

Article

Not peer-reviewed version

Optimized Right-Turn Pedestrian Collision Avoidance System Using Intersection LiDAR

Soo-Yong Park and [and Seok-Cheol Kee](#) *

Posted Date: 2 September 2024

doi: 10.20944/preprints202409.0047.v1

Keywords: LiDAR; Virtual Simulator; Pedestrian Detection; Cooperative Perception



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Optimized Right-Turn Pedestrian Collision Avoidance System Using Intersection LiDAR

Soo-Yong Park ¹ and Seok-Cheol Kee ^{2,*}

- ¹ Department of Smart Car Engineering, Chungbuk National University, Cheongju-si 28644; adrenalin94@chungbuk.ac.kr
- ² Department of Intelligent Systems and Robotics, Chungbuk National University, Cheongju-si 28644, Republic of Korea
- * Correspondence: sckee@chungbuk.ac.kr

Abstract: The incidence of right-turning pedestrian accidents is increasing in South Korea. Most of the accidents occurred when the larger vehicle was turning right, and the main cause of the accidents was found to be the driver's limited field of vision. After these accidents, the government has implemented a series of institutional measures with the objective of preventing such accidents. However, despite the institutional arrangements in place, pedestrian accidents continue to occur. We focused on the many limitations that autonomous vehicles, like humans, can face in such situations. To address this issue, we propose a right-turn pedestrian collision avoidance system by installing a LiDAR sensor in the center of the intersection to facilitate pedestrian detection. Furthermore, the urban road environment is considered, as this provides the optimal conditions for the model to perform at its best. During this research, we collected data on right-turn accidents using the CARLA simulator and ROS interface and demonstrated the effectiveness of our approach in preventing such incidents. Our results suggest that the implementation of this method can effectively reduce the incidence of right-turn accidents in autonomous vehicles.

Keywords: LiDAR; virtual simulator; pedestrian detection; cooperative perception

1. Introduction

In today's road traffic, intersections and their associated traffic systems are very complex. As intersections are a complex environment with many factors, vehicles and pedestrians in the intersection need to be more careful about their safety. However, as is often reported in the media, right-turning pedestrian accidents are one of the most common types of accidents in Korea. In response, the Korea Road Traffic Authority (KRTA) investigated right-turn accidents at all intersections over the past three years, and the results are shown in Figure 1 below.

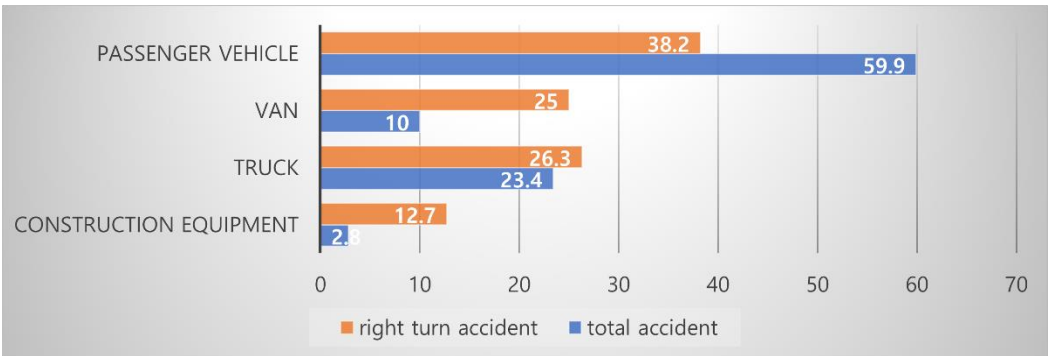


Figure 1. Comparing the right-turn accident rate with the overall accident rate 2018~2020.

In Figure 1, we can see the percentage of total accidents and right-turn accidents by vehicle type. As a result, we can note that the ratio of right-turn accidents to total pedestrian accidents is overwhelmingly higher for larger vehicles than for smaller passenger cars. Recognizing the seriousness of right-turn accidents, South Korea implemented the 'Right Turn Pause' law in January 2023. This law requires all vehicles to pause once when turning right to protect pedestrians. Since the law came into effect, all police forces in the country have been enforcing right-turn pauses, and local governments have been promoting the law, but even today, South Korea still suffers from right-turn accidents involving large vehicles, and the cause has been found to be the driver's limited field of vision.

On the other hand, today's autonomous vehicle technology has come a long way from the past [1–3]. It has been variously suggested that high-precision sensors and recognition systems can replace human object recognition intelligence [4–6], but it is still difficult to solve this problem technologically if the vehicle's field of view is insufficient. For example, Figure 2 below is one of the common road environments in Korea. A vehicle planning to make a right turn is very likely to have insufficient obstacle information recognition due to the presence of equipment on the curbstone or street trees.



Figure 2. Road equipment that obstructs visibility.

To solve this problem of insufficient perception information, many researchers have investigated various strategies. Cooperative perception is a method in which the sensor information of multiple agents is shared and complemented by each other to improve the cognitive performance of a single agent [7–9]. Agents can be not only vehicles, but also mobile robots and urban infrastructure systems, suggesting that sensors and systems designed for each environment can compensate for the limited field of view of a robot or vehicle. In particular, [10–12] propose various sensor fusion methods and 3D point cloud-based generation methods for collaborative perception among connected autonomous vehicles. These papers mainly propose solutions to the data occlusion phenomenon by fusing the features of point cloud data between different frames. However, these methods need further technical implementation and safety verification to be applied safely in real world, and it is not reliable that they can prevent right turn accidents when the vehicle is in a dynamic state. In [13,14], a cooperative recognition system is proposed using the infrastructure in urban areas. The implementation of the recognition system by attaching additional sensors to the infrastructure is one of the key components of the cooperative recognition system. This approach overcomes the limitations of existing recognition functions performed by individual vehicles or devices alone and enables more accurate and reliable recognition by cooperating with the infrastructure. However, most of the infrastructure facilities proposed in the literature are designed to target only a specific area, which limits their ability to cover the entire area within an intersection.

There are also studies related to right-turning pedestrian accidents in South Korea, such as [15]. In [15], the authors present criteria that can be legally reviewed to ensure that autonomous vehicles can safely make right turns. This study redesigned the right-turning process based on several case studies and presented an algorithm in the form of a flowchart. In [16], an algorithm is proposed to detect pedestrians by installing a LiDAR sensor in the middle of an intersection. This paper proposes a 128-channel LiDAR sensor for pedestrian detection, but 128-channel LiDAR is generally classified

as a high-cost sensor, which is very uneconomical in terms of equipment. Therefore, this paper proposes a more practical right turn prevention system by compensating for these shortcomings, and verifies it in a virtual environment.

The primary contributions of this paper are as follows:

- Optimization of a right-turning pedestrian detection system: The use of 64-channel LiDAR is sufficient to satisfy the requirements of the proposed system and demonstrates that the use of 64-channel LiDAR is reasonable in terms of system efficiency and cost. These results show that the system can maximize the efficiency of the sensor configuration while maintaining high performance recognition capability.
- Cost Effectiveness: The study shows that pedestrian detection systems in intersection can complement the perceptual capabilities of existing autonomous vehicles and reduce the cost of additional sensors on the vehicle.
- Right turn scenario configuration and dedicated dataset: As there are not enough datasets for right turn scenarios compared to the existing studies, we constructed a dedicated dataset based on various right turn scenarios in the simulator separately.
- Robustness in Various Weather Conditions: For pedestrian detection, the robustness of the pedestrian detection system was enhanced by using only LiDAR sensors that are less sensitive to weather changes, as opposed to other sensors that are sensitive to weather changes.

2. System Architecture

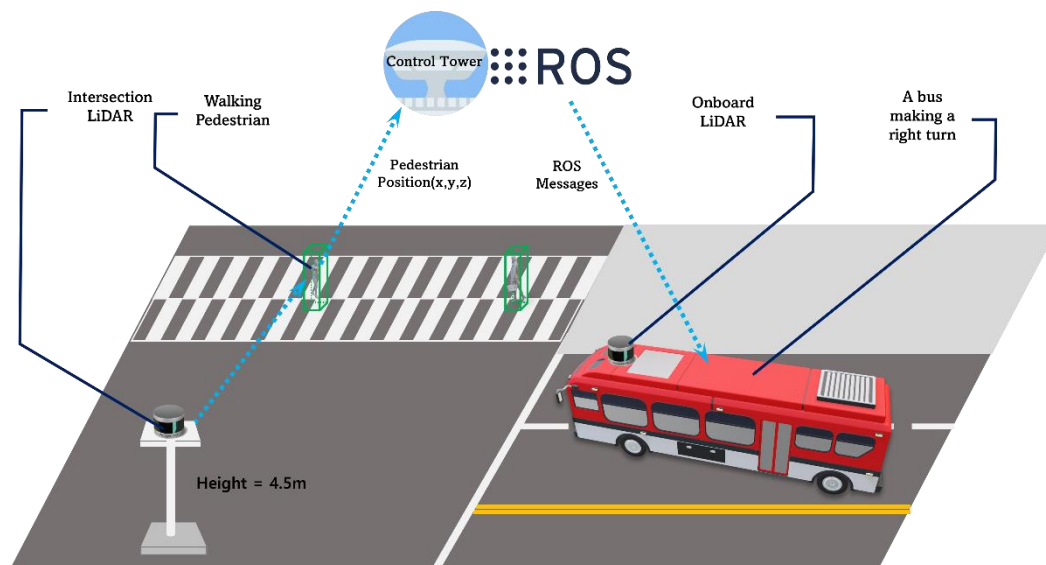


Figure 3. Proposed system in real world.

