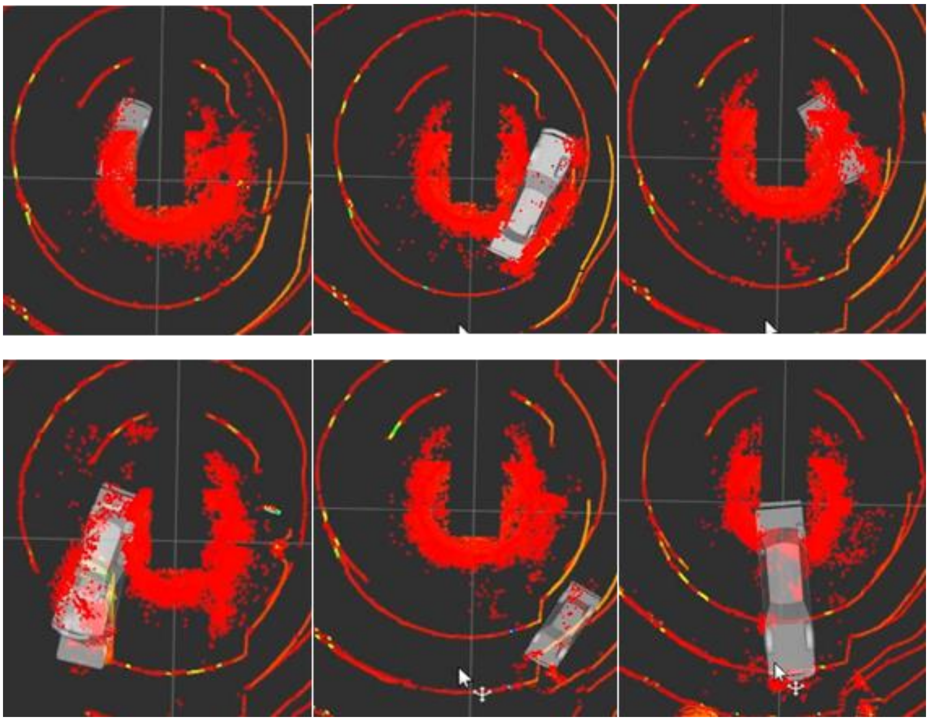


Figure 3.13. Mis-recognition cases in Real Fog

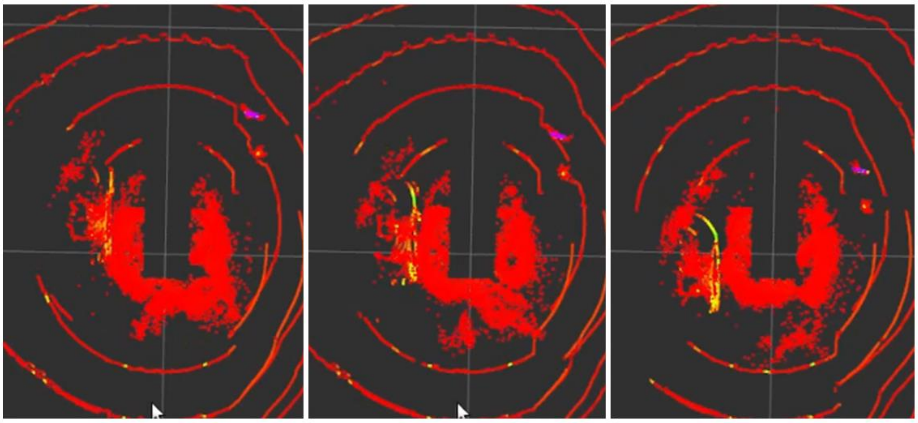


Figure 3.14. Un-recognition cases in Real Fog

It was found that the un/misrecognized cases occur in real-life heavy fog situations. It is not easy to quantitatively discuss the fog density of the day. However, Driving visibility is less than 20 to 35 meters, Korea Meteorological Administration officially announced on November 20, 2021.

The content will be discussed as follows when the fog model is used; in several cases, videos of un/miscognition are uploaded to the youtube channel. Moreover, the capture case of the video is depicted below[Lee’22(f)].

Un/Mis-recognition cases with Fog Model.

We wanted to know how the detection result by the Fog model changes qualitatively using the Fog model. Therefore, the fog model was applied according to the change of beta value in LGSVL simulator. Moreover, at the same time, we operated the PointPillar network and saw the detection result. The first 30 seconds of the Youtube video [Lee'22(f)] are the detection results for vehicles, and from 30 seconds onwards are the detection results for people.

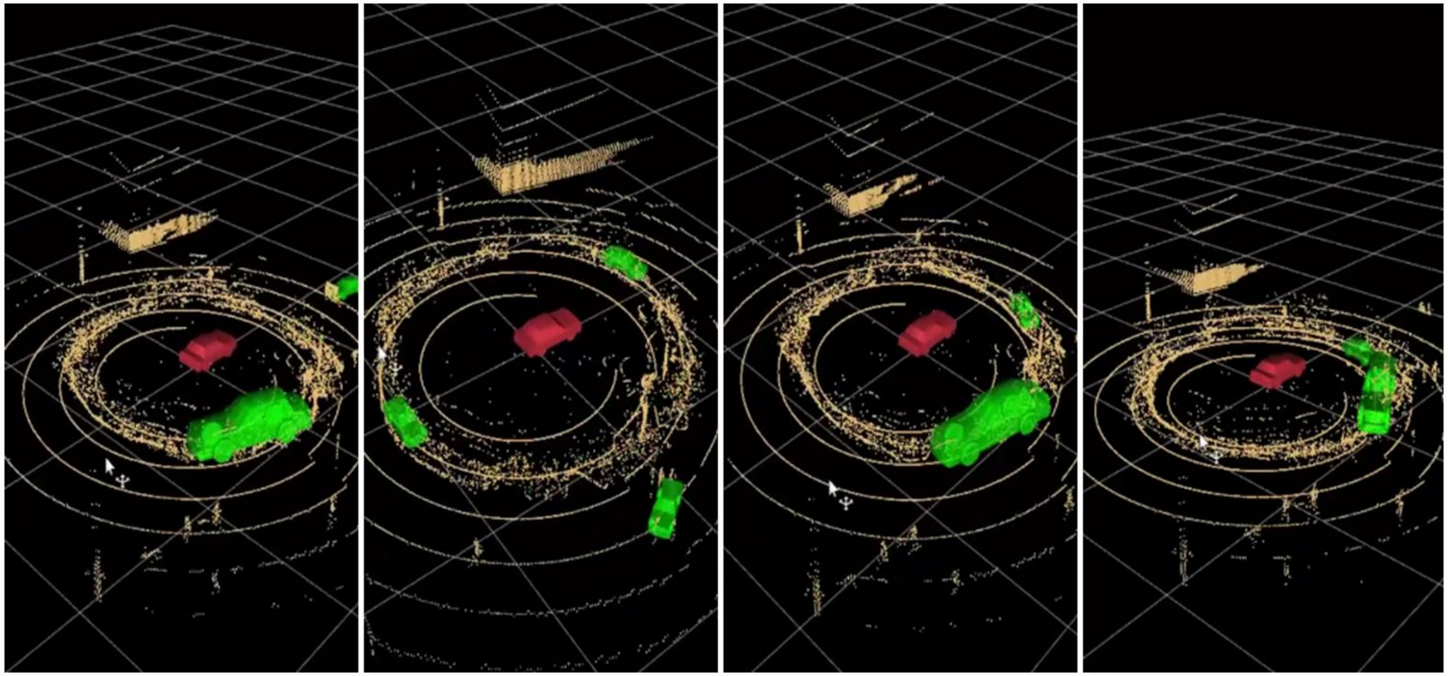


Figure 3.15. Mis-Recognition cases, Yellow points : Fog & Fog affected points, Green vehicle : PointPillar Detection result

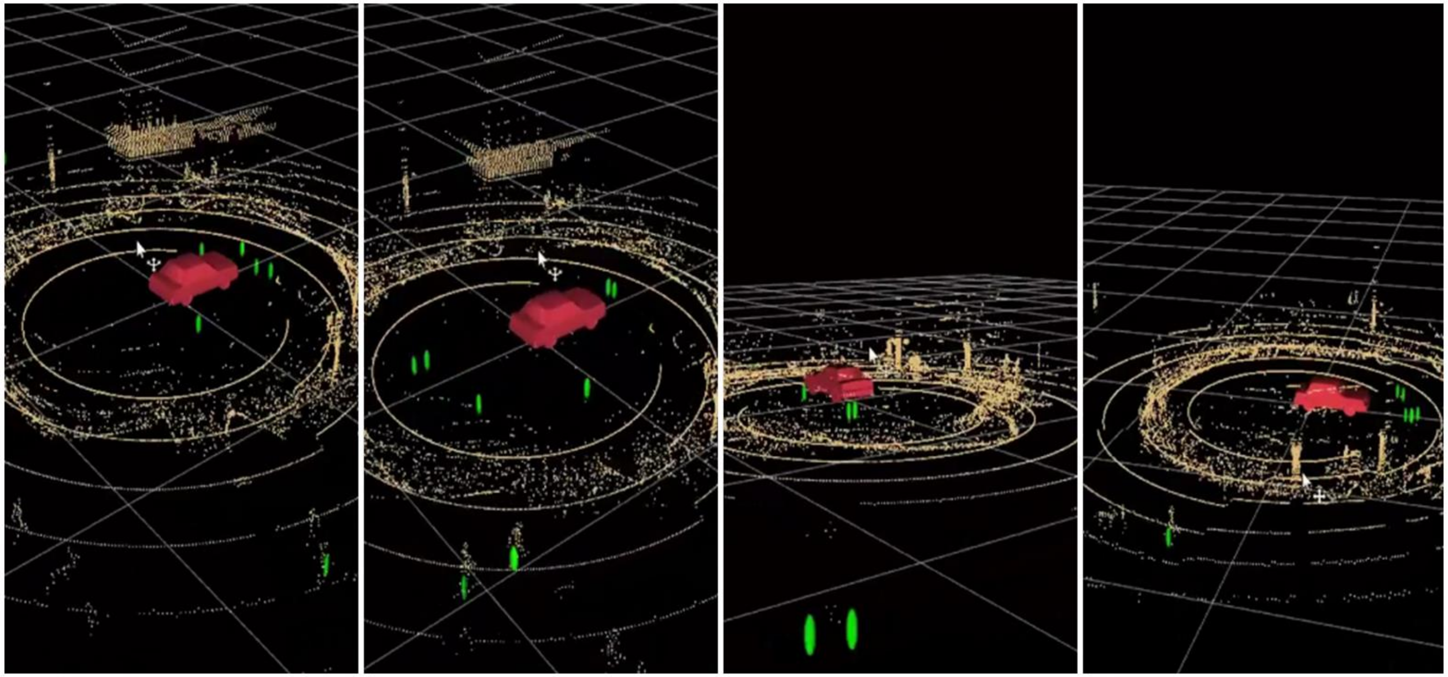


Figure 3.16. Mis-Recognition & Un-Recognition cases, Yellow points : Fog & Fog affected points, Green vehicle : PointPillar Detection result

3.5 Details about training

Overfitting verification

The method of preventing overfitting is as follows.

1) How to check the overfitting point: Evaluate the test datasets for each set epoch. The reason is that it is impossible to know if overfitting occurs only with train loss. Thus, empirically, after a few epochs, the result values are compared with the validation dataset.

2) Technic Method: Use augmentation techniques to increase data diversity or design optimization parameters to change over time. In the case of a network, adding a random element is also a representative example. Because supervised learning relies on large amounts of data, overfitting should be avoided. Overfitting has a high variance for other datasets when intensively trained on specific data. Since overfitting may occur if the amount of data is small, the higher the number of learning data, the more advantageous it is to learn. In this paper, feature augmentation was conducted. The features vector is obtained using distance information of nearby points using geometrical information and intensity. This our network could be likely to be overfitted. Therefore, two sets of feature aggregators in our network are designed to achieve anti-overfitting and efficiency. The feature aggregator executes effectively, preserves local geometric information, and increases receptive fields.

In this paper, feature augmentation was conducted. The features vector is obtained using distance information of nearby points using geometrical information and intensity. It uses only the neighboring points around each point and the encoding of positional information. The AFE used is based on vector self-attention and uses subtraction for relation. A Typical Attentive mechanism is used in the AFE module. $\{\boldsymbol{R}_i^k, \widehat{\boldsymbol{F}}_i^k\}$ feature set product with After softmax activated function matrix. After that, sum up the weighted Vectors. Computing attention scores; Given a local feature set $\widehat{\boldsymbol{F}}_i^k$, we design a shared function G() to learn a unique attentive score for each feature. The function G() consists of sMLPs and softmax, where W is the learnable weight of sMLP. In weighted sum part, the attentively learned scores are considered a soft mask that automatically selects essential features. It means that essential feature data with high weight is calculated through $\widehat{\boldsymbol{F}}_i^k$and calculated $\boldsymbol{S}_i^k$ through the weighted sum; After that, the mapped point data can be calculated.

Anti-sensor-specific learning

This part is essential in the learning process. This part corresponds to the two datasets which use real-driving fog data and fog model-based augmentation dataset in chapter 4.1.

The fog model could be changed by density with hyperparameters. In this paper, a total of three datasets were used. Among them, the second dataset corresponds to the fog model-based simulator LiDAR data + fog model-based real-world LiDAR data. The third dataset was acquired in real-driving fog data in the real world.

In the second data set using the fog model, the learning was conducted using a data set with different sensor configurations to prevent overfitting. It tends to prevent a network tailored to one type of sensor set and prevent sensor-specific algorithms. Kittidataset is utilized, and the data set is acquired with Velodyne HDL-64 Eller scanner, which has 64channels and could transfer 100k points in sensor frame. LiDAR mounting position is on the top of the vehicle. Moreover, this paper uses not only the kiti dataset but also the simulator dataset and the real driving dataset. In the case of the simulator data set, the simulator vehicle sensor configuration for data acquisition is shown in Figure 4.3. Data were obtained by mounting two 32-channel LiDARs on the top of the vehicle. In addition, the real-driving data acquisition sensor set used Robosense 32 channel lidar, and the sensor was mounted on the front of the vehicle to obtain data. A total of three datasets were used to derive quantitative evaluation data results. Thus, it was possible to prevent the overfitting part and solve the sensor-specific part.

