



## Article

# Ship Detection in Maritime Scenes under Adverse Weather Conditions

Qiuyu Zhang <sup>1</sup>, Lipeng Wang <sup>1,\*</sup>, Hao Meng <sup>1</sup>, Zhi Zhang <sup>1</sup> and Chunsheng Yang <sup>2</sup>

<sup>1</sup> The College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; s313047015@hrbeu.edu.cn (Q.Z.); menghao@hrbeu.edu.cn (H.M.); zhangzhi1981@hrbeu.edu.cn (Z.Z.)

<sup>2</sup> National Research Council Canada, Ottawa, ON K1A 0R6, Canada; chunsheng.yang@carleton.ca

\* Correspondence: wanglipeng@hrbeu.edu.cn

**Abstract:** Point cloud-based detection focuses on land traffic, rarely marine, facing issues with ships: it struggles in bad weather due to reliance on adverse weather data and fails to detect ships effectively due to overlooking size and appearance differences. Addressing the above challenges, our work introduces point cloud data of marine scenarios under realistically simulated adverse weather conditions and a dedicated Ship Detector tailored for marine environments. To adapt to various maritime weather conditions, we simulate realistic rain and fog in collected marine scene point cloud data. Additionally, addressing the issue of losing geometric and height information during feature extraction for large objects, we propose a Ship Detector. It employs a dual-branch sparse convolution layer for extracting multi-scale 3D feature maps, effectively minimizing height information loss. Additionally, a multi-scale 2D convolution module is utilized, which encodes and decodes feature maps and directly employs 3D feature maps for target prediction. To reduce dependency on existing data and enhance model robustness, our training dataset includes simulated point cloud data representing adverse weather conditions. In maritime point cloud ship detection, our Ship Detector, compared to adjusted small object detectors, demonstrates the best performance.

**Keywords:** object detection; LiDAR; autonomous driving; data simulation



**Citation:** Zhang, Q.; Wang, L.; Meng, H.; Zhang, Z.; Yang, C. Ship Detection in Maritime Scenes under Adverse Weather Conditions. *Remote Sens.* **2024**, *16*, 1567. <https://doi.org/10.3390/rs16091567>

Academic Editor: Joaquín Martínez-Sánchez

Received: 18 March 2024

Revised: 17 April 2024

Accepted: 25 April 2024

Published: 28 April 2024



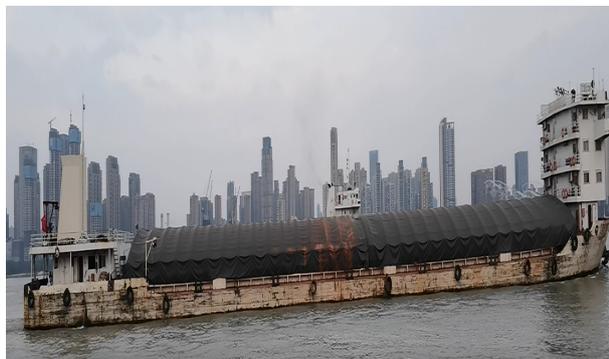
**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Deep learning has significantly advanced the field of image object detection. However, 2D images do not capture depth perception and are sensitive to changes in lighting conditions [1]. To surmount these limitations associated with spatial information and adverse weather, the application of 3D LiDAR point clouds is gaining importance. Nevertheless, current research in 3D perception largely focuses on terrestrial objects [2–5], such as cars, pedestrians, and bicycles, and currently, there is no research on ship detection in maritime point cloud scenarios. Environmental perception applied to the marine environment is distinctly different from that applied to terrestrial environments. Firstly, the objects of perception are different, with the volume of ships being tens or even dozens of times larger than that of vehicles. Second, the appearance of perception targets is different. Vehicles tend to have similar appearances, whereas ships, even within the same category, can vary greatly in appearance due to their design or the cargo they carry as shown in Figure 1. Additionally, marine environments often encounter adverse weather conditions. Therefore, object detectors designed for land use often fail to meet the needs of ship detection. This necessitates the development of methods specifically tailored for the marine environment.