The overall system proposed in this study is shown in Figure 3. A LiDAR sensor is placed near the middle of the intersection to observe pedestrians on the crosswalk. A bus is chosen as the experimental vehicle for this purpose because, unlike a regular passenger car, a vehicle with a high height has a limited field of view. It is assumed that this vehicle is capable of autonomous driving, so we will install a LiDAR on top of the bus vehicle to obtain vehicle-centric perception data at the same time. In order to solve the problem presented in this study, it is necessary to recognize pedestrian information from sensors at the intersection and transmit pedestrian information to the bus vehicle before an accident occurs. Therefore, we propose that a control tower is required to apply this system in the real world. However, in this experiment, we replace the control tower with the ROS platform server in a virtual simulator environment.

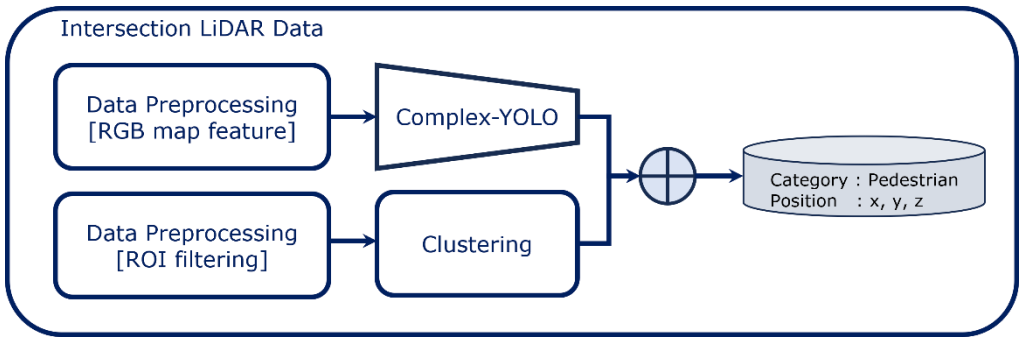


Figure 4. Intersection LiDAR process.

The pedestrian detection algorithm used in this paper is shown in Figure 4. First, we chose Complex-YOLO as the pedestrian detection deep learning model because unlike other deep learning algorithms, it is appropriate to use a one-stage detection model with fewer constraints on the inference speed [17]. Despite the primary classification of pedestrian classes from the deep learning model, further work needs to be done, and the reasons are as follows. Unlike vehicles and structures, pedestrians generate data from LiDAR that is regularly hard to identify (Figure 5), which can sometimes reduce the accuracy of Complex-YOLO. Therefore, filtering and clustering of regions of interest designated as dangerous areas are performed to determine the presence or absence of objects. Afterwards, the existing results are combined to increase the stability of pedestrian detection. To compare the performance of pedestrian detection using this approach, we used PV-RCNN as the pedestrian detection algorithm for onboard LiDAR [18]. It is important to note that PV-RCNN outperforms Complex-YOLO in terms of accuracy, although it is weaker in inference real-time, so it is used in this experiment in terms of accuracy comparison with Complex-YOLO.

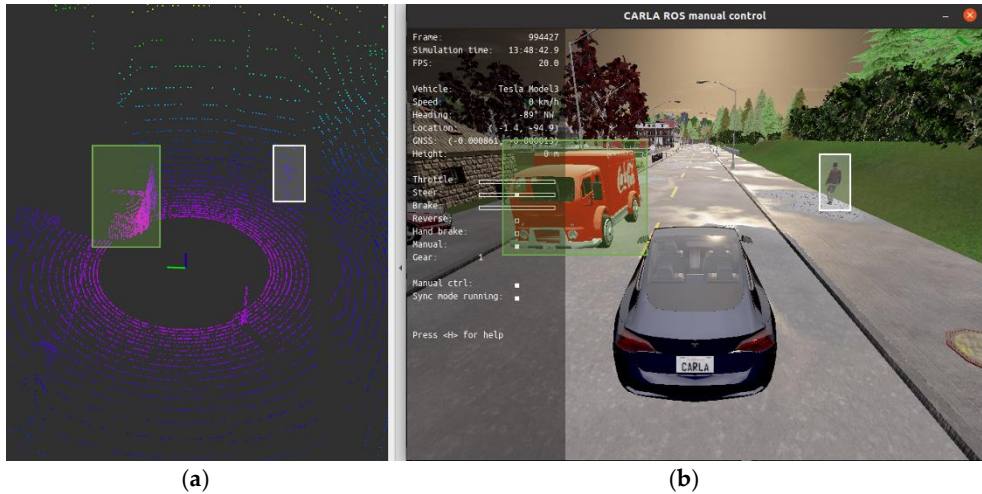


Figure 5. Comparison between vehicle(green box) and pedestrian data(white box). (a) Screenshot of LiDAR sensor frame data in ROS-rviz tool. (b) Screenshot of frame of CARLA simulator.

3. Materials

3.1. Simulator Selection

We used a simulation environment for the purpose of this study. Today, there is a wide variety of simulator software for autonomous driving and road traffic simulation. However, we chose the CARLA simulator for the following reasons [19,20].

- Easy to access
- The CARLA simulator is free and open software. Because it is open source, the simulator is continuously used by many users around the world and is well maintained for any functional errors

or bugs. We found the CARLA simulator to be a good choice for this study, given its cost and available features.

- Integration with the ROS platform

The CARLA simulator is a simulator that ensures compatibility with multiple robot platforms. Particularly, it is very convenient to integrate with the Robot Operation System (ROS), and many users have conducted research on integrating ROS and CARLA [21–23]. ROS is a robot operating system and an open-source framework for robot software development [24]. It has the advantage of offering various tools and libraries needed to develop robot applications, and it is very well connected with the CARLA simulator. In this study, we chose ROS because we needed to simultaneously create and control multiple agents on a map to collect data. At the same time, we investigated several simulators aimed at autonomous vehicle research, and based on [25], we compared the ROS platform with CARLA and other simulators and determined that the CARLA simulator was the most suitable simulator for this experiment. The Figure 6 below shows the actual interworking of ROS and CARLA simulator, and it can be seen that the data of the LiDAR installed on the vehicle is output normally by the ROS-rviz tool.

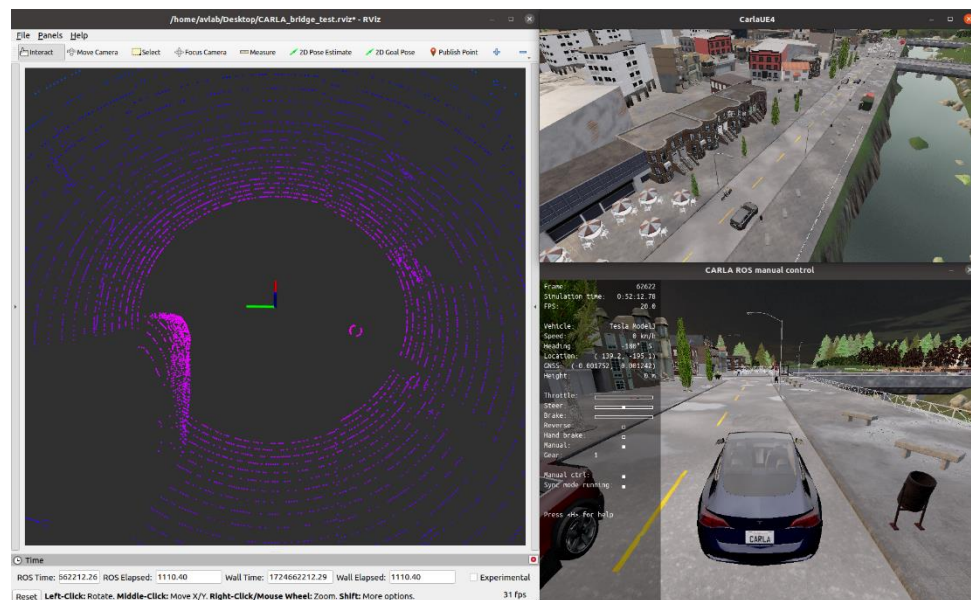


Figure 6. Screenshots of CARLA simulator with ROS platform.