For the third data set, real-driving fog data, several types of data sets do not exist, such as the second data set. The vehicle was used to obtain fog data with the sensor configuration, as shown in Figure 1.6. Thus, data were acquired using one vehicle, but data was acquired in many scenarios, and the fog density changed very rapidly. Thus, when using the data set, a learning epoch was conducted while checking the overfitting point among the above-mentioned overfitting verification methods.

Although it is not tested after learning every epoch, for example, 3-5 epochs are learned, and results are made with a validation set. The reason is that it is not known whether overfitting occurs when the training loss is reduced. This part was explained earlier in chapter 3.5.

Evaluation metrics & Loss function

The AP(Average precision) evaluation metric is commonly used evaluation metric in semantic segmentation. AP metric displays each frame for Recall and Precision. Then, sort the detection result according to the high confidence. Precision is a number that indicates how much a prediction is accurate. Moreover, Recall indicates how well the algorithm detects all positives.

$$\begin{aligned} &\text{Precision} = \frac{TP}{TP + FP} = 1 - FDR \\ &\text{Recall(sensitivity)} = \frac{TP}{TP + FN} = 1 - FNR \end{aligned} \tag{3.7}$$

Definitions of Precision(also called Positive predictive value, PPV) and Recall(also called True positive rate, sensitivity) are shown in equation 3.7. In table 3, there are four results of the detector. True negative refers to a situation in which we have not found the object that we have to find. False-negative refers to a situation in which the detector also did not find an object we did not want to find. False-positive refers to a situation in which the detector finds an object we do not want to find. In a true positive situation, the degree of success of the detector is expressed as an IoU(Intersection of Union) value. The details will be introduced in chapter 4.2 P-R Curve, ROC Curve, and F-score part.

Table 3. Definition of classification and four outcome table

		Predicted condition	
		Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True Positive (TP), hit	False Negative(FN), type2 error
	Negative (N)	False Positive(FP), type 1 error	True Negative(TN)

Setting We use the Adam optimizer as optimizer. Optimization techniques in learning are those that update learning parameters so that the cost value of the loss function is low, typically SGD (Stochastic Gradient Descent), SGD+Momentum, NAG (Nesterov Momentum), AdaGrad, Adam, Etc., which are most commonly used in this study. Adam is a collection of two-pronged optimizers and is the most commonly used optimizer. Based on the Momentum series centered on the direction, the index average of the slope is stored, and similar to the RMS Prop technique, the index average of the slope square value is stored. In addition, it is an Adam optimization technique that combines the Ada series centered on stride and appropriately adjusts the stride and direction[Kingma'17]. The initial parameters m and v of the Adam optimizer are initialized to zero and used. Alpha, or learning rate, is set to 0.001 and is reduced by 5% for each epoch. The remaining beta, beta1, and epsilon use 0.9, 0.999, and 10^(-8), respectively. The number of KNN points in K is 16. In test, the whole raw PCD is fed into network. However, in training process, we fed fixed number of points as input. This process was conducted on an NVIDIA RTX2070 GPU.

Chapter 4 Performance Evaluation

In this thesis, a total of three types of data were used. Quantitative and qualitative evaluation of three types of data is described, and only qualitative evaluation is derived about real-driving data that cannot be evaluated quantitatively.

4.1 Qualitative Analysis

About Chamber dataset CEMERIA's climate chamber is a large dataset containing four very realistic road scenarios[Bijelic'20]. Furthermore, data were obtained in a chamber that can control the weather, such as fog and rain [Gruber'19]. The climate control room data set contains 10-100 mm rain intensity fog visibility ranges of 15 mm, 33 mm, and 55 mm/h. Rainfall is controlled by a proximity loop at a certain level. The fog record was recorded up to 100 m during continuous dissipation and measurement of the actual visibility provided by the reference system of the climate chamber, starting with a visibility of 10 m[Heinzler'20]. This procedure allows for very accurate determination of visibility and has been repeated three times to produce more samples for each meteorological visibility.

Moreover, regarding labeling, after obtaining the ground truth position of objects by obtaining lidar data in a clear situation, auto labeling of fog and rain is performed through weather control. This is a significant advantage of the chamber dataset. Therefore, point-wise weather labeling is possible. The chamber dataset contains a total of 72,800 data. Then, take the ratio of training, validation, and test set to 7:1:2.

Chamber data evaluation The qualitative evaluation results of chamber dataset are shown in figures 4.1, 4.2. Figure 4.2. is the video captured picture. In qualitative evaluation sub-section, detailed prediction results will be discussed.

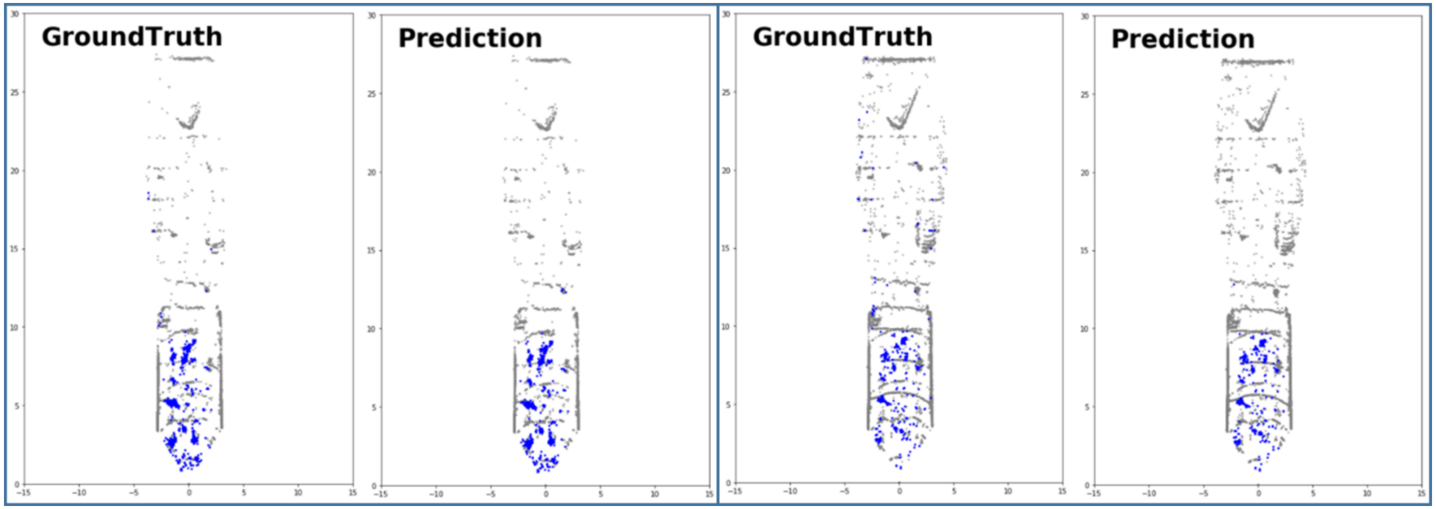


Figure 4.1. Prediction result with Chamber data (Blue Point : Fog points , Gray Point : Valid points).

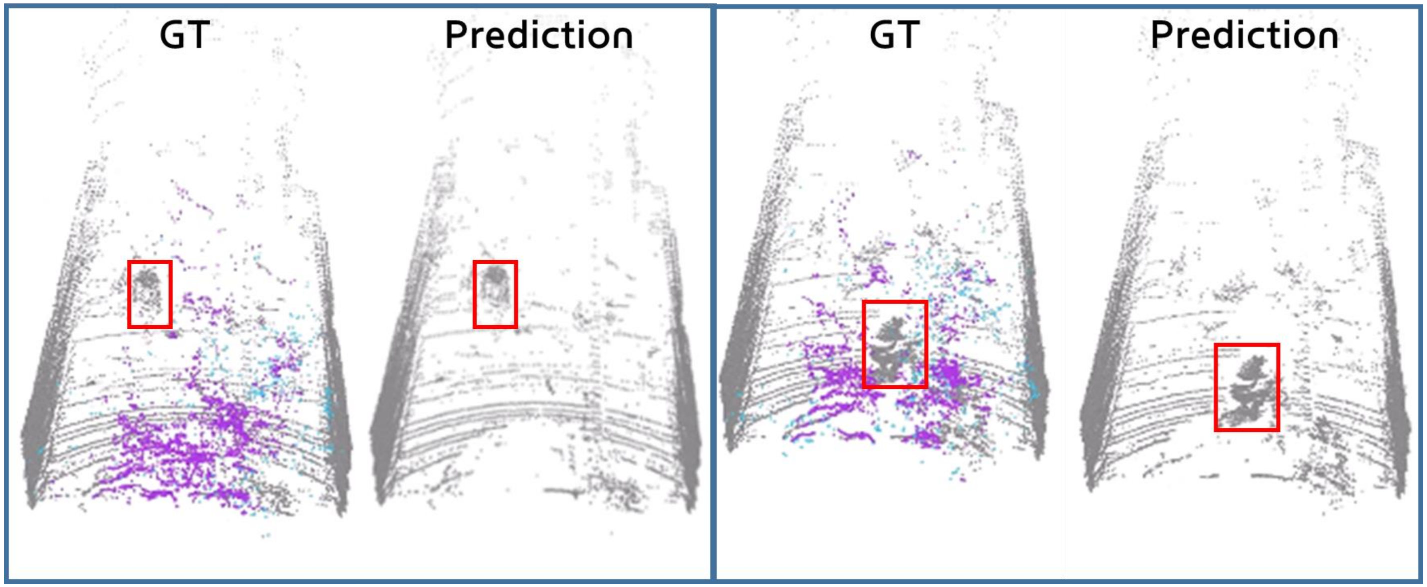


Figure 4.2. Prediction result with Chamber data, Bicycle riding person in Chamber (Magenta Point : Fog points , Blue Point : Rain, Gray Point : Valid points).

Fog model based LGSVL simulator vehicle configuration When acquiring data from the simulator, when setting the sensor configuration of the vehicle, the same sensor configuration as the actual test vehicle was selected. As in Figure 4.3, two Velodyne 32-c puck sensors were installed at the height of 1.5m.

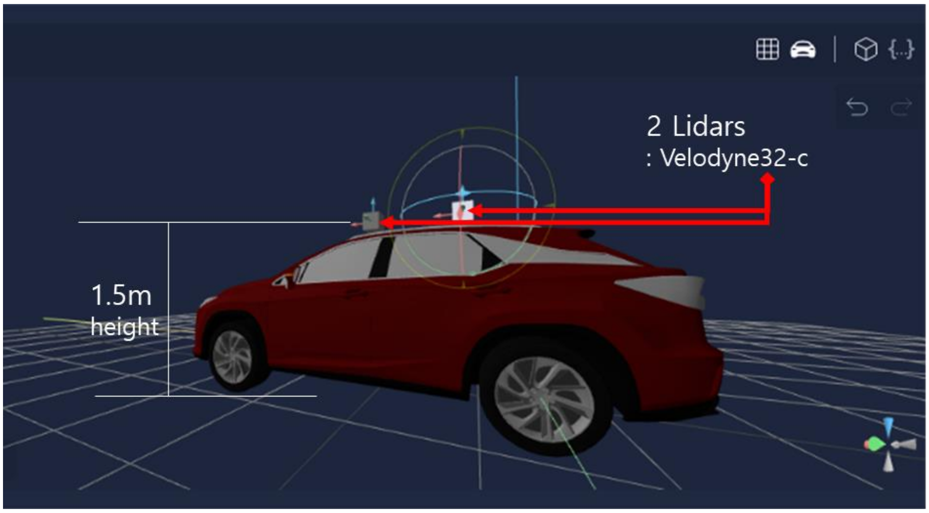


Figure 4.3. Simulator vehicle configuration

Fog model used LGSVL simulator based evaluation The qualitative evaluation results of simulator with fog model are shown in figures 4.4, 4.5, 4.6.

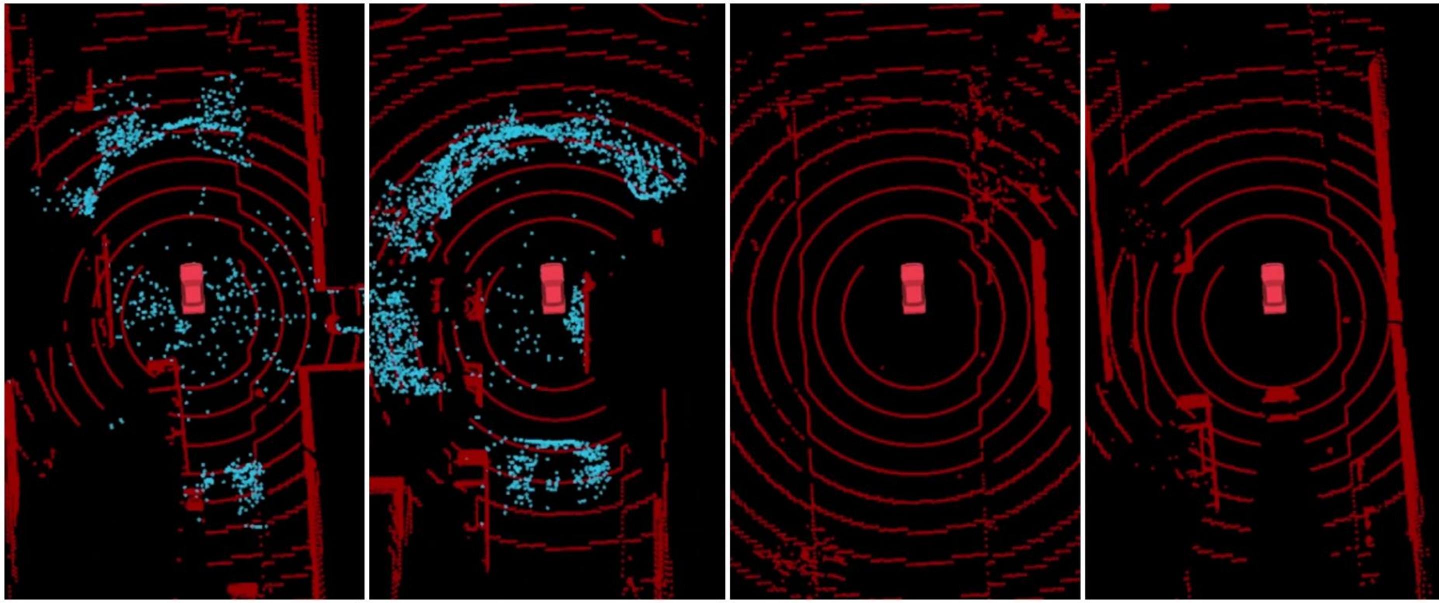


Figure 4.4. De-noising result with Fog model (Red Points : Left points after denoising , Blue Points : Fog model scattering points).