LiDAR data collected under adverse weather are inherently complex and chaotic, encompassing an abundance of noise and outliers. In conditions of adverse weather, including rain and fog, the strength of the laser pulses produced by LiDAR sensors decreases significantly with their journey through the atmosphere. When the LiDAR cannot penetrate aerosol particles present in the air, it generates noises. This results in data collection

uncertainties due to weather and distance, leading to potential target misidentification or false detection. Hence, environmental perception in challenging weather conditions continues to be a major hurdle in this domain. One solution to tackle this challenge involves gathering data from point clouds during rainy and foggy weather, although this strategy is expensive and poses difficulties in labeling [6,7]. However, no works have focused on point cloud datasets for marine scenes under adverse weather conditions. Another method is to simulate adverse weather laser point cloud scenes using a physically based approach with point cloud data collected in clear weather [8–11]. Ref. [12] conducts a quantitative analysis of the impact of rain on laser sensors and develops a mathematical model for LiDAR degradation. Ref. [6] simulates fog by randomly selecting points within a normal distribution, a method deemed overly simplistic. Ref. [13] enhances LiDAR data under adverse weather conditions using a hybrid Monte Carlo method. Hahner et al. [14,15] develop a physical model considering the scattering rates and attenuation of snow and fog particles, LiDAR range, distance accuracy, and laser wavelength to simulate fog and snow. However, these methodologies are designed for vehicular LiDAR sensors within a 150 m urban environment, while our goal is to simulate marine scenes over a range of 1 nautical mile.



(a) Cargo ships



(b) Tour boats

**Figure 1.** Display of ship images from the training dataset, with (a) representing Cargo Ships and (b) representing Tour Boats.

Point cloud data directly provide the shape, size, and spatial position of objects in three-dimensional space. Zhang et al. [16] utilize 3D point cloud data to identify overloaded ships. Lu et al. [17] conduct the precise tracking of key points on ships to facilitate intelligent docking. In 3D point cloud detection, objects are delineated through annotated three-dimensional bounding boxes naturally and effectively [3]. The rigidity of objects within point cloud scenes confers an inherent invariance, facilitating the detection of targets across various scales and distances. Consequently, the utilization of point cloud data is essential for accurate ship identification. However, ships typically exceed vehicles in size, and their

representations in 3D space differ markedly, indicating that detectors designed for vehicles may not accommodate the larger scale and complex structure of ships. Moreover, ship detection often requires a broader spatial range, and the maritime environment presents unique features, such as waves, reflections, and varying weather conditions. Vehicle detectors [2–5] may struggle to process such extensive spatial data, which are fundamentally different from terrestrial environments. These factors pose significant challenges for 3D ship detection in maritime scenarios. Vehicle-based detection methods employ sophisticated 3D detection structures to map point clouds onto a bird's eye view (BEV) [18], followed by 2D convolution for target prediction. This direct transformation of 3D features into 2D representations can lead to a substantial loss of appearance information, particularly regarding height, due to variations in the target size. Furthermore, a significant amount of geometric information can be lost during the 2D convolution process.