3.2. Simulator Setting

The CARLA simulator supports agent assets that can be controlled by the user by default, and the properties contained in the assets are managed by a library of blueprints. In addition, all properties that can be controlled in maps also have blueprints, and information about the various features can be found in [26]. We used these to set up our experimental environment as follows.

① Experiment Configuration

The CARLA simulator is a program based on the Unreal graphic engine. To run this simulator, a high-end PC environment is usually required. Therefore, to reduce the load on the PC consumption, we configured the simulation environment to be divided into two PCs as shown in Figure 7 above.

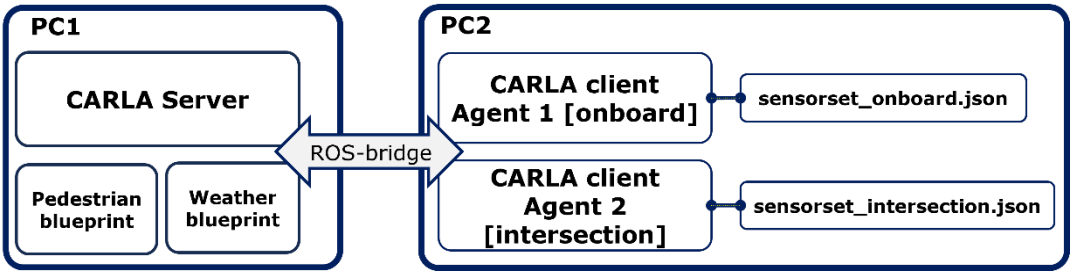


Figure 7. ROS and CARLA integration architecture.

The role of each PC is as follows First, we set up the CARLA simulator server to run on PC1, which is equipped with a high-end GPU. We also set up PC1 to create and control pedestrian blueprints and weather blueprints to design scenarios over the maps loaded on the server. On the other hand, we set up a network environment for the client to run on a separate PC2, and we associated an onboard agent with the sensorset json file on the target vehicle and an intersection agent with the sensorset json file in the center of the intersection, respectively (Tables 1 and 2).

In order to integrate CARLA with ROS, you must use a separately distributed package, and all data generated on CARLA can be accessed through the ROS interface.[27] We experimented by connecting each client to the CARLA server through the package, and when connecting to the server, you must specify the IP address of the host server and run the launch file.

Table 1. Attributes of onboard agent sensor set json file.

Sensor List	Type	ID
Camera	sensor.camera.rgb	rgb_view
LiDAR	sensor.LiDAR.ray_cast	LiDAR
GNSS	sensor.other.gnss	gnss
Objects	sensor.pseudo.objects	objects
Odom	sensor.pseudo.tf	tf
Tf	sensor.pseudo.odom	odometry
Speedometer	sensor.pseudo.speedometer	speedometer
Control	sensor.pseudo.control	control

Table 2. Attributes of intersection agent sensor set json file.

Sensor List	Type	ID
Camera	sensor.camera.rgb	rgb_view
LiDAR	sensor.LiDAR.ray_cast	LiDAR
GNSS	sensor.other.gnss	gnss
Objects	sensor.pseudo.objects	objects
Tf	sensor.pseudo.odom	odometry

② Map

For this study, we chose the largest intersection in Town 10 as the experimental area to simulate accidents that occur within the urban area (Figure 8).

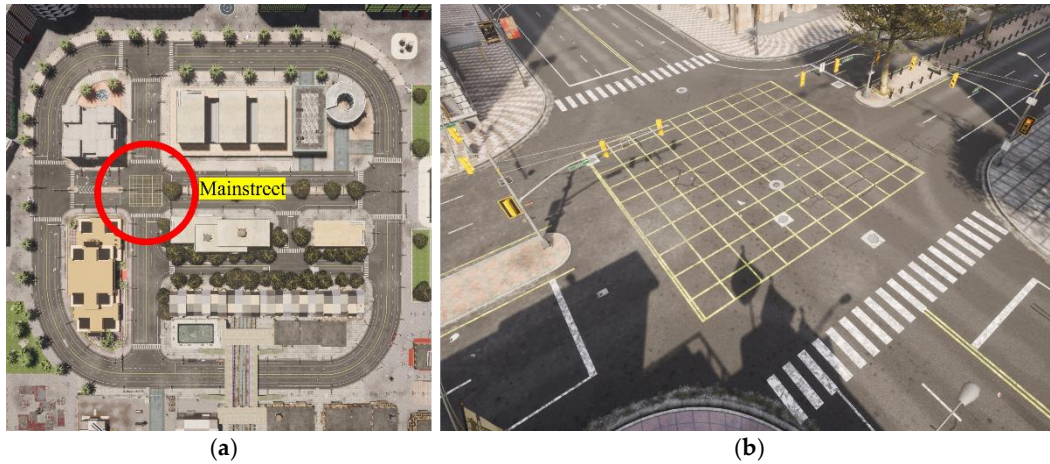


Figure 8. Used map in our experiment. (a) Location of intersection named Mainstreet in town 10. (b) Screenshot of Mainstreet.

③ Onboard Sensor

For this study, we utilized a Mitsubishi-Fusorosa model bus vehicle asset (Figure 9) and specified a 32-channel LiDAR sensor to be attached to it. The data showed the following limitations of the onboard LiDAR sensor. First, the vehicle's height is higher than that of a normal vehicle, which inevitably leads to blind spots, as shown in Figure 10. Second, the 32-channel LiDAR has a smaller resolution compared to other high-channel LiDARs, so the data sparsity is strongly affected by distance (Figure 11). As a result, we can see that the same problems with LiDAR that we encounter in the real world can also be applied in the simulator



Figure 9. Onboard agent named Mitsubishi-Fusorosa.

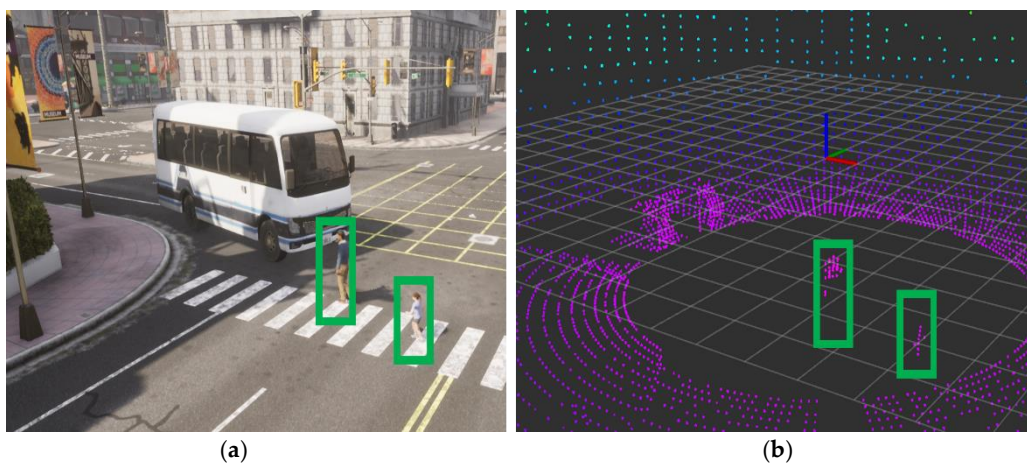


Figure 10. Example of blindspot situation. (a) Simulator screen (b) ROS-rviz screen.

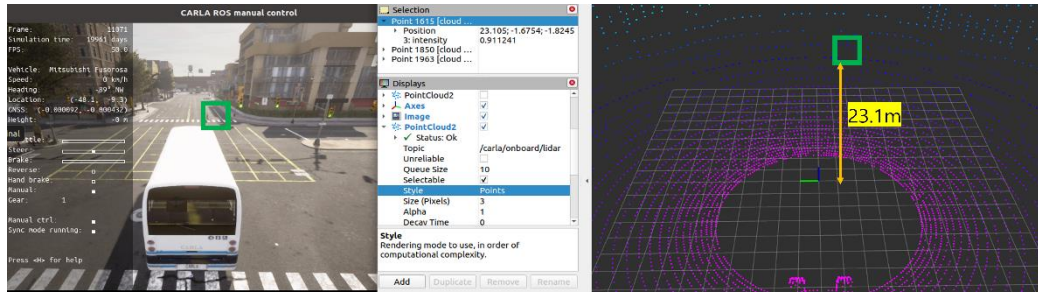


Figure 11. An example of LiDAR data sparsity, where the data points are significantly sparse at a distance of 23.1 meters from the person's location (green box).