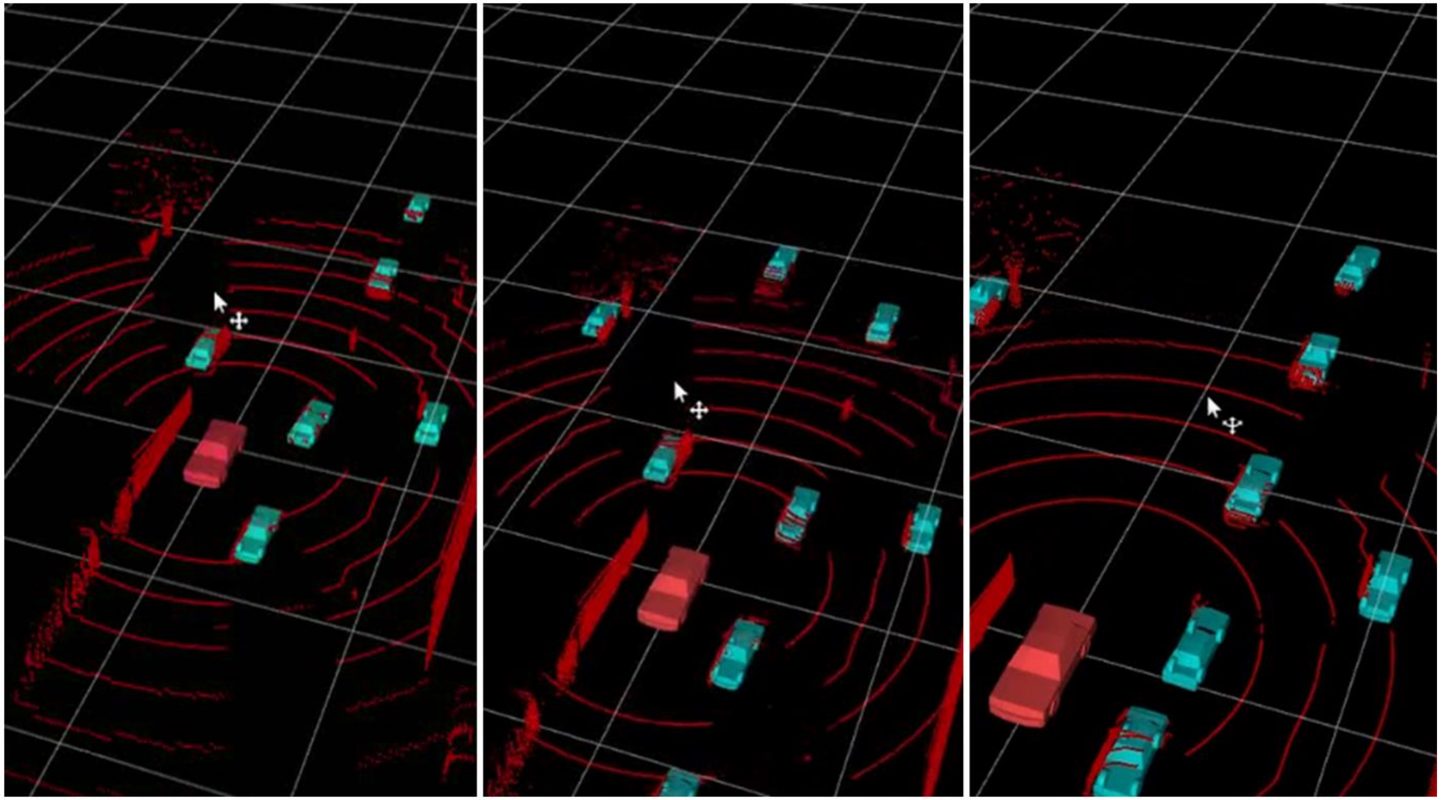


Figure 4.5. De-noising result & PP based object detection result with Fog model (Red Points : Left points after

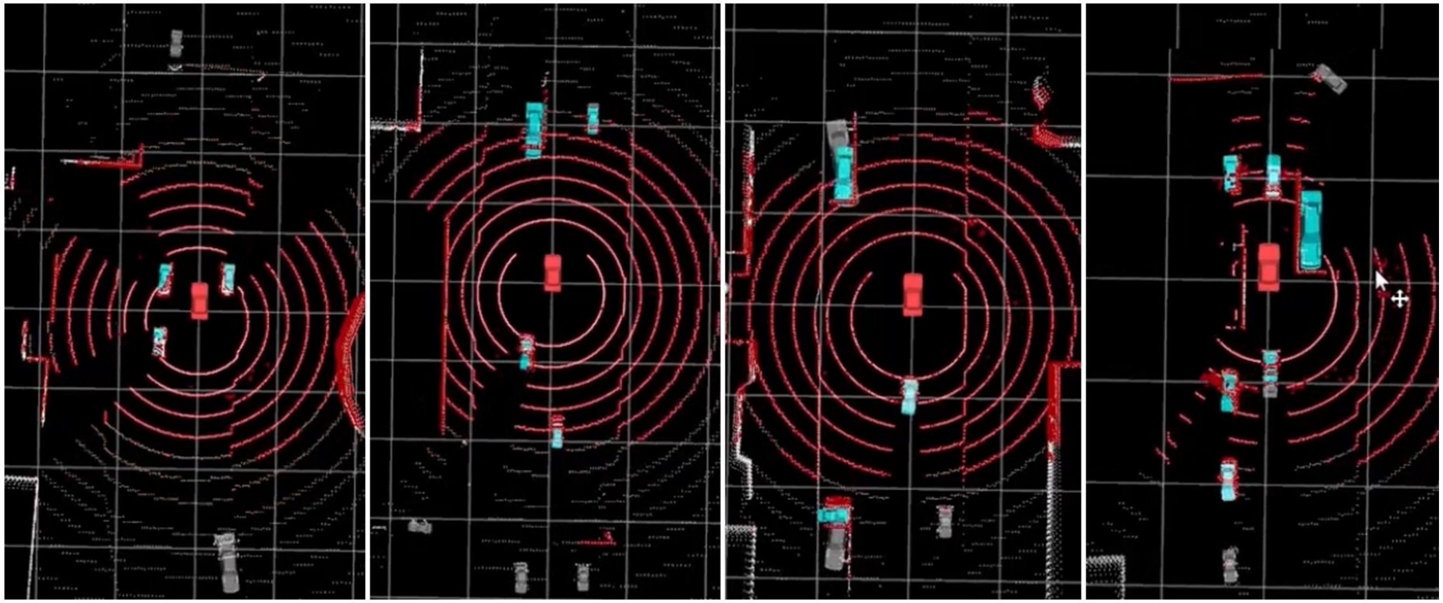


Figure 4.6. De-noising result & PP based object detection result with Fog model (Red Points : Left points after denoising , Blue vehicle : Left Red Points based PP vehicle detected result, White vehicle : Original raw Points based PP vehicle detected result)

Evaluation results Figure 4.4 shows the result of displaying valid points as red points after de-noising. Figures 3.8, 3.9, and 3.11 show that numerous valid points are covered by the shape of fog when viewing raw data + fog model results through fog model operation. In addition, it can be seen from Figures 3.13-3.16 that the un/misrecognition result occurs in the perception system due to the real fog or fog model. However, in Figure 4.4, the red points show a very clear shape compared to other figures. Moreover, in Figure 4.5, it can be seen that qualitative perception results are satisfactory when the de-noising + Point Pillar detector is used. Compared to Figure 3.13-3.16, there will be a clear difference. Furthermore, in figure 4.6, it can be seen that the distance of the valid points decreased due to the fog model. In Figure 4.6, the blue vehicle is a detector based on red points (points remaining after de-noising). In Figure 4.6, the white vehicle is the result of the raw lidar data-based detector. It seems clear that the cognitive distance has decreased due to the use of the fog model. However, the mis/un-recognition phenomenon is not shown in the detector of red points.

All of the above figures are the capture results of the video result. Thus, the video results for all captures were uploaded to the YouTube channel.

4.2 Quantitative Analysis

Chamber data based evaluation

Table 4. Results on the Chamber data set, Mean Intersection-over-Union(mIoU) comparison with other published strong baselines about Chamber data.

Model	IoU				Runtime	Parameter
	mIOU(%)	Fog	Clear	Rain	ms	Mio
DROR [Charron'18]	34.15	6.94	88.13	7.37	100	$4e^{-6}$
RangeNet53 [Milioto'19]	81.09	77.32	74.73	91.22	51.90	66.17
RangeNet21 [Milioto'19]	76.35	71.40	71.53	86.13	**33.83**	38.5
LiLaNet [Piewak'19]	83.48	79.57	82.75	88.16	91.93	7.84
WeatherNet [Heinzler'20]	89.11	86.40	91.65	89.29	34.95	1.53
Ours	**90.6**	**88.30**	**92.20**	**91.30**	60.00	**1.23**

Fog model used LGSVL simulator based evaluation

Quantitative evaluation is possible in the simulator. This is because labeling raw point data and fog model points are possible with the fog model.

However, as shown in Table 4, comparison between data is not possible. This is because there are no comparative studies using this data. In this study, the mIoU about fog is 90.31% when compared as equation (4.1). From the qualitative comparison results, it can be seen that there are many visible changes.

$$\text{mIoU} = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FP_c + FN_c} \tag{4.1}$$

where TP_c , FP_c , FN_c represents true positive, false positive and false negative predictions respectively for class c, and C is the number of classes, which is 3 or 2 in this thesis. As defined in the equation, the mIoU is the mean value of IoU over all classes.

P-R Curve, ROC Curve, and F-score

The term Precision(positive predictive value) is used for accuracy. Recall (sensitivity) indicates how well the target objects are captured without omission, and precision indicates how accurate the detected result is, that is, how much the actual object is contained in the detection results.

When deriving the results, a comparison was conducted with the situation of using one FA module as an ablation study. It was verified that the AUC value was slightly higher with the proposed method.

We also derived the F1 score and F beta score. F-score means that precision and recall have a trade-off relationship with each other, so it is a method of expressing it as a single value through average. The process of making it close to the lower two values using the harmonic mean is performed. The Harmonic mean is calculated so that the more extreme the value used in the mean calculation, the more likely the penalty is to be close to the smaller value. Moreover, it is a value determined by thinking that F beta recall is twice as important as precision. For example, if we think the recall is twice as important as precision, set the beta value to 2. In this paper, when calculating the F beta value, the beta was set to 2 and 0.5.

The results are shown as follows figures:

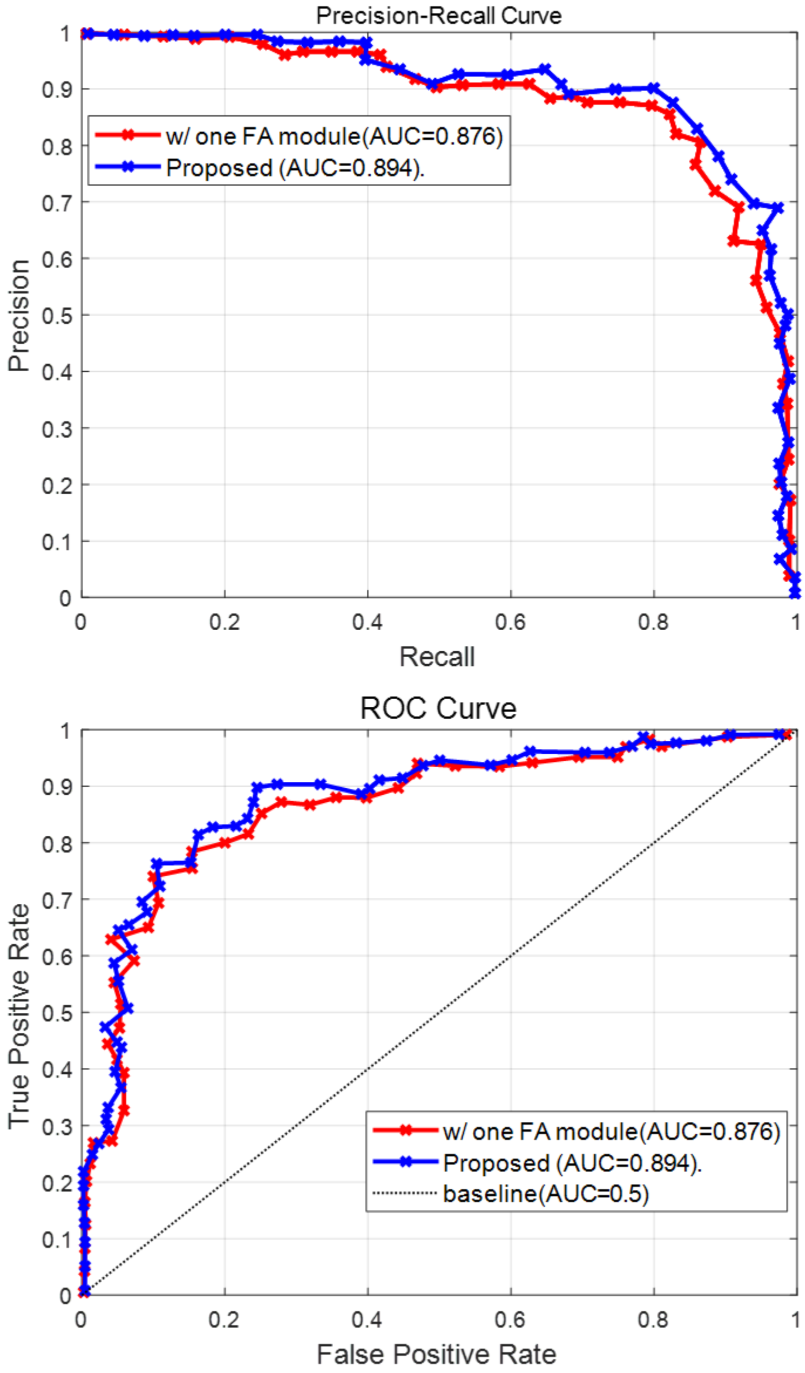


Figure 4.7. PR curve and ROC curve figure with AUC value

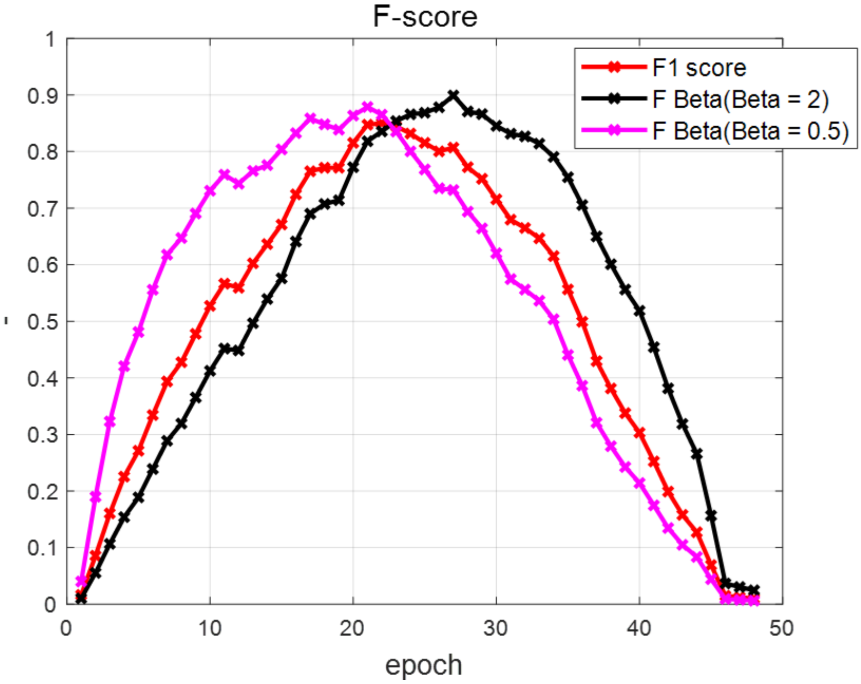


Figure 4.8. F1 and F beta values with changing Beta value

Table 5. Results on the fog model dataset, Mean Intersection-over-Union(mIoU) comparison with other published strong baselines about fog model(augmentation).

Model	mIoU	Runtime	Parameter
	Fog	ms	Mio
DROR [Charron’18]	6.94	100	$4e^{-6}$
RangeNet53 [Milioto’19]	87.19	51.90	66.17
RangeNet21 [Milioto’19]	79.94	**33.83**	38.5
LiLaNet [Piewak’19]	88.74	91.93	7.84
WeatherNet [Heinzler’20]	88.81	34.95	1.53
Ours	**90.31**	45.00	**1.23**

4.3 Retraining with real-driving Fog data

Real-driving fog data was learned using the same network. The biggest problem in learning is the difficulty of point-wise labeling. Thus, point-wise labeling was performed to learn driving fog data. The following method was selected to reduce the error as much as possible.

1. Collect incoming points for each sensor frame.
2. Use height and x,y area to determine the area where fog points enter, as shown in Figure 4.7.
3. Extract geometrical information and intensity features of extracted points and label them separately
4. Label the remaining points with valid points except for fog points

In the same way as above, 5000 sets of data sets that clearly show fog points were extracted from real-driving fog data of 3 hours (500GB).

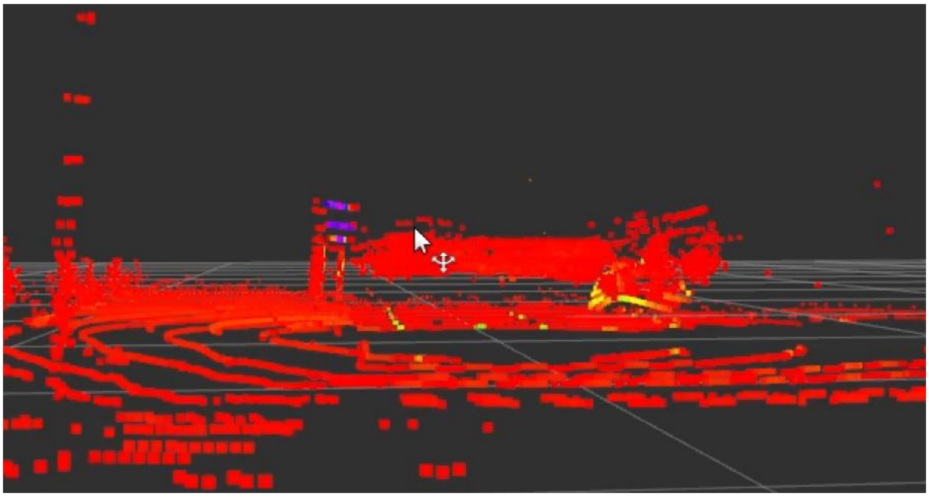


Figure 4.9. Fog points shape in real-driving data set

4.3.1 Qualitative Analysis

In the case of a real-driving fog situation, the fog model is not used. The network is directly operated using actual acquired lidar data. Then, check qualitatively how the results come out. Then, among the results of the video, the situation in which the vehicle is around and the result in the empty space are shown in Figures 4.8 and 4.9 below. Predicted fog and fog removed points are shown. It can be seen that the results are qualitatively erased well. Detailed result video is uploaded in author's YouTube channel.

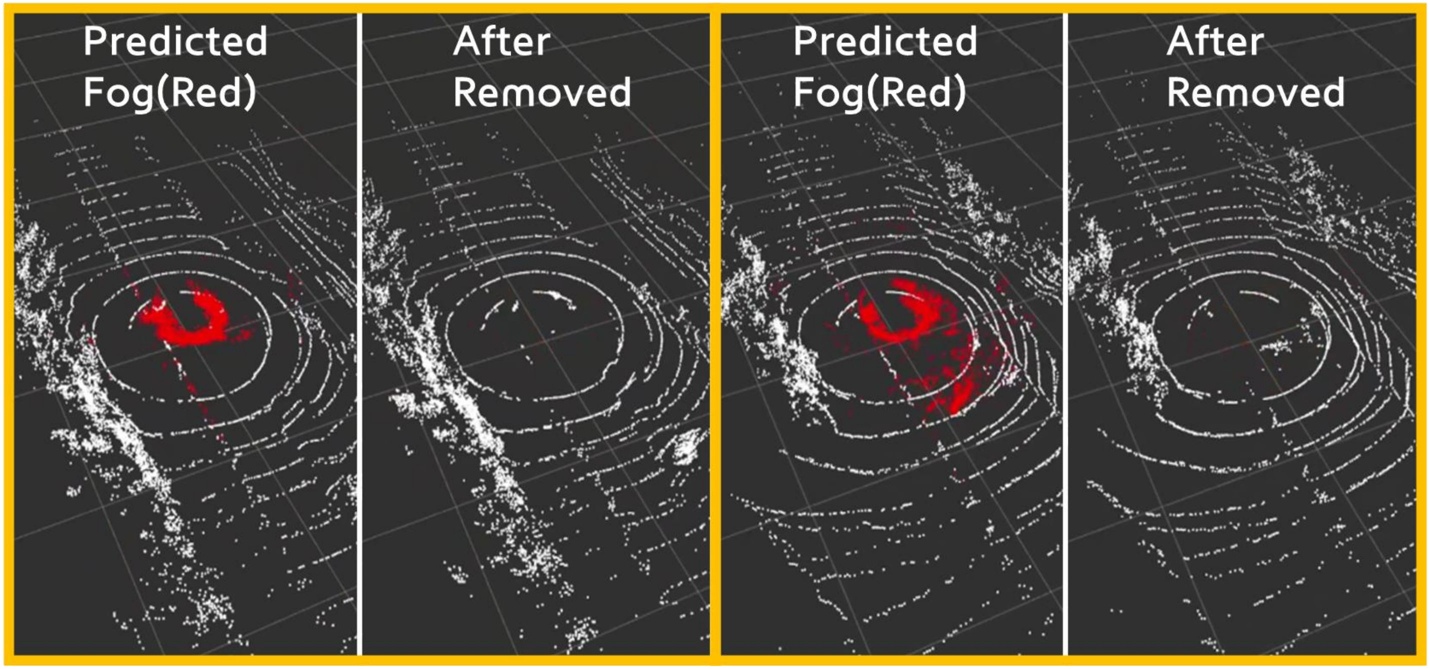


Figure 4.10. Real-driving fog based qualitative results (red points : network infereced fog points , white points : valid points)

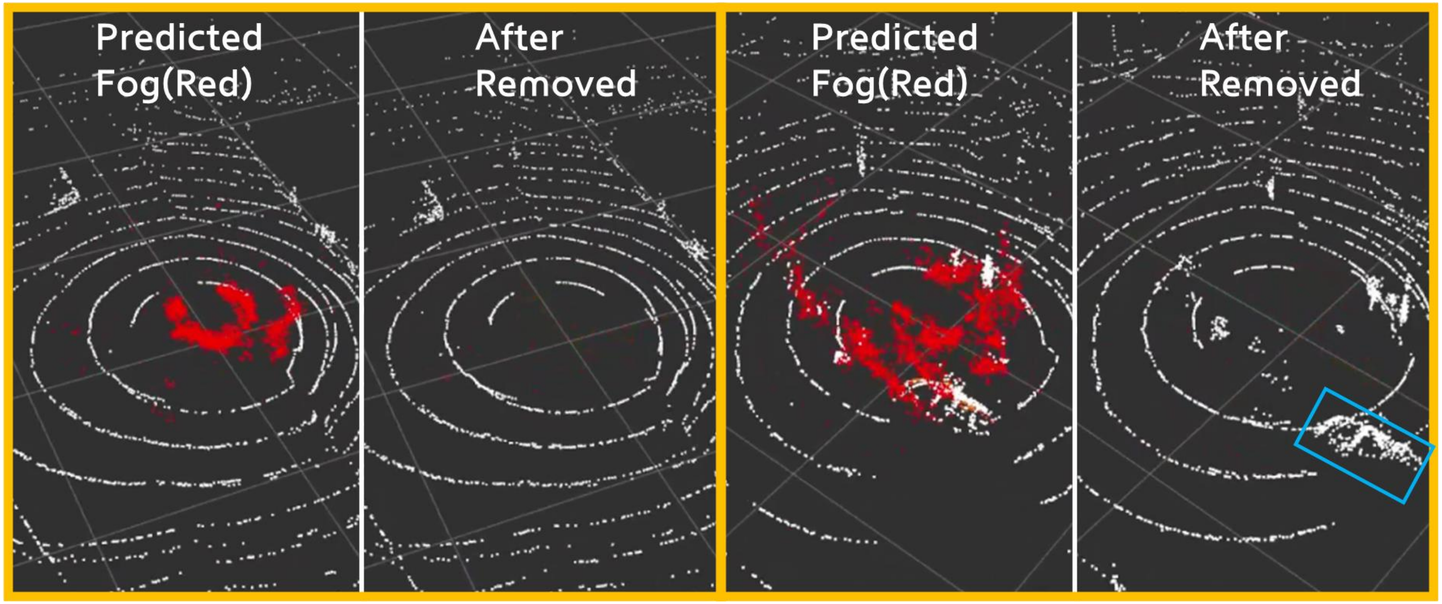


Figure 4.11. Failure cases on real-time test with additional training. It shows predicted boxes for car (pink), bus (red), unknown object (green).

Chapter 5 Max Detectible-range Estimation Module

The maximum detectible range of a vehicle is a crucial index in autonomous driving, especially in adverse weather conditions. Since particulates in the air have been shown to affect LiDAR visibility, there exists a need to determine the maximum detectible range, dependent on weather conditions. When driving in heavy fog situations, visibility with the naked eye decreases. In such situations, the Meteorological Administration classifies the severity of the fog through a "visible distance in fog" metric and recommends a reduced speed limit depending on fog density. This perspective inspired the development of the maximum detectible range module. The module was designed to be a pre-processing step to the perception and planning module, where perception range of the vehicle is obtained depending on the current severity of fog. Detailed structure of proposed module is depicted in Figure 5.1.

First, a reference perception algorithm was required to derive a perception range. This thesis uses a LiDAR point cloud Network-based perception algorithm called Point Pillars [Lang’19] as the reference perception algorithm. Point Pillars was chosen due to its high rank in detection performance (KITTI object detection task 5th), the fact that is it open-sourced, and that it was able to perform in real-time(62Hz).

The scene of the process of acquiring real-life heavy fog data acquired by the research team is as follows. Figure 5.2 shows a heavy dense fog driving scenarios capture scene. Detailed videos are uploaded on author’s YouTube channel [Lee’22(a)].