④ Intersection Sensor

The perception sensors that can be utilized within the CARLA simulator are camera, LiDAR, and Radar sensors, and the output of each sensor can be seen as ROS message data. We installed a 64-channel LiDAR sensor at the intersection as a sensor for pedestrian detection and did not consider installing other sensors other than LiDAR for the following reasons. First, CARLA can handle extreme weather conditions, such as heavy fog or rain. For example, as shown in Figure 12a, we can see that the camera data does not produce a good image of the pedestrian in the fog, even though there are two pedestrians on the road, and the radar data also cannot output pedestrian data in the fog (Figure 12b). However, even in the fog, the LiDAR sensor can capture the person on the crosswalk unlike other sensors, and the ROS message data is obtained properly (Figure 12c). Second, unlike cameras and radar sensors, LiDAR sensors have a 360-degree field of view. The unique advantage of LiDAR sensors is that they can cover the entire area, not just one area within the range. Therefore, in addition to the proposed experimental area in this paper, the rest of the Mainstreet and the pedestrian crossing area can be observed simultaneously, which suggests that the LiDAR sensor is a very universal sensor compared to other sensors. Therefore, in this paper, LiDAR is selected as the main sensor for pedestrian identification for these reasons.

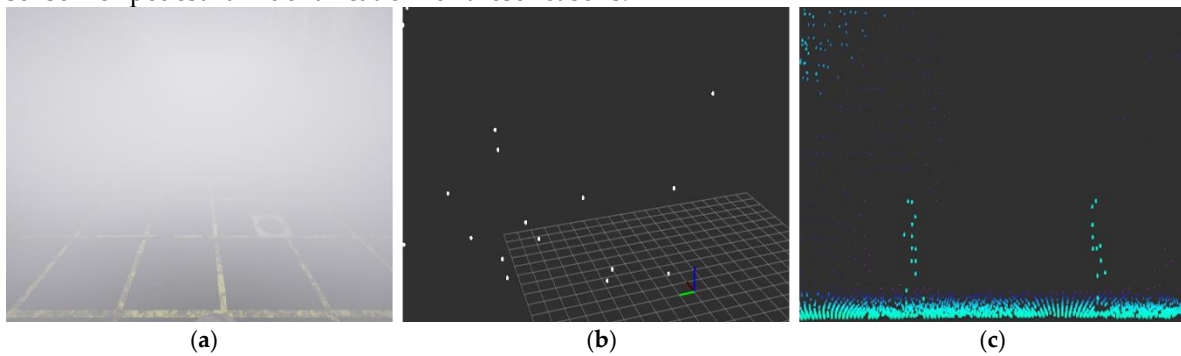


Figure 12. Sensor data display in dense fog condition. (a) Camera. (b) Radar. (c) LiDAR.

However, instead of the approach proposed in [16], we used a 64-channel LiDAR, which is lower than 128 channels. Table 3 below compares the specifications of 64-channel LiDAR and 128-channel LiDAR, and Figure 13 captures one frame of 128-channel LiDAR and one frame of 64-channel LiDAR. From the screen, we can see that the shape of the pedestrian on the crosswalk is displayed correctly.

Table 3. Comparison between 64 channel LiDAR and 128 channel LiDAR.

LiDAR Channel	Range(m)	Points per second	Upper FoV(deg)	Lower FoV(deg)	Rotation frequency(hz)
64	80	2,621,440	22.5	-22.5	100
128	80	5,242,880	22.5	-22.5	100

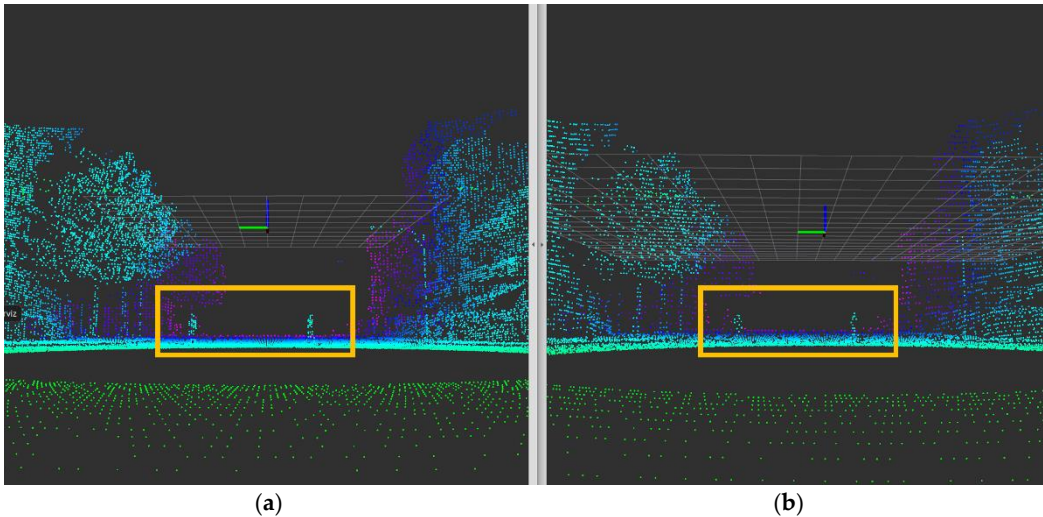


Figure 13. Screenshots of pedestrian pointcloud in ROS-rviz tool. (a) 128 channel. (b) 64 channel.

⑤ Weather Condition

CARLA also supports blueprints for weather, so you can create any weather situation whatever you want except for snow conditions. Table 4 below shows representative parameters for each weather, and we created sunny_day, clear_night, fog_day, fog_night, rain_day, and rain_night, respectively, and used them in our experiments, and the results can be seen in Figure 14 below.

Table 4. Parameters for adjusting weather conditions.

	Cloudiness	Sun altitude angle	Precipitation	Fog density	Fog fallout	Precipitation deposits
Sunny day	0.0	70.0	0.0	-	-	-
Clear night	0.0	-70.0	0.0	-	-	-
Fog day	60.0	70.0	0.0	35.0	7.0	-
Fog night	60.0	-70.0	0.0	25.0	7.0	-
Rain day	85.0	70.0	80.0	10.0	-	100.0
Rain night	85.0	-70.0	80.0	10.0	-	100.0

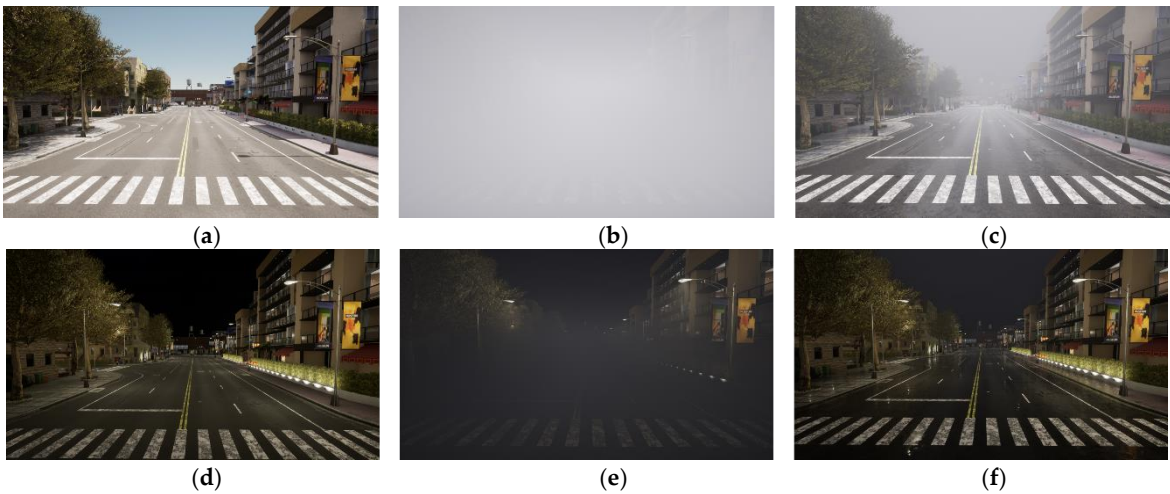


Figure 14. Screenshots of a various of weather conditions. (a)Sunny day (b)Fog day (c)Rain day (d)Clear night (e)Fog night (f)Rain night.