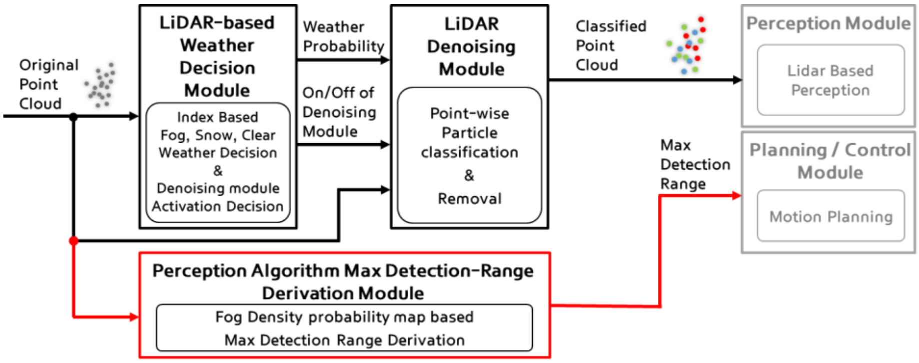

Original Point Cloud
LiDAR-based Weather Decision Module
Index Based Fog, Snow, Clear Weather Decision & Denoising module Activation Decision
Weather Probability
On/Off of Denoising Module
LiDAR Denoising Module
Point-wise Particle classification & Removal
Classified Point Cloud
Perception Module
Lidar Based Perception
Max Detection Range
Planning / Control Module
Motion Planning
Perception Algorithm Max Detection-Range Derivation Module
Fog Density probability map based Max Detection Range Derivation

Figure 5.1. Architecture of De-noising module and Detectible-range estimation module.

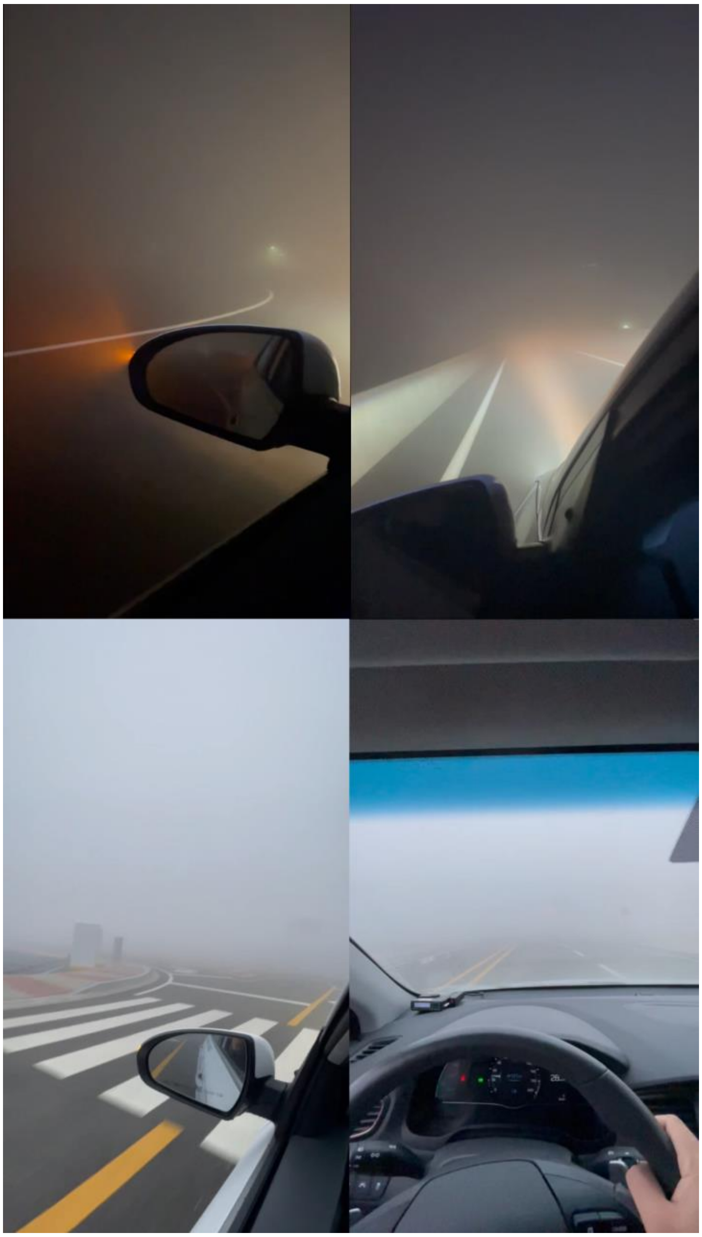


Figure 5.2. Dense Fog Video capture (Driving visibility is less than 20 to 35 meters, Korea Meteorological Administration officially announced on November 20, 2021) [Lee’22(a)]

The algorithm observed LiDAR raw data in foggy conditions before development. As illustrated in figure. 5.3, as a result of checking the data at the same time(around 8 pm) in the same place(in the test track), it was found that the intensity of point cloud, the circle-shaped fog, and the maximum reaching distance of LiDAR point was changed. After that, the question was whether the local area PCD’s density would affect the maximum perception distance of the perception algorithm of an autonomous vehicle.

Thus, the influence of surrounding points and perception distance in the fog situation was analyzed. Features for analysis are divided into three categories as follows.

1. In fog situations, the maximum arrival distance of points is reduced.
2. Ring-shaped points appear in the area around the LiDAR sensor.
3. The points move continuously, not fixed objects.

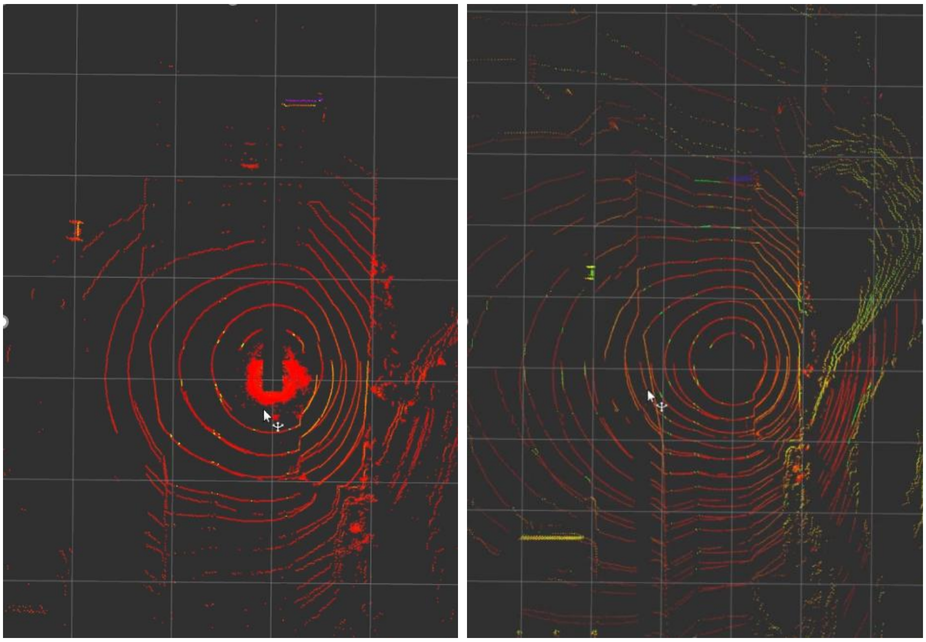


Figure 5.3. LiDAR points result during Fog/Clear at same location and same time (Left : Dense Fog , Right : Clear weather)

5.1 Maximum detectible-range estimation process

The maximum range estimation module was developed to quantify Fog density. While simultaneously estimating the detectible range in a dense fog scenario. A method of quantifying fog will be described, and the detailed process of deriving maximum detectible range is as follows:

1) When detection results of the PointPillars [Lang’19] algorithm is 50% or more, detectible range and point cloud data are stored simultaneously. Detailed scenes are depicted in Figure 5.4.

2) Omnidirectional point cloud data based on lane width (3.5m) is then extracted.

3) Data is converted into 2D to utilize the 2D Occupancy Grid Map method (OGM). Detailed figures are depicted in Figure 5.5.

4) Fog density is then quantified by summing probabilities for each pixel (grid) using Occupancy Grid Map(OGM). Fig 5.6. shows the results of 20 scenes.

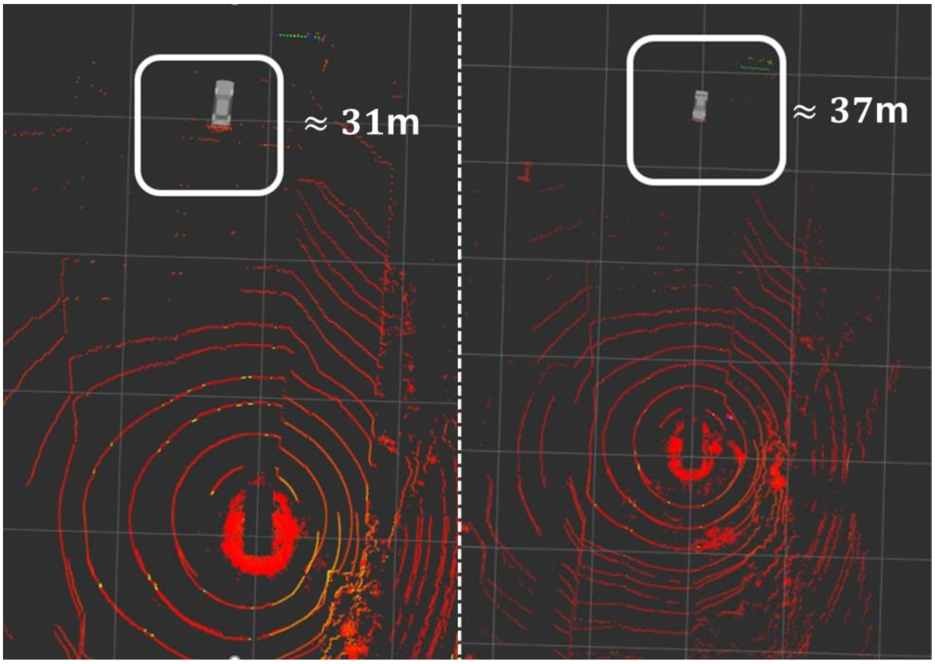


Figure 5.4. processing step one (Left: 31m detection result with PointPillars, Right: 37m detection result with PointPillars, Fog shapes are not the same in the two figures)

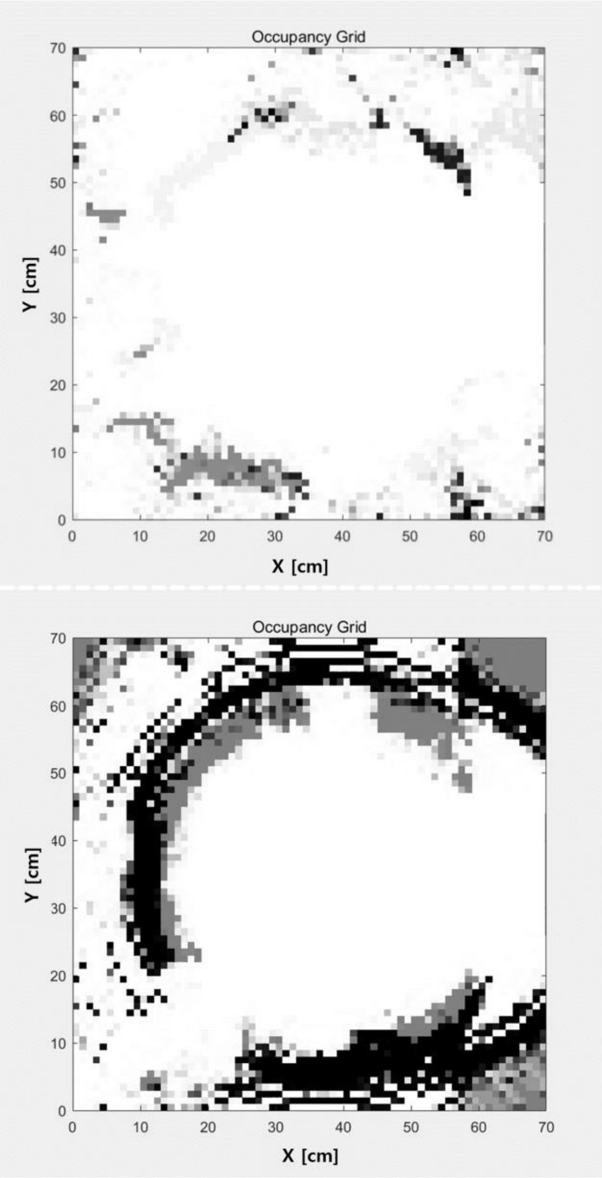


Figure 5.5. occupancy Grid Map values and figures, original videos are uploaded in Youtube [Lee’22(b)] (Up: max detectible range 37~42m OGM, Down: max detectible range 25~28m OGM shape, black=occupied (=1), white=free (=0))

In Figure 5.6, Among the real fog driving data, 20 Occupancy Grid Map (OGM) data were drawn from datasets that were able to recognize the vehicle. The x-axis represents fog density probability, and the y-axis represents the maximum detectible range. The ‘original data’ in Figure 5.6 represents the detectible range obtained through Figure 5.4 and the summed probability of the aperture grid map at the relevant moments, obtained through Figure 5.5. A little more numerical analysis was conducted in Figure 5.5. First, second, and third order fitting were performed, and the first fitting results were finally used in the vehicle. The fitting result is shown in Figure 5.7.

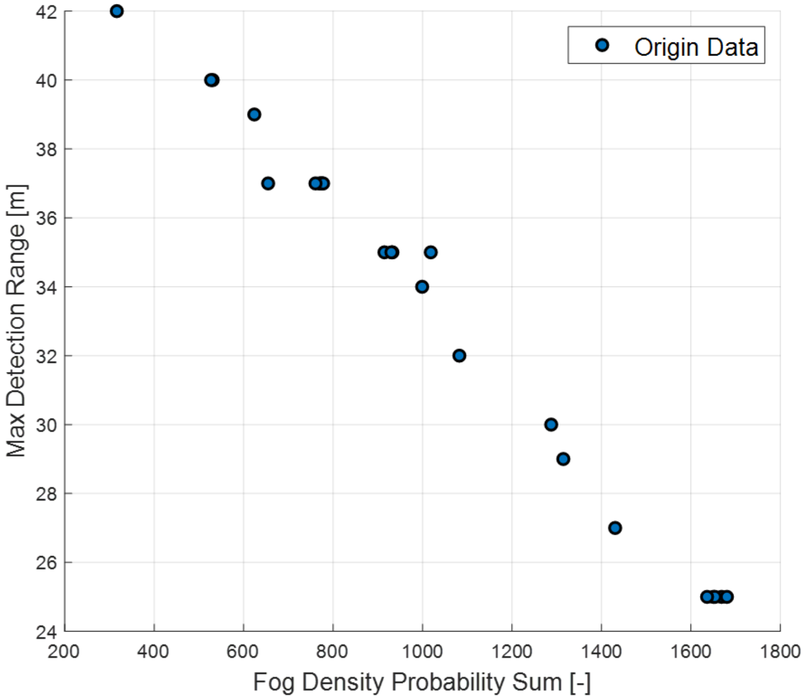


Figure 5.6. Fog density Sum and Max detectible range map

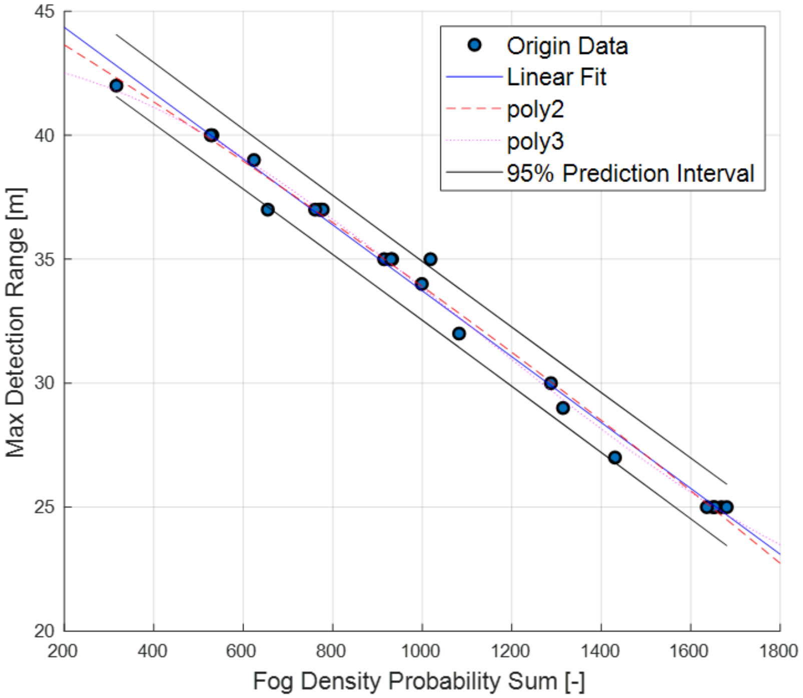


Figure 5.7. Fog density Sum and Max detectible range map with fitting

Linear, second-order, and third-order fittings were performed on the original data. From the linear fit, it could be confirmed that most points were within 2 sigma (95%) values of the fit. Hence, it can be said that the OGM's pixel probability sum and the PointPillars network (PP) detectible range for representing the fog density show a linear correlation. The SSE (Explained Sum of Square) value, R-squared value and the RMSE (Root Mean Square Error) values were determined to be 7.345, 0.989, and 0.577 respectively and shown in Table 6. Coefficients trust limit is 95%, P_1 and P_2 from $f(x) = P_1 \cdot x + P_2$ is -0.1329 and 47.01 independently. Detail information are described in Table 6.

Table 6. Results of first-order fitting

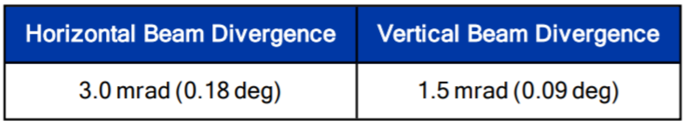
SSE (Explained sum of squared) $SSE = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2$	7.345
R-squared $R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$	0.989
RMSE (Root Mean Square Err) $RMSE = \sqrt{\sum_{i=1}^{n}\frac{(\hat{y}_i - \bar{y})^2}{n}}$	0.577

Furthermore, the presence of a map, the real-time detectible range value can be defined using the obtained linear fit result in fog situations. For this purpose, the proposed algorithm was developed with guaranteed real-time properties. R-squared means that this estimation has 98.9% of model explanation ability.

5.2 Characteristics of Lidar sensors in adverse weather conditions

After data acquisition, while observing the results of point cloud data, we discovered the physical characteristics of lidar that occur in heavy fog situations. It may be related to the type of sensor used. Therefore, we checked it through the Velodyne 32 manual[Velodyne'18], which sensor was used for data acquisition. As shown in Figure 5.8, the laser beam of 32 channel LiDAR laser is shot in the horizontal and vertical directions with a divergence of about 0.18 deg and 0.09 deg.

Furthermore, many people misunderstand the physical properties of LiDAR as follows. It is called a LiDAR “point” cloud, and one laser spot is shot from the transmitter and received from the receiver. However, referring to the manual, it shows the shape of ‘laser beam’ rather than ‘laser spot’ as shown in figure 5.9. [Velodyne'18] The sensor’s laser "beam" is in a small rectangular area, three small roller bars—9.5 mm tall by 12.7 mm wide.



Horizontal Beam Divergence	Vertical Beam Divergence
3.0 mrad (0.18 deg)	1.5 mrad (0.09 deg)

Figure 5.8. VLP-32C Beam Divergence [Velodyne'18]

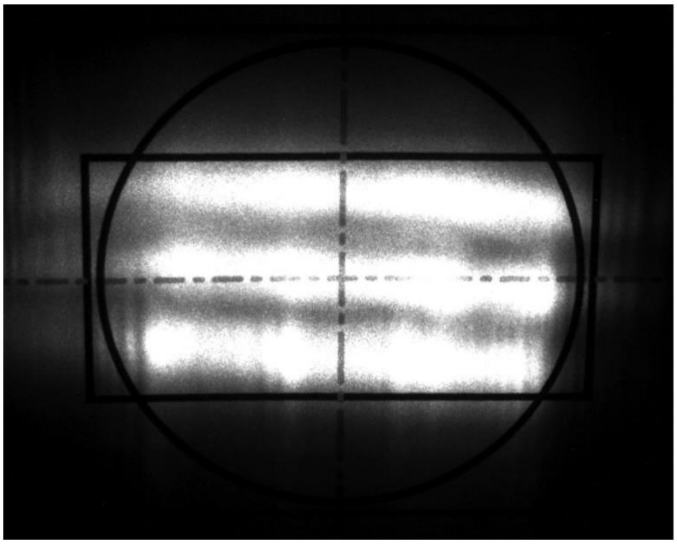


Figure 5.9. VLP-32C Laser Spot Shape (In reality, The sensor's laser "spot" is a small rectangular area, three smaller bars. 9.5mm tall by 12.7mm wide)[Velodyne'18]

From the standpoint of the LiDAR production company (Velodyne), the goal is to transmit the most significant point to users.

Figure 5.10. illustrates how the LiDAR's receiver selects the point data. Transmit the point with the strongest intensity or power.

The user could choose the hardware mode whether to take the strongest point, whole points, or have the last return point in LiDAR's hardware configuration. In acquiring data, we set "strongest return mode" set to default by Velodyne and acquired the data. The reason for this is that setting it to default can obtain the most meaningful data.

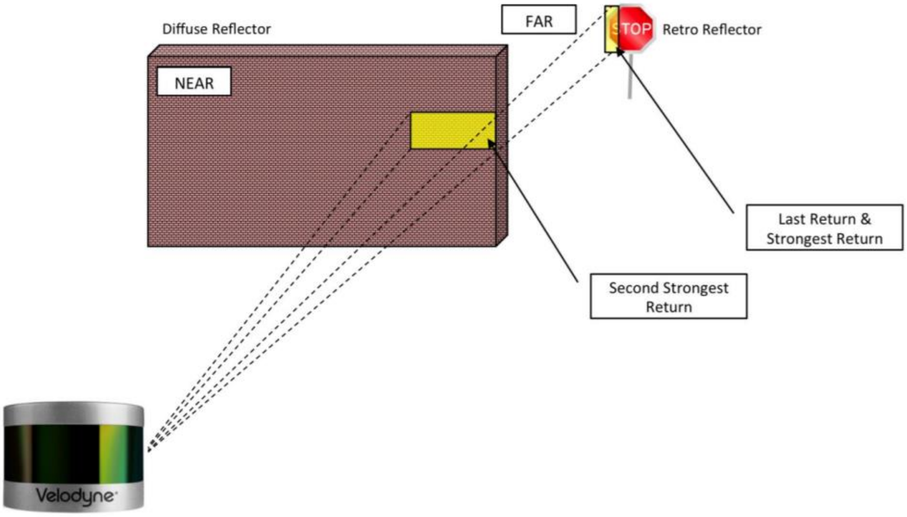


Figure 5.10. Velodyne LiDAR's mode description in manual

[Velodyne'18]

The phenomenon mentioned above was found during the acquisition of dense fog data. Figure 5.11 shows the result of selecting the hardware setting of the LiDAR sensor as "strongest return mode." For more information, refer to the author's YouTube channel.

To explain the phenomenon, there are trees close to the right at the beginning of the video(left figure). Thus, it can be seen that the density of the point is low in the right part of the ring-shaped fog points. However, the fog point density of the right side increases toward the second half of the video life right figure. To explain the reason, if there is an object nearby (Left figure), the laser beam's intensity is strong (the yellow part in config in figure 5.11), and points that reach the tree are sent to the user. However, in the right figure, the density of the ring-shaped points on the right side increased and the density on the back decreased.

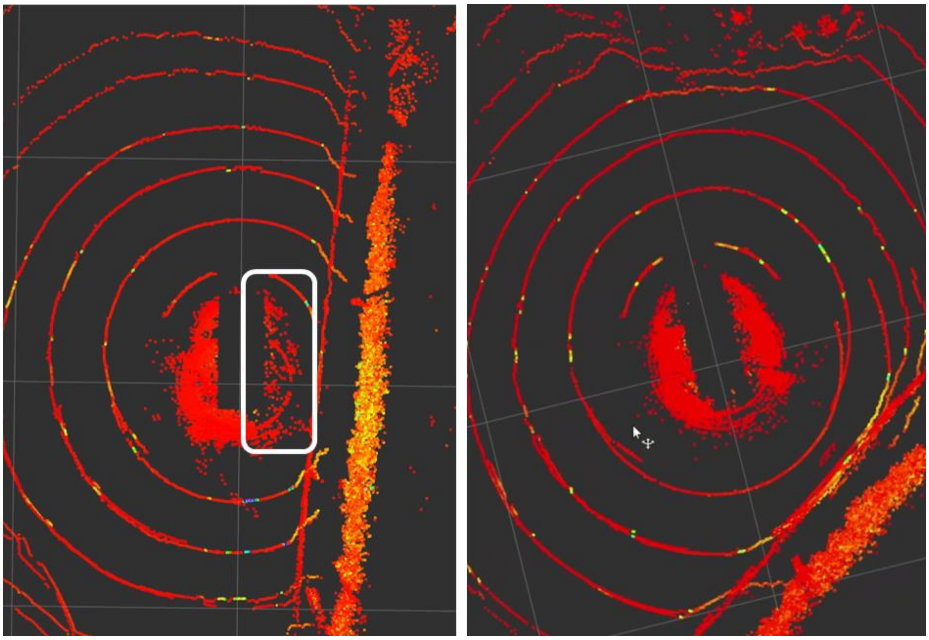


Figure 5.11. Ring-shaped fog points density change video capture (Left : beginning of video, Right : middle of video) [Lee’22(c)]

Chapter 6 Conclusions and Future Works

This thesis proposes a point-wise semantic segmentation module for an autonomous driving perception system. In front of the segmentation module is a point-based weather decision module. After the segmentation module, we propose a method to quantify the correlation between fog density and detectible range in fog situations. Therefore, an algorithm was developed to derive the maximum cognitive distance of the autonomous vehicle's perception system in fog conditions.

The three described modules operate in the vehicle in real-time as one system.

1. The most important part of the de-noising network was 1) real-time operation in the sensor frame and 2) geometrical information preserving. The two aspects correlate with each other. Real-time operation is impossible if the network is too deep to preserve geometrical information. Thus, we tried to find the optimal point. The random sampling method was used, and encoding and decoding layers were used. We used a K-Nearest Neighborhood (KNN) method using a Feature Aggregator. Thus, downsampling of the point is performed, but the point intensity feature is preserved to the receptive field for the geometrical information, and the feature is well expanded. Moreover, while performing the upsampling, use the attraction feature extraction (AFE) module to aggregate them to present the point's structural information registration. AFE uses only the neighboring points around each point and the encoding of positional

information.

2. Develop a Point cloud-based weather decision module and proceed with weather decision using only point. Data clustering is performed by obtaining point data of three weather conditions: Fog, rain, and clear. It is an algorithm that determines the current weather by using the difference between the data collected through k-mean clustering and the current weather data.

3. The maximum range estimation module was developed to quantify Fog density. While simultaneously estimating the detectible range in a dense fog scenario. A reference perception algorithm was required to derive a perception range. Finally, derived numerical relationship of fog density to detectible range.

In the validation part of the de-noising algorithm, the Chamber data set, fog model utilized simulator, and real-driving fog data was used. Quantitative and qualitative results were derived to validate the de-noising algorithm. Results show the significant de-noising results on three kinds of the dataset.

The occurrence of failure cases during operating 'De-noising module' in the vehicle was considered. The most tricky part was the occurrence of chatting in the Weather decision module, where more than 80% of the weather was decided. However, there was no problem when 80% was actually set to the threshold. The validated test data is a PCD obtained with driving data of about 3 hours.

The novelty of the study is summarized as follows.

1) Proposal of sequential algorithm structure only using Points Cloud Data

2) Real-time considered Network structure and methodologies were proposed

3) Adverse Weather's characteristic-based weather decision module was first proposed

4) Newly proposed Fog density ↔ detection range relationship

5) Three kinds of the dataset was used to evaluate the Network

5.1) Real-driving Fog data was first proposed and tested

Finally, to summarize the contents, a module as shown in Figure 6.1 below, was developed, and the result of operating in a vehicle with hardware configuration is shown in Figure 6.2. Figure 6.3. is the captured picture of the video operating in real-time with an Autonomous vehicle. The fog percentage value is derived from grasping the weather in the current state numerically. This part is the content of Chapter 2. The value for the weather probability is then used to determine whether the De-noising module operates. The vehicle test was conducted for the above serial situation, and the result was classified as shown in Fig. 6.3.