4. Experiment

4.1. Dataset

Since many existing datasets do not fit the purpose of this study, we designed a dedicated dataset for this experiment and prepared the dataset in the following order.

① Pedestrian Setting

There are three types of pedestrian blueprints in the CARLA simulator: normal adult, overweight adult, and children (Figure 15), and CARLA supports each type of pedestrian with a different blueprint ID number, so you can create or control a pedestrian of that type at any time. This can be seen in Table 5 below.



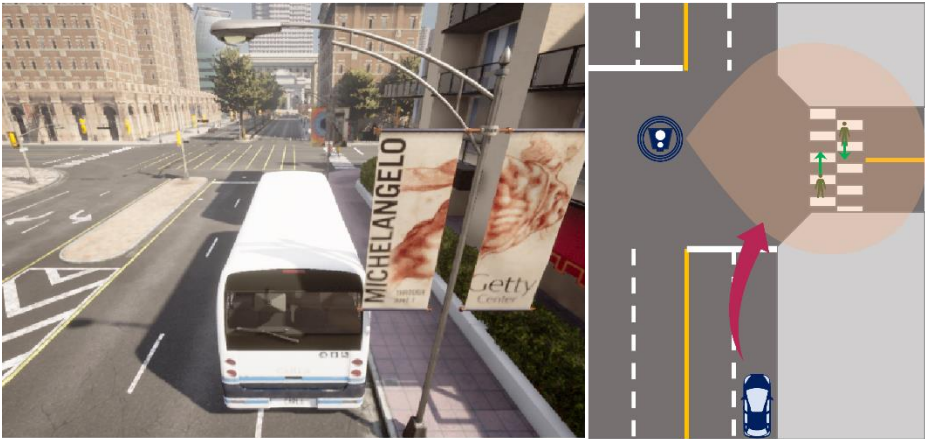
Figure 15. Three types of pedestrians in CARLA simulator. (a) Normal adult. (b) Overweight adult. (c) children.

Table 5. Blueprint ID lists for different pedestrian types.

Type	blueprint ID
Normal Adult	0001/0005/0006/0007/0008/0004/0003/0002/0015
	0019/0016/0017/0026/0018/0021/0020/0023/0022
	0024/0025/0027/0029/0028/0041/0040/0033/0031
Overweight Adult	0034/0038/0035/0036/0037/0039/0042/0043/0044
	0047/0046
Children	0009/0010/0011/0012/0013/0014/0048/0049

② Scenario Configuration

In this study, a right-turning accident scenario was designed to collect data. The target area is the Mainstreet area within town10, and various situations were created using pedestrian type, pedestrian speed, and number of pedestrians as detailed variables. The speed of the target vehicle was manually controlled at 15-25km/h as shown in Figure 16 below, and 18 scenarios were designed by applying various weather conditions during driving (Table 6). Through this scenario design, we were able to generate a dataset that more closely resembles and accurately reflects real-world right-turn accident situations.



(a) (b)

Figure 16. Screenshot of a part of scenario. (a) Control bus by manual (b) Scenario plan diagram.**Table 6.** Scenario cases for dataset.

	Weather	Type of pedestrians	Direction of pedestrian	Speed of pedestrian	Vehicle speed
#1	Sunny day	Adult : 2 (normal)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#2	Sunny day	Adult : 2 (normal, overweight)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#3	Sunny day	Adult : 1, Children :1	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#4	Clear night	Adult : 2 (normal)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#5	Clear night	Adult : 2 (normal, overweight)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#6	Clear night	Adult : 1, Children :1	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#7	Fog day	Adult : 2 (normal)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#8	Fog day	Adult : 2 (normal, overweight)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#9	Fog day	Adult : 1, Children :1	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#10	Fog night	Adult : 2 (normal)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#11	Fog night	Adult : 2 (normal, overweight)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#12	Fog night	Adult : 1, Children :1	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#13	Rain day	Adult : 2 (normal)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#14	Rain day	Adult : 2 (normal, overweight)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#15	Rain day	Adult : 1, Children :1	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#16	Rain night	Adult : 2 (normal)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#17	Rain night	Adult : 2 (normal, overweight)	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)
#18	Rain night	Adult : 1, Children :1	Ped1: 0°/Ped2: 180°	Ped1: 1.0(m/s)/Ped2: 1.5(m/s)	15-25(km/h)

③ Data Time Synchronization

For our proposed method to prove correctly, the onboard agent and the intersection agent must be simulated at the same time and within the same period, which can be adjusted through a parameter named passive in CARLA. This parameter is specified as a bool type, and we were able to synchronize the time of the ROS Topic data through this parameter. In Figure 17 below, you can see how the time synchronization is achieved through the True/False of the passive parameter, and in this way we collected about 40 Gb of ROS bag files.

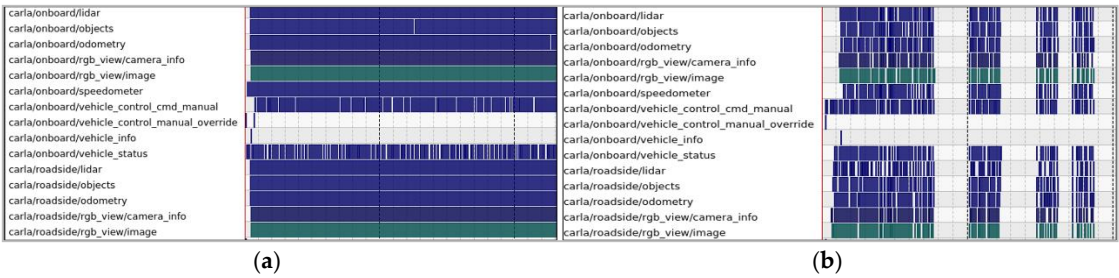


Figure 17. Comparison of synchronization status. (a) Synchronous ros messages (b) Asynchronous ros messages.

④ Data Extraction

To compare the pedestrian detection performance between the two agents, we extracted LiDAR bin files for onboard and intersection from the ROS bag files and label txt files containing pedestrian information. The ROS-ApproximateTimeSynchronizer package was utilized for extraction, and the label files stored label data in the order of x, y, z, dx, dy, dz, heading, and class type for 3D LiDAR detection training. As a result, the number of extracted data for each scenario is shown in Table 7 below.

Table 7. Number of datasets.

	Sunny day	Clear night	Fog day	Fog night	Rain day	Rain night	Total
Onboard	5,299	4,066	5,143	4,876	5,420	5,555	30,359
Intersection	5,299	4,066	5,143	4,876	5,420	5,555	30,359
							60,718

4.2. Training Details

We trained the PV-RCNN algorithm on the onboard LiDAR data and the Complex-YOLO algorithm on the intersection LiDAR data from the extracted data. The GPU specification of the PC used as a training environment is NVIDIA GEFORCE-RTX-3090ti, and the parameter properties of each algorithm set for training are shown in Table 8 below. We utilized 70% of the total dataset as training data, and the training data was randomly mixed with scenario data of pedestrian type and weather type.

Table 8. Training parameters.

	Class list	Anchors	Filtering[m]	Feature type	Feature size
Complex-YOLO [Intersection]	Pedestrian	[1.08, 1.19]	Min/Max x : 0, 40	BEV	[512,1024,3]
			Min/Max y : 0, 40		
			Min/Max z : -4.5, 0		
PV-RCNN [Onboard]	Pedestrian	[0.96, 0.88, 2.13]	Min/Max x : ± 75.2	Voxel	[0.1,0.1,0.15]
			Min/Max y : ± 75.2		
			Min/Max z : ± 4.0		

4.3. Clustering Details

We used Complex-YOLO for pedestrian detection and clustering at the same time. Since our experiment targets pedestrians on crosswalks, we first performed filtering by specifying a region of interest. By filtering the region of interest, the obstacles on the crosswalk are more visible. Then we applied the DBSCAN clustering method to cluster the objects on the crosswalk. The DBSCAN method

is a density-based clustering algorithm that is robust to noise and performs well in clustering objects in the target area. You can see the sequential clustering process in Figure 18 below.

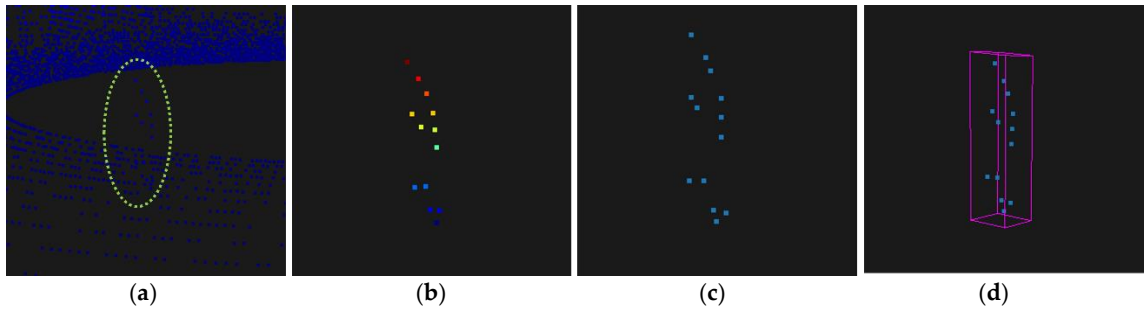


Figure 18. Clustering process. (a) Raw data.(pedestrian in green circle) (b) Filtered data by roi.(region of interest) (c) Clustered data. (d) Bounding box of object(magenta color).

5. Experimental Result

We evaluated the performance of the algorithm proposed in this paper with about 30% of the data from our own dataset. At this time, we prepared evaluation datasets for each type of pedestrian according to the designed scenarios (Table 9) so that they can be evaluated in various environments, and the number of datasets prepared for each scenario can be found in Table 10 below..

Table 9. Classification of pedestrian for evaluation.

Attributes of cases	
Case 1(#1)	Normal adult: 2
Case 2(#2)	Normal adult: 1 , Overweight adult: 1
Case 3(#3)	Normal adult: 1, Children: 1

Table 10. Evaluation dataset.

Weather Type	Num.	Total
Sunny Day	#1: 609 / #2: 499 / #3: 483	1,591
Clear Night	#1: 390 / #2: 389 / #3: 442	1,221
Fog Day	#1: 494 / #2: 519 / #3: 531	1,544
Fog Night	#1: 456 / #2: 437 / #3: 571	1,464
Rain Day	#1: 566 / #2: 555 / #3: 507	1,628
Rain Night	#1: 561 / #2: 585 / #3: 523	1,669
Total		9,117

5.1. Onboard Evaluation

5.1.1. Qualitative Evaluation

The PV-RCNN qualitative evaluation of Onboard’s data is shown below. As shown in Figure 19, it can be seen that pedestrians located within the blind spot of the bus are not detected properly. In addition, pedestrians located outside of a certain range are also poorly detected. From these results, we can conclude that large vehicles need to perform safe recognition functions for autonomous driving, and the larger the volume, the wider the blind spot tends to be. This shows not only the limitations of deep learning but also the need for additional sensors to compensate for blind spot, but we can conclude that installing additional sensors may not be economically reasonable.

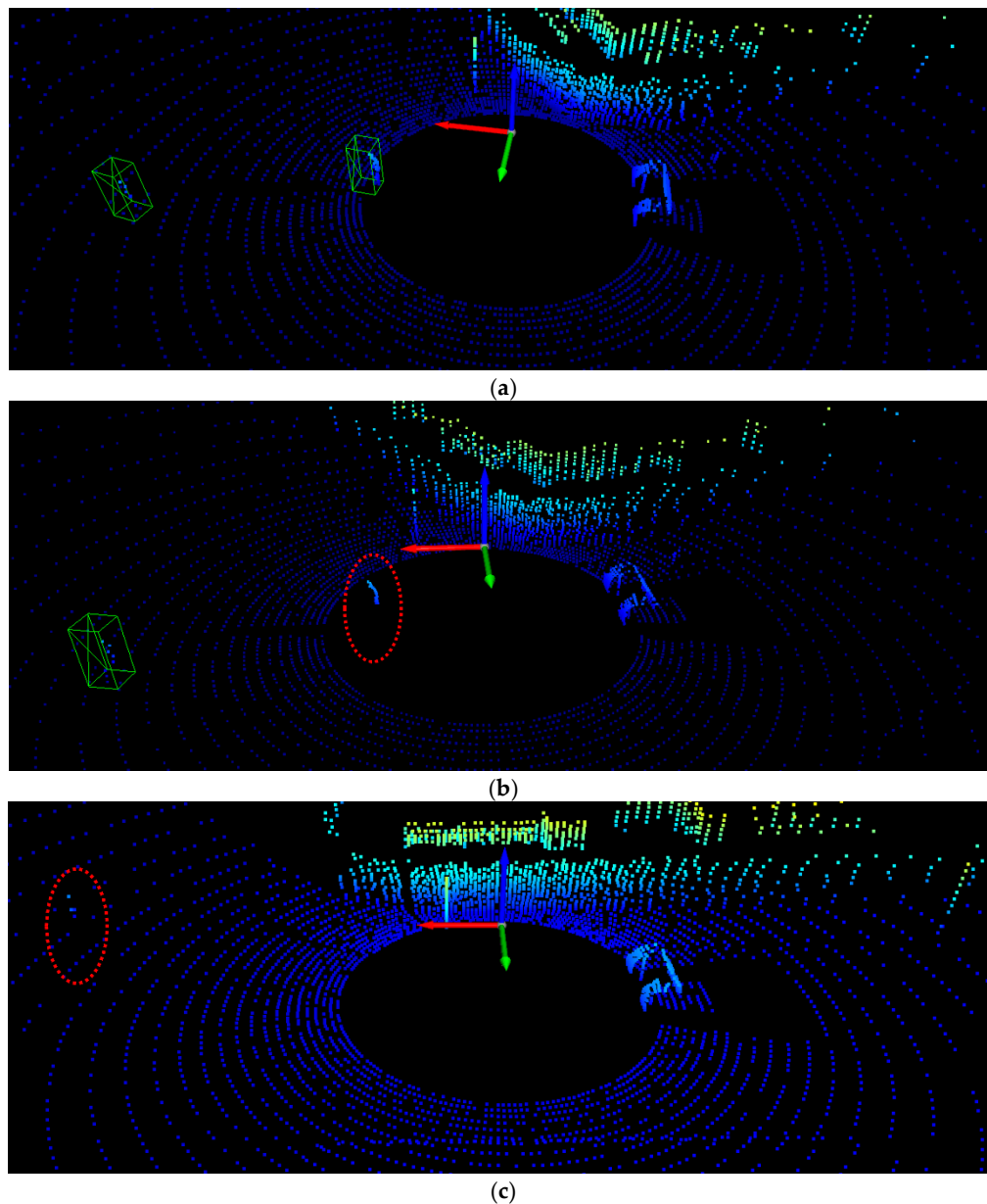


Figure 19. PV-RCNN result images. (a) True positive. (b) Undetected pedestrian due to blind spot. (c) Undetected pedestrian due to data sparsity by range limit.

5.1.2. Quantitative Evaluation

The PV-RCNN quantitative evaluation of Onboard’s data is shown below. As can be seen from Table 11, the results show low accuracy due to the failure to detect pedestrians within the blind spot and pedestrians beyond a certain distance. In particular, the number of LiDAR data that constituted pedestrians beyond a certain distance was too few to infer the model. (Figure 20) Therefore, it can be concluded that the 32-channel LiDAR in the simulator is not reliable for vehicle safety, and the lack of perception data in autonomous vehicles necessarily raises the need for external sensor involvement.

Table 11. Quantitative evaluation of onboard LiDAR process.

Sunny Day			Clear Night			Fog Day			Fog Night			Rain Day			Rain Night		
#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3

				0														
PV-RCNN	0.53	0.60	0.53	.	0.54	0.35	0.36	0.43	0.39	0.51	0.45	0.61	0.37	0.43	0.52	0.34		
				2														

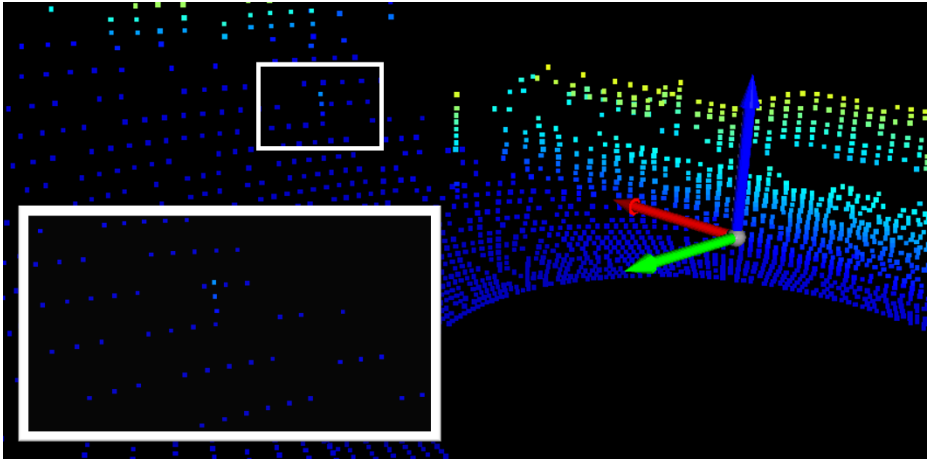


Figure 20. Pedestrian data captured by a 32 channel LiDAR(onboard).