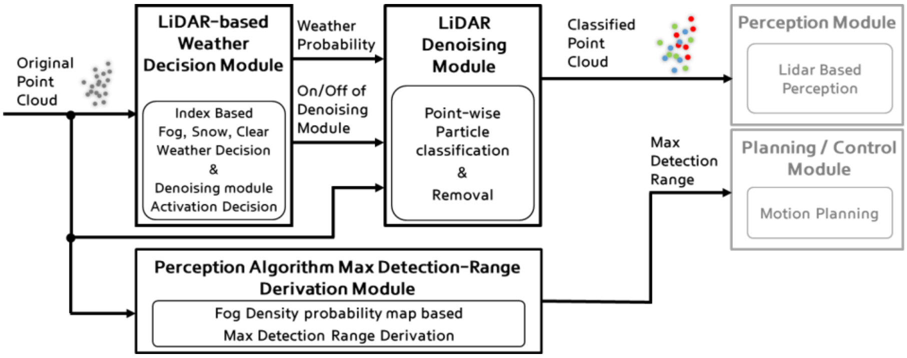

Original Point Cloud
LiDAR-based Weather Decision Module
Index Based Fog, Snow, Clear Weather Decision & Denoising module Activation Decision
Weather Probability
On/Off of Denoising Module
LiDAR Denoising Module
Point-wise Particle classification & Removal
Classified Point Cloud
Perception Module
Lidar Based Perception
Max Detection Range
Planning / Control Module
Motion Planning
Perception Algorithm Max Detection-Range Derivation Module
Fog Density probability map based Max Detection Range Derivation

Figure 6.1 Serial operating module in-vehicle

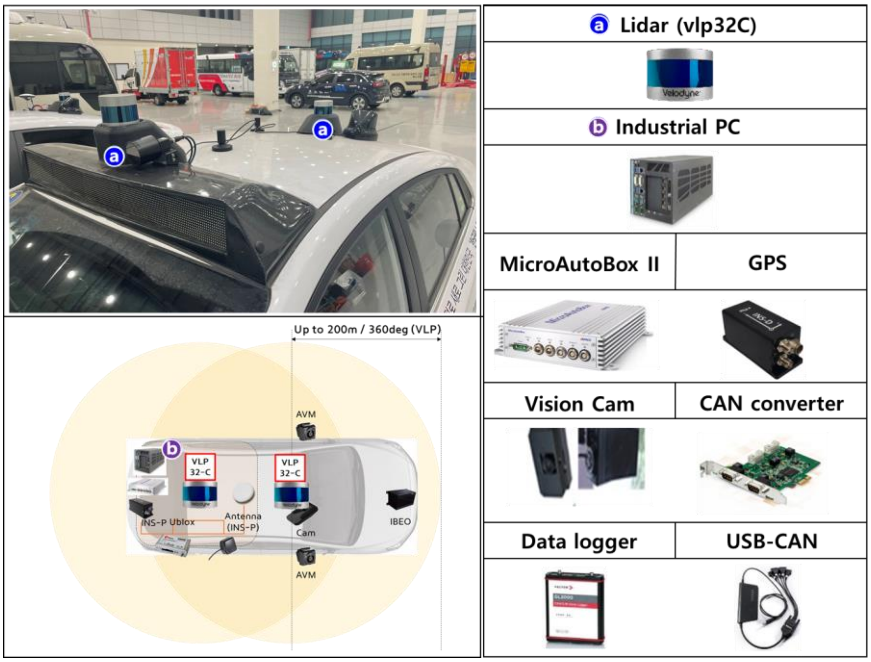
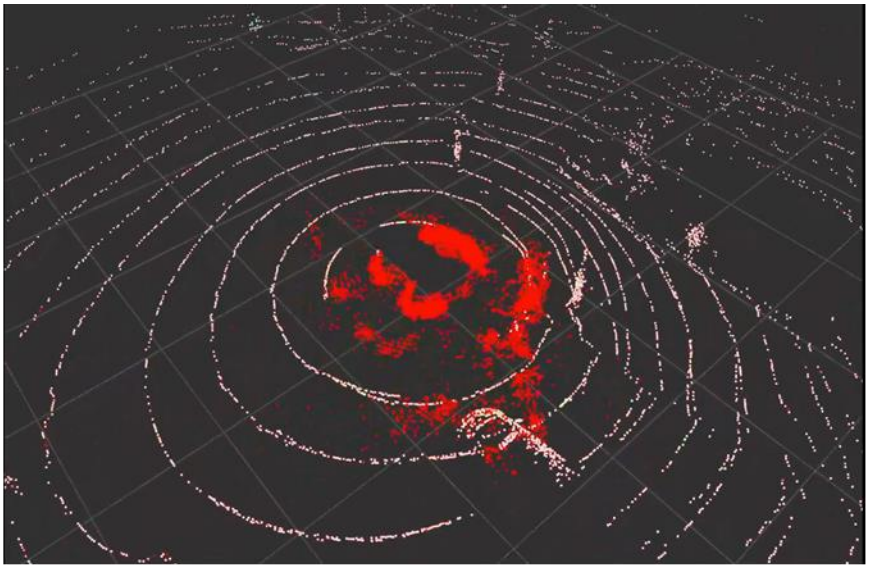


Figure 6.2 Test AV Hardware configuration



```
'De-noising Activate ==1fog Percentage='0.922134
```

Figure 6.3 Real-time operating video (captured figure)

In future work, we plan to develop a network that creates realistic data for fog situations by converting clear data into fog data through a fog generator. The fog model used now represented the fog best, but the actual fog data had different shapes. Therefore, the network that produces the most similar real fog will be carried out using the GAN method.

As the perception range decreases, the planning and control algorithms should consider how to control the vehicle safely. For example, in the lane change part, with the decreased detectible range, the algorithm should decide whether it is possible to make a lane change or not—Furthermore, How to design the desired velocity in a lane change scenario with heavy fog, et al. These problems need to be considered a lot in future.

Bibliography

Lang, Alex H. et al. "PointPillars: Fast Encoders for Object Detection From Point Clouds." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 12689-12697.

J.Lee (a). "[IEEE VTM, Supplementary Material] Fog scenes, Pointcloud shapes," YouTube, Feb. 2, 2022 [Video file]. Available: https://youtu.be/nvbroibrSCo. [Accessed: Feb. 2, 2022].

J.Lee (b). "[IEEE VTM, Supplementary Material] Occupancy Grid Map videos," YouTube, Feb. 2, 2022 [Video file]. Available: https://youtu.be/l_NPcWdB8eY. [Accessed: Feb. 2, 2022].

M. BIJELIC, T. GRUBER, AND W. RITTER. A benchmark for lidar sensors in fog: Is detection breaking down? IEEE Intelligent Vehicles Symposium (IV), page 760–767, 2018.

N. CHARRON, S. PHILLIPS, AND S. L. WASLANDER. De-noising of lidar point clouds corrupted by snowfall. Conference on Computer and Robot Vision (CRV), page 254–261, 2018.

A. DJURICIC AND B. JUTZI. Supporting uavs in low visibility conditions by multiple-pulse laser scanning devices. The international archives of the photogrammetry, remote sensing and spatial information sciences, 1, 2013

R. HEINZLER, P. SCHINDLER, J. SEEKIRCHER, W. RITTER, AND W. STORK. Weather influence and classification with automotive lidar sensors. IEEE Intelligent Vehicles Symposium (IV), page 1527–1534, 2019.

M. KUTILA, P. PYYKONEN, H. HOLZH¨UTER, M. COLOMB, AND P. DUTHON. Automotive lidar performance verification in fog and rain.

International Conference on Intelligent Transportation Systems (ITSC), page 1695–1701, 2018.

T. G. PHILLIPS, N. GUENTHER, AND P. R. MCAREE. When the dust settles: The four behaviors of lidar in the presence of fine airborne particulates. Journal of field robotics, 34:985–1009, 2017.

J.Lee (c). "LiDAR point data in Heavy Fog, Fog points density changes because of LiDAR physical property," YouTube, Mar. 23, 2022 [Video file]. Available: https://youtu.be/PjrSqgMKM18. [Accessed: Mar. 23, 2022].

Velodyne Lidar Ultra Puck Manual, "VLP-32C User Manual 63-9325 Rev. B," Icave2.cse.buffalo.edu.Available:https://icave2.cse.buffalo.edu/resources/sensor-modeling/VLP32CManual.pdf. [Accessed: Feb. 12, 2018].

Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles J3016, SAE international standard, issued Jan, 2014, revised Jun ,2018

H. Cho, Y.-W. Seo, B. V. Kumar, and R. R. Rajkumar. A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In Robotics and Au-tomation (ICRA), 2014 IEEE International Conference on, pages 1836–1843. IEEE, 2014.

G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Seg-mentation and recognition using structure from motion point clouds. In Proceedings of the IEEE European Conference on Computer Vision, pages 44–57. Springer, 2008.

M. Colomb, J. Dufour, M. Hirech, P. Lacˆote, P. Morange, and J.-J. Boreux. Innovative artificial fog production device-a technical facility for research activities. In Atmospheric Research, 2004.

D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel. Path planning for autonomous vehicles in unknown semi-structured environments. The International Journal of Robotics Research, 29(5):485–501, 2010. 3

P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedes-trian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(4):743–761, 2012. 2

M. Al Naboulsi, "Fog attenuation prediction for optical and infrared waves," Optical Engineering, vol. 43, no. 2, p. 319, 2004.

A. Filgueira, H. Gonz´alez-Jorge, S. Lag¨uela, L. D´ıaz-Vilari˜no, and P. Arias, "Quantifying the influence of rain in LiDAR performance," Measurement, vol. 95, pp. 143–148, 2017.

S. Hasirlioglu, I. Doric, A. Kamann, and A. Riener, "Reproducible Fog Simulation for Testing Automotive Surround Sensors," in 2017 IEEE 85th Vehicular Technology Conference (VTC Spring), 2017, pp. 1–7.

J. Ryde and N. Hillier, "Performance of laser and radar ranging devices in adverse environmental conditions," Journal of Field Robotics, vol. 26, no. 9, pp. 712–727, 2009.

T. G. Phillips, N. Guenther, and P. R. McAree, "When the dust settles: The four behaviors of LIDAR in the presence of fine airborne particulates," Journal of field robotics, vol. 34, no. 5, pp. 985–1009, 2017.

D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In Advances in neural information processing systems, pages 2366–2374, 2014.

Keisuke Yoneda, Naoki Suganuma, Ryo Yanase, Mohammad Aldibaja, Automated driving recognition technologies for adverse weather conditions, IATSS Research, Volume 43, Issue 4, 2019, Pages 253-262, ISSN 0386-1112, https://doi.org/10.1016/j.iatssr.2019.11.005.

B. Graham, M. Engelcke, and L. van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. CVPR, 2018

Chen, Y., Liu, S., Shen, X., & Jia, J. (2019). Fast point r-cnn. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 9775-9784).

C. Connolly, “Cumulative generation of octree models from range data,” in Robotics and Automation. Proceedings. 1984 IEEE International Conference on, vol. 1, mar 1984, pp. 25 – 32.

R. Martin, I. Stroud, and A. Marshall, “Data reduction for reverse engineering,” RECCAD, Deliverable Document 1 COPERNICUS project, No 1068, p. 111, 1997.

T. Weyrich, M. Pauly, S. Heinzle, R. Keiser, S. Scandella, and M. Gross, “Post-processing of Scanned 3D Surface Data,” Symposium on Point-Based Graphics, pp. 85–94, 2004.

C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271), pp. 839–846, 1998.

S. B. Gokturk, H. Yalcin, and C. Bamji, “A time-of-flight depth sensor System description, issues and solutions,” IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, vol. 2004-January, no. January, 2004.

S. Ronnback and A. Wernersson, "On filtering of laser range data in snowfall," 2008 4th International IEEE Conference Intelligent Systems, 2008, pp. 17-33-17-39, doi: 10.1109/IS.2008.4670551.

Y. Li, J. Li, L. Wang, J. Zhang, D. Li, and M. Zhang, “A weighted least squares algorithm for time-of-flight depth image denoising,” Optik International Journal for Light and Electron Optics, vol. 125, no. 13, pp. 3283–3286, 2014.

R. Chaudhary and H. Dasgupta, “An Approach for Noise Removal on Depth Images,” no. 1, pp. 1–2, 2016.

A. V. Le, S. W. Jung, and C. S. Won, “Directional joint bilateral filter for depth images,” Sensors (Switzerland), vol. 14, no. 7, pp. 11362–11378, 2014.

B. Huhle, T. Schairer, P. Jenke, and W. Straßer, “Fusion of range and color images for denoising and resolution enhancement with a non-local filter,” Computer Vision and Image Understanding, vol. 114, no. 12, pp. 1336–1345, 2010.

B. Huhle, T. Schairer, P. Jenke, and W. Straßer, “Robust non-local denoising of colored depth data,” 2008 IEEE Computer Society Con-ference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops, pp. 0–6, 2008.

L. Chen, H. Lin, and S. Li, “Depth image enhancement for kinect using region growing and bilateral filter,” Pattern Recognition (ICPR), 2012 21st . . . , no. Icpr, pp. 3070–3073, 2012.

CARBALLO, Alexander, et al. LIBRE: The multiple 3d lidar dataset. In: 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2020. p. 1094-1101.

O. Schall, A. Belyaev, and H.-P. Seidel, “Robust filtering of noisy scat-tered point data,” in Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings, pp. 71–144, IEEE, 2005.

P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer, “Bayesian point cloud reconstruction,” in Computer Graphics Forum, vol. 25, pp. 379–388, Wiley Online Library, 2006.