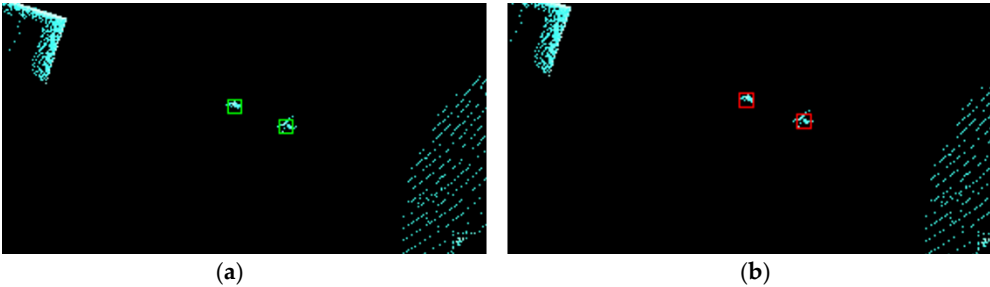
5.2. Intersection Evaluation

5.2.1. Qualitative Evaluation

The qualitative results of Complex-YOLO evaluated on our dataset were as follows. We found that sometimes two pedestrians were not detected even though they were on the crosswalk. The reason for this is that the number of LiDAR data corresponding to pedestrians is less than 128-channel LiDAR data, so BEV features are not fully reflected in the training. In particular, we found that child pedestrians, as shown in Figure 21, have significantly fewer points than adult pedestrians, so the general deep learning method sometimes fails to detect them. Figure 22 shows the examples of correctly detected and incorrect results when compared to the ground truth in each case (Table 9).



Figure 21. Pedestrian data captured by a 64 channel LiDAR (intersection).



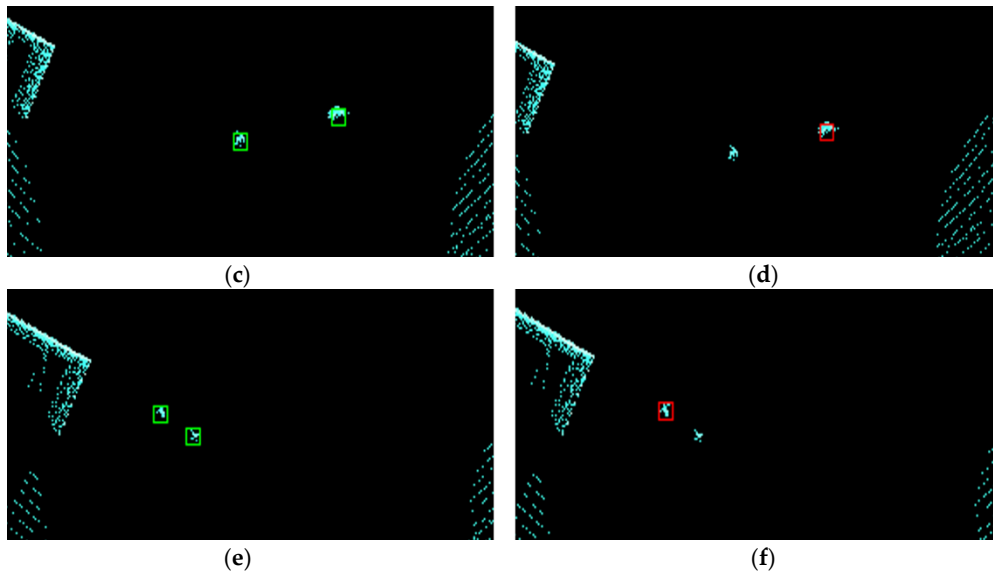


Figure 22. Complex-YOLO result images. (a) Ground Truth of case 1. (b) Complex-YOLO output of case 1. (c) Ground Truth of case 2. (d) Complex-YOLO output of case 2. (e) Ground Truth of case 3. (f) Complex-YOLO output of case 3.

Figure 23 below is an example of the clustering results. As you can see in the result figures, after performing the clustering operation, we reconstructed the boxes in a 2D Bird Eye View (BEV) view, and we can clearly see that even objects that were not detected by Complex-YOLO were correctly detected.

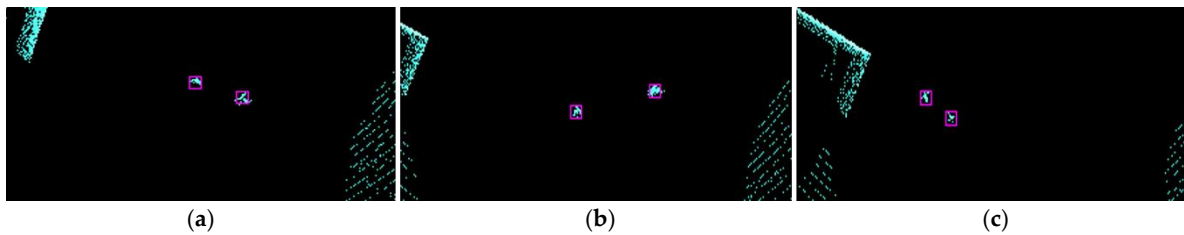


Figure 23. Clustering result images. (a) Clustering output of case 1. (b) Clustering output of case 2. (c) Clustering output of case 3.

Figure 24 below shows the final output of the fusion of Complex-YOLO and clustering results. The fusion of the bounding box from Complex-YOLO and the bounding box from clustering complements the results of Complex-YOLO. As a result, we can note that the current location of the pedestrian can be determined based on the Intersection LiDAR.

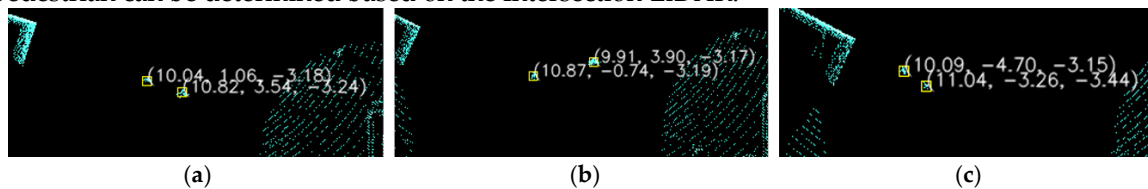


Figure 24. Output of bounding boxes' fusion. (a) Output of case 1. (b) Output of case 2. (c) Output of case 3.

5.2.2. Quantitative Evaluation

The following is a quantitative evaluation of the Intersection LiDAR process. Table 12 shows the quantitative results of Complex-YOLO and our proposed model for each scenario. We can see that most of the results of Complex-YOLO detect pedestrians at a certain accuracy level, but when we fused the clustering results, we can see that the accuracy results are mostly above 0.9. This shows that

our model is better at detecting pedestrians when fused with the clustering process, suggesting that LiDAR is better than other sensors at detecting pedestrians in a simulated environment. Furthermore, the performance is not significantly different from the previously proposed 128-channel LiDAR experiment, suggesting that the 64-channel LiDAR in our system is sufficient and can be replaced.(Table 13)

Table 12. Quantitative evaluation of intersection LiDAR process (64 channels).

	Sunny Day			Clear Night			Fog Day			Fog Night			Rain Day			Rain Night		
	#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3
Complex YOLO	0.81	0.68	0.90	0.88	0.86	0.86	0.85	0.92	0.90	0.80	0.83	0.70	0.66	0.79	0.92	0.72	0.71	0.81
Ours	0.98	0.91	0.88	0.98	0.95	0.94	0.93	0.98	0.97	0.97	0.98	0.94	0.93	0.94	0.96	0.97	0.95	0.91

Table 13. Quantitative evaluation of intersection LiDAR process (128 channels).

	Sunny Day			Clear Night			Fog Day			Fog Night			Rain Day			Rain Night		
	#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3	#1	#2	#3
Complex YOLO	0.73	0.85	0.91	0.90	0.78	0.83	0.78	0.82	0.80	0.88	0.79	0.66	0.66	0.54	0.66	0.66	0.86	0.56
Ours	0.96	0.98	0.98	0.93	0.85	0.94	0.98	0.99	0.88	0.98	0.98	0.75	0.88	0.97	0.88	0.88	0.95	0.98

6. Conclusion And Discussion

Through this experiment, we found that the main cause of right-turning pedestrian accidents is the limited field of view, which is a problem that can also occur in large scale autonomous vehicles. Therefore, we proposed a solution for cooperative recognition with vehicles through simulation of pedestrian detection experiments using LiDAR installed at intersections, and confirmed that pedestrian detection performance using 64-channel LiDAR is excellent. However, this experiment was only verified in a virtual simulation, and there are limitations to implementing it in the real world, such as the following.

- ① Sensitivity to weather: The LiDAR sensor used in this experiment is considered to be less sensitive to noise in different weather conditions because it produces idealized data in virtual environment. However, in the real world, LiDAR sensors are very sensitive to extreme external environments, making data collection difficult [28]. Therefore, in future research, we will install LiDAR in an urban environment in a real testbed (Figure 25) and collect and analyze LiDAR data directly as the weather changes.



Figure 25. C-track test bed. (a) Aerial filming of C-track test bed (b) Urban area in C-track test bed.

- ② Design of sensor-mounted equipment and wireless data transmission: For the system to work in the real world, a process is required to transmit pedestrian data from the sensors to the vehicle without loss. In this study, the data transmission was performed by replacing the ROS server,

but in the real world, we plan to design a separate communicable hardware to further design the data transmission and reception process and verify the safety. In addition, the role of sensor-mounted equipment is very important in this system, and many studies have been conducted in this regard [29–31]. Based on this, we plan to implement this system by designing a device that satisfies the Korean road traffic law.

Author Contributions: Conceptualization, S.-Y.P.; methodology, S.-Y.P.; software, S.-Y.P.; validation, S.-Y.P.; investigation, S.-Y.P.; data curation, S.-Y.P.; writing-original draft preparation, S.-Y.P.; writing-review and editing, S.-C.K.; supervision, S.-C.K.; funding acquisition, S.-C.K. All agreed to the published version of the manuscript.

Funding: This research was funded by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A5A8026986), The Korea Planning & Evaluation Institute of Industrial Technology(KEIT) grant funded by the Korea government, Ministry of Trade, Industry and Energy(MOTIE) (RS-2024-00408818)“

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the first author.

Acknowledgments: This study originates from an EVS37 oral presentation in Seoul, South Korea (26 April 2024). Special thanks to the EVS37 program for the opportunity to submit this research paper to this WEVJ Special Issue.

Conflicts of Interest: Soo-Yong Park is a master course student at Smart-Car Engineering department, Chungbuk National graduate. The remaining authors declare no conflict of interest.

References

1. Chen, L. et al. Milestones in Autonomous Driving and Intelligent Vehicles: Survey of Surveys. *IEEE Transactions on Intelligent Vehicles*, **2023**, vol. 8, no. 2, pp. 1046-1056
2. Hussain, R.; Zeadally, S. Autonomous Cars: Research Results, Issues, and Future Challenges. *IEEE Communications Surveys & Tutorials*, **2019**, vol. 21, no. 2, pp. 1275-1313
3. Parekh, D.; Poddar, N.; Rajpurkar, A.; Chahal, M.; Kumar, N.; Joshi, G.P.; Cho, W. A Review on Autonomous Vehicles: Progress, Methods and Challenges. *Electronics* **2022**, *11*, 2162
4. Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. *Sensors* **2021**, *21*, 2140
5. Fayyad, J.; Jaradat, M.A.; Gruyer, D.; Najjaran, H. Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review. *Sensors* **2020**, *20*, 4220
6. Ahangar, M.N.; Ahmed, Q.Z.; Khan, F.A.; Hafeez, M. A Survey of Autonomous Vehicles: Enabling Communication Technologies and Challenges. *Sensors* **2021**, *21*, 706
7. Zhu, Z.; Du, Q.; Wang, Z.; Li, G. A Survey of Multi-Agent Cross Domain Cooperative Perception. *Electronics* **2022**, *11*, 1091
8. Shan, M.; Narula, K.; Wong, Y.F.; Worrall, S.; Khan, M.; Alexander, P.; Nebot, E. Demonstrations of Cooperative Perception: Safety and Robustness in Connected and Automated Vehicle Operations. *Sensors* **2021**, *21*, 200
9. Ngo, H.; Fang, H.; Wang, H. Cooperative Perception With V2V Communication for Autonomous Vehicles. *IEEE Transactions on Vehicular Technology*, **2023**, vol. 72, no. 9, pp. 11122-11131
10. Xiang, C. et al. Multi-Sensor Fusion and Cooperative Perception for Autonomous Driving: A Review. *IEEE Intelligent Transportation Systems Magazine*, **2023**, vol. 15, no. 5, pp. 36-58
11. Q. Chen; S. Tang; Q. Yang; Fu, S. Cooper: Cooperative Perception for Connected Autonomous Vehicles Based on 3D Point Clouds. 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS), Dallas, TX, USA, 2019.
12. Qi Chen et al. F-cooper: feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds. Proceedings of the 4th ACM/IEEE Symposium on Edge Computing. Association for Computing Machinery, New York, NY, USA, 2019.
13. Sun, P.; Sun, C.; Wang, R.; Zhao, X. Object Detection Based on Roadside LiDAR for Cooperative Driving Automation: A Review. *Sensors* **2022**, *22*, 9316.
14. Bai, Z.; Wu, G.; Qi, X.; Liu, Y.; Oguchi, K.; Barth, M.J. Infrastructure-Based Object Detection and Tracking for Cooperative Driving Automation: A Survey. 2022 *IEEE Intelligent Vehicles Symposium (IV)*, Aachen, Germany, 2022.
15. Lee, S. A Study on the Structuralization of Right Turn for Autonomous Driving System. *Journal of Korean Public Police and Security Studies*, **2022**, vol. 19, no. 3, pp. 173-190

16. Park, S.; Kee, S.-C. Right-Turn Pedestrian Collision Avoidance System Using Intersection LiDAR. EVS37 Symposium, COEX, Seoul, Korea, April 23–26, 2024.
17. Martin, S. et al. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. Proceedings of the European conference on computer vision (ECCV) workshops, Munich, Germany, 2018.
18. Shaoshuai, S. et al. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, Seattle, Washington, USA, 2020.
19. CARLA official site. Available online: <https://CARLA.org/> (accessed on 2017).
20. Alexey, D. et al. CARLA: An open urban driving simulator. Proceedings of Machine Learning Research, Mountain View, California, USA, November 13–15, 2017.
21. Gómez-Huélamo, C.; Del Egido, J.; Bergasa, L.M. et al. Train here, drive there: ROS based end-to-end autonomous-driving pipeline validation in CARLA simulator using the NHTSA typology. *Multimed Tools Appl* **2022**, *81*, 4213–4240.
22. Rosende, S.B.; Gavilán, D.S.J.; Fernández-Andrés, J.; Sánchez-Soriano, J. An Urban Traffic Dataset Composed of Visible Images and Their Semantic Segmentation Generated by the CARLA Simulator. *Data* **2024**, *9*, 4.
23. Lee, H.-G.; Kang, D.-H.; Kim, D.-H. Human–Machine Interaction in Driving Assistant Systems for Semi-Autonomous Driving Vehicles. *Electronics* **2021**, *10*, 2405.
24. ROS(Robot operating system) official site. Available online: <https://www.ros.org/> (accessed on 2007)
25. Rosique, F.; Navarro, P.J.; Fernández, C.; Padilla, A. A Systematic Review of Perception System and Simulators for Autonomous Vehicles Research. *Sensors* **2019**, *19*, 648.
26. CARLA blueprint document site. Available online: https://CARLA.readthedocs.io/en/latest/bp_library/ (accessed on 2017)
27. CARLA-ROS bridge package official site. Available online: <https://github.com/CARLA-simulator/ros-bridge> (accessed on 2018)
28. Vargas, J.; Alsweiss, S.; Toker, O.; Razdan, R.; Santos, J. An Overview of Autonomous Vehicles Sensors and Their Vulnerability to Weather Conditions. *Sensors* **2021**, *21*, 5397.
29. Torres, P.; Marques, H.; Marques, P. Pedestrian Detection with LiDAR Technology in Smart-City Deployments—Challenges and Recommendations. *Computers* **2023**, *12*, 65.
30. B. Lv et al. LiDAR-Enhanced Connected Infrastructures Sensing and Broadcasting High-Resolution Traffic Information Serving Smart Cities. *IEEE Access*, vol. 7, pp. 79895–79907, 2019.
31. Yaqoob, I.; Khan, L. U.; Kazmi, S. M. A.; Imran, M.; Guizani, N.; Hong, C. S. Autonomous Driving Cars in Smart Cities: Recent Advances, Requirements, and Challenges. *IEEE Network*, vol. 34, no. 1, pp. 174–181

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.