O. Schall, A. Belyaev, and H.-P. Seidel, “Adaptive feature-preserving non-local denoising of static and time-varying range data,” Computer-Aided Design, vol. 40, no. 6, pp. 701–707, 2008.

R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 1-4, doi: 10.1109/ICRA.2011.5980567.

W. Hu, X. Li, G. Cheung, and O. Au, “Depth map denoising using graph-based transform and group sparsity,” 2013 IEEE International Workshop on Multimedia Signal Processing, MMSP 2013, pp. 1–6, 2013.

Velodyne, HDL-32E User’s Manual and Programing Guide, 2016. Rev. K.

R. Heinzler, F. Piewak, P. Schindler and W. Stork, "CNN-Based Lidar Point Cloud De-Noising in Adverse Weather," in IEEE Robotics and

Automation Letters, vol. 5, no. 2, pp. 2514-2521, April 2020, doi: 10.1109/LRA.2020.2972865.

M. Bijelic, et al., "Seeing Through Fog Without Seeing Fog: Deep Multimodal Sensor Fusion in Unseen Adverse Weather," in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 2020 pp. 11679-11689

J.Lee (d). “Heavy Fog Video,” YouTube, Mar. 31, 2022 [Video file]. Available: https://youtu.be/I7WKqLIBr5M. [Accessed: Mar. 31, 2022].

B. Graham, M. Engelcke, and L. van der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. CVPR, 2018

Chen, Y., Liu, S., Shen, X., & Jia, J. (2019). Fast Point R-CNN. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 9774-9783.

Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. CVPR, 2020.

Guo, Yulan, et al. "Deep learning for 3d point clouds: A survey." IEEE transactions on pattern analysis and machine intelligence 43.12 (2020): 4338-4364.

F. J. Lawin, M. Danelljan, P. Tosteberg, G. Bhat, F. S. Khan, and M. Felsberg, “Deep projective 3D semantic segmentation,” in CAIP, 2017.

A. Boulch, B. Le Saux, and N. Audebert, “Unstructured point cloud semantic labeling using deep segmentation networks.” in 3DOR, 2017.

M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou, “Tangent convolutions for dense prediction in 3D,” in CVPR, 2018.

B. Wu, A. Wan, X. Yue, and K. Keutzer, “SqueezeSeg: Convo- lutional neural nets with recurrent crf for real-time road-object segmentation from 3D lidar point cloud,” in ICRA, 2018.

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 ICRA, 2019.

F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 MB model size,” in ICLR, 2016.

A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, “RangeNet++: Fast and accurate lidar semantic segmentation,” in IROS, 2019.

J. Huang and S. You, “Point cloud labeling using 3D convolu- tional neural network,” in ICPR, 2016.

L. Tchapmi, C. Choy, I. Armeni, J. Gwak, and S. Savarese, “SEG- Cloud: Semantic segmentation of 3D point clouds,” in 3DV, 2017.

J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional net- works for semantic segmentation,” in CVPR, 2015

H.-Y. Meng, L. Gao, Y.-K. Lai, and D. Manocha, “VV-Net: Voxel vae net with group convolutions for point cloud segmentation,” in ICCV, 2019.

D. Rethage, J. Wald, J. Sturm, N. Navab, and F. Tombari, “Fully- convolutional point networks for large-scale point clouds,” in ECCV, 2018.

A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner, “ScanComplete: Large-scale scene completion and semantic segmentation for 3D scans,” in CVPR, 2018.

C. Choy, J. Gwak, and S. Savarese, “4D spatio-temporal convnets: Minkowski convolutional neural networks,” in CVPR, 2019.

H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, “SplatNet: Sparse lattice networks for point cloud processing,” in CVPR, 2018.

R. A. Rosu, P. Schutt, J. Quenzel, and S. Behnke, “LatticeNet: Fast ¨ point cloud segmentation using permutohedral lattices,” arXiv preprint arXiv:1912.05905, 2019.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in CVPR, 2017.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “PointNet++: Deep hierarchical feature learning on point sets in a metric space,” in NeurIPS, 2017.

M. Jiang, Y. Wu, and C. Lu, “PointSIFT: A sift-like network mod- ule for 3D point cloud semantic segmentation,” arXiv preprint arXiv:1807.00652, 2018.

F. Engelmann, T. Kontogianni, J. Schult, and B. Leibe, “Know what your neighbors do: 3D semantic segmentation of point clouds,” in ECCVW, 2018.

H. Zhao, L. Jiang, C.-W. Fu, and J. Jia, “PointWeb: Enhancing lo- cal neighborhood features for point cloud processing,” in CVPR, 2019.

Z. Zhang, B.-S. Hua, and S.-K. Yeung, “ShellNet: Efficient point cloud convolutional neural networks using concentric shells statistics,” in ICCV, 2019.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in NeurIPS, 2017.

J. Yang, Q. Zhang, B. Ni, L. Li, J. Liu, M. Zhou, and Q. Tian, “Modeling point clouds with self-attention and gumbel subset sampling,” in CVPR, 2019.

L.-Z. Chen, X.-Y. Li, D.-P. Fan, M.-M. Cheng, K. Wang, and S.- P. Lu, “LSANet: Feature learning on point sets by local spatial attention,” arXiv preprint arXiv:1905.05442, 2019.

C. Zhao, W. Zhou, L. Lu, and Q. Zhao, “Pooling scores of neighboring points for improved 3D point cloud segmentation,” in ICIP, 2019.

B.-S. Hua, M.-K. Tran, and S.-K. Yeung, "Pointwise convolutional neural networks," in CVPR, 2018.

H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "KPConv: Flexible and deformable convolution for point clouds," in ICCV, 2019.

X. Ye, J. Li, H. Huang, L. Du, and X. Zhang, "3D recurrent neural networks with context fusion for point cloud semantic segmentation," in ECCV, 2018

L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in CVPR, 2018.

F. Engelmann, T. Kontogianni, and B. Leibe, "Dilated point con- volutions: On the receptive field of point convolutions," in ICRA,2020

Z. Zhao, M. Liu, and K. Ramani, "DAR-Net: Dynamic aggrega- tion network for semantic scene segmentation," arXiv preprint arXiv:1907.12022, 2019.

F. Liu, S. Li, L. Zhang, C. Zhou, R. Ye, Y. Wang, and J. Lu, "3DCNN-DQN-RNN: A deep reinforcement learning framework for semantic parsing of large-scale 3D point clouds," in ICCV, 2017.

Z. Kang and N. Li, "PyramNet: Point cloud pyramid attention network and graph embedding module for classification and segmentation," in ICONIP, 2019.

Y. Ma, Y. Guo, H. Liu, Y. Lei, and G. Wen, "Global context reasoning for semantic segmentation of 3D point clouds," in WACV, 2020.

N. Audebert, B. Le Saux, and S. Lefevre, "Semantic segmentation ` of earth observation data using multimodal and multi-scale deep networks," in ACCV, 2016.

S. Wang, S. Suo, W.-C. Ma, A. Pokrovsky, and R. Urtasun, "Deep parametric continuous convolutional neural networks," in CVPR, 2018.

Q. Huang, W. Wang, and U. Neumann, “Recurrent slice networks for 3D segmentation of point clouds,” in CVPR, 2018.

L. Wang, Y. Huang, Y. Hou, S. Zhang, and J. Shan, “Graph attention convolution for point cloud semantic segmentation,” in CVPR, 2019.

B. Gao, Y. Pan, C. Li, S. Geng and H. Zhao, "Are We Hungry for 3D LiDAR Data for Semantic Segmentation? A Survey of Datasets and Methods," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2021.3076844.

Fabian Groh, Patrick Wieschollek, and Hendrik P. A. Lensch. Flex-convolution (million-scale point-cloud learning beyond grid-worlds). In ACCV, 2018.

Oren Dovrat, Itai Lang, and Shai Avidan. Learning to sam-ple. In CVPR, 2019.

Abubakar Abid, Muhammad Fatih Balin, and James Zou. Concrete autoencoders for differentiable feature selection and reconstruction. In ICML, 2019.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. PointCNN: Convolution on X-transformed points. In NeurIPS, 2018.

Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

Liu, Yongcheng, et al. "Relation-shape convolutional neural network for point cloud analysis." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.

Y. Li, P. Duthon, M. Colomb and J. Ibanez-Guzman, "What Happens for a ToF LiDAR in Fog?," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2020.2998077.

J.Lee (e). “Fog weather Object Detection Result with mis/unrecognitions (PointPillar Network based Detector),” YouTube, April. 14, 2022 [Video file]. Available: https://youtu.be/xQpZciL16SU. [Accessed: April. 14, 2022].

J.Lee (f). “Un/Mis Recognition cases occured in Fog Model (Detector is PointPillar),” YouTube, April. 15, 2022 [Video file]. Available: https://youtu.be/VZ0JCNM3ya0. [Accessed:April. 15, 2022].

T. Gruber, M. Bijelic, F. Heide, W. Ritter, and K. Dietmayer, “Pixel-accurate depth evaluation in realistic driving scenarios,” Jun. 2019.[Online]. Available: https://arxiv.org/abs/1906.08953 http://arxiv.org/abs/1906.08953

M. Colomb, K. Hirech, P. André, J. J. Boreux, P. Lacôte, and J. Dufour, “An innovative artificial fog production device improved in the Euro-pean project “FOG”,” Atmospheric Res., vol. 87, no. 3-4, pp. 242–251, 2008.

M. Kutila, P. Pyykonen, W. Ritter, O. Sawade, and B. Schaufele, “Automo-tive LiDAR sensor development scenarios for harsh weather conditions,” in Proc. IEEE 19th Int. Conf. Intell. Transp. Syst., 2016, pp. 265–270.

J. Shen and S. C. S. Cheung, “Layer depth denoising and completion for structured-light RGB-D cameras,” in Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., 2013, pp. 1187–1194.

F. Piewak et al., “Boosting LiDAR-based semantic labeling by cross-modal training data generation,” in Proc. Lecture Notes Comput. Sci. (including Subser. Lecture Notes Artif. Intell. Lect. Notes Bioinformatics), 2019, vol. 11134 LNCS, pp. 497–513.

S. Fan, Q. Dong, F. Zhu, Y. Lv, P. Ye and F. -Y. Wang, "SCF-Net: Learning Spatial Contextual Features for Large-Scale Point Cloud Segmentation," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 14499-14508, doi: 10.1109/CVPR46437.2021.01427.

D. P. Kingma, J. L. Ba, “ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION”, arXiv:1412.6980v9, Jan 2017

초 록

악천후상황 자율주행 인지 시스템을 위한 LiDAR 포인트별 객체 검출 기반 잡음 제거 및 유효인지거리 추정

최근에는 자율주행 차량 연구의 인지 부분에 관해 많은 연구가 진행되었다. LiDAR(Light Detection And Ranging) 센서는 자율주행 차량의 기본 센서로, 많은 선행 연구에서 사용되었다. LiDAR 센서는 카메라, 레이더, 초음파 센서들의 단점을 보완해 정밀한 거리 정보를 제공한다. 그리하여 자율주행 분야에서 많이 사용이 되어왔다. 그러나 비, 안개 또는 눈과 같은 악천후 날씨가 LiDAR 포인트 데이터에 많은 영향을 미친다는 것으로 나타났다. LiDAR 를 사용한 물체 인식 알고리즘에 관해서만 많은 연구가 수행되었지만, 악천후 조건에서 LiDAR 의 인지 성능 저하 혹은 악천후 상황에서의 잡음제거에 관한 연구는 여전히 부족한 것으로 간주된다. 악천후 조건에서, 비, 눈, 안개 방울과 같은 공기 중의 미립자는 LiDAR 센서가 방출하는 레이저를 반사하는 것으로 나타났으며, 이는 이러한 미립자를 물체로 잘못 해석하는 결과를 초래했다. 이러한 잡음이 있는 데이터가 인지 알고리즘에 주어지면 알고리즘의 성능 저하가 발생한다.

따라서 본 논문은 객체 검출 기반으로 포인트별 잡음 제거 알고리즘을 개발하는 데 초점을 맞췄다. 제안된 잡음제거 딥 뉴럴

네트워크에서는 수신 필드를 점진적으로 증가시키면서 정보 손실 및 오분류 오류를 최소화하기 위해 특징 집합이 수행됩니다. 잡음 제거 모듈 앞쪽에서 작동하는 모듈은 LiDAR 포인트 클라우드 데이터(PCD)를 사용하여 날씨 상황을 판단하고 잡음제거 모듈에 현재 날씨를 확률적 정보를 전송한다. 또한 잡음 제거 모듈 뒷단의 모듈은 안개로 분류된 점을 사용하여 현재 상태에서의 안개 밀도를 점유 격자 맵 방법으로 도출하여 최대 인지 거리를 도출한다. 그리고 최대 인지 거리를 판단/제어 알고리즘 모듈로 전송한다.

악천후 데이터 중 특히 심한 안개 중 주행한 LiDAR 의 포인트 클라우드 데이터(Point Cloud Data)는 취득하기가 매우 힘들며 구하기가 힘들다. 해당 연구를 위해 심한 안개 속 포인트 데이터를 취득하였고 연구에 활용하였다. 일부 연구팀은 CEREMA 의 기후 챔버 시스템으로부터 라이다 포인트를 얻었다. 그러나 차량이 정지한 상태에서 통제된 기상 환경에서의 데이터만 취득하였다. 따라서 본 연구에서는 제안된 모듈을 통제된 기상 환경에서 얻은 대규모 데이터 세트를 통해 실험하고 검증했을 뿐만 아니라 실제 직접 취득한 짙은 안개 속 주행 포인트 데이터로도 검증했다. 제안된 잡음 제거 및 감지 범위 도출 모듈은 악천후 영향을 받는 데이터를 성공적으로 잡음 분할 및 필터링 하였다. 제안된 잡음 제거 및 감지 범위 도출 모듈의 효과는 실제 주행 데이터를 통해 평가된다.

주요어: 자율 주행, 잡음 제거, 객체 검출, 악천후 상황, 센서 성능 저하, 안개 날씨, 포인트 클라우드 밀도