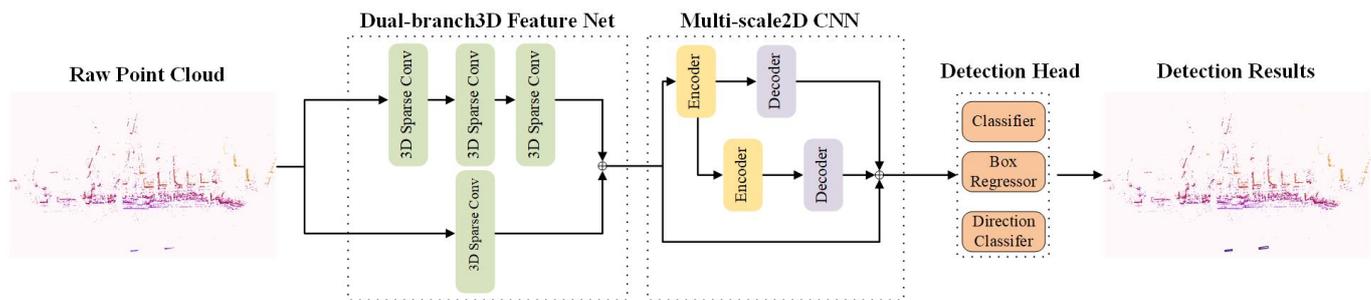
Adverse weather conditions lead to a reduction in LiDAR functionality. Incorporating data from rainy and foggy conditions into the dataset helps improve the performance of detectors. This work simulates point clouds under adverse weather conditions through physical simulation of the collected ocean scene point clouds. The sparsity of point clouds makes extracting features from point cloud data challenging, especially in marine environments, where the detection range is broader and the sparsity is even greater. Three-dimensional sparse convolution is a specialized convolutional processing method designed for sparse data in three-dimensional spaces, suitable for situations where the data are unevenly distributed with most areas being empty. Second [2] is a classic network that uses sparse convolution for small target detection. It segments point clouds into voxels and employs sparse convolution to extract three-dimensional point cloud features, which are then fed into a 2D convolution network to extract 2D features. Second uses only one channel to downsample the point cloud for extracting 3D features, which is not suitable for large-scale targets. Furthermore, using only the 2D features processed through the 2D convolution network for target localization results in a loss of 3D information. In this work, we introduce a pioneering detector designed for maritime scenarios, capable of directly processing large-scale three-dimensional point clouds for ship detection. The overall framework of the detector is shown in Figure 2. Tailored to accommodate vessels spanning several tens of meters, our detector adjusts the voxel size accordingly. To preserve the geometric characteristics of the objects more effectively, we employ a dual-branch sparse convolution layer approach. The 3D point clouds undergo processing through two distinct channels utilizing 3D sparse convolution: one branch undergoes sequential downsampling, each stage reducing the scale by half, while the other branch undergoes only one stage of downsampling, reducing the input to one-eighth of its original size. Sparse features from both branches are stacked for subsequent 2D feature extraction. To further retain detailed features, the multi-scale 2D feature extraction is employed, where the sparse features are encoded and decoded, and directly concatenated with the original sparse features for prediction. Additionally, to enhance the detector's understanding of the scene, we simulate point cloud data of maritime scenarios under adverse weather conditions for training. We also adapt vehicle-specific detectors for maritime use by fine-tuning their parameters and evaluating their performance using a maritime point cloud dataset. Our tailored detector demonstrates superior performance, excelling in all metrics.

The key contributions of our work are as follows:

- We propose a physical model for simulating LiDAR point clouds of maritime scenes under different adverse weather conditions.
- We introduce the first detector specifically engineered for maritime vessel detection. Concurrently, we have adapted the parameters of existing vehicle detectors to enable their training with maritime point cloud datasets. Our Ship Detector achieves state-of-the-art results, surpassing the performance of these modified vehicle detectors.
- We propose a dual-branch sparse convolution layer method that effectively preserves more of the ship's appearance characteristics.

- We introduce a multi-scale 2D feature extraction technique that extracts 2D features and, in addition, directly utilizes 3D sparse features for prediction.

The structure of the subsequent sections of this work is as follows: Section 2 provides work related to this paper, Section 3 outlines the realistic simulation under adverse conditions, Section 4 introduces the Ship Detector specifically designed for maritime scenes, Section 5 elaborates on the basic content of the experimental design, network parameter design, ablation experiments, and experimental results, and Section 6 and Section 7 discuss and conclude this work, respectively.



**Figure 2.** Overall framework of Ship Detector. Our detector takes raw point cloud data as input and outputs prediction results.

## 2. Related Work

### 2.1. Data Augmentation

The quantity and quality of data play a crucial role in achieving a high-performance detector [19]. Initially, techniques borrowed from 2D image enhancement [20–22], such as rotation, scaling, and adding noise to point coordinates, are applied to 3D point clouds. These methods, while straightforward, are essential for enhancing the models' robustness against variations in point cloud data. Due to the development of deep learning, it is gradually being applied to the field of data augmentation. Generative Adversarial Networks (GANs) [23] have been utilized to generate synthetic point cloud data or learn optimal augmentation strategies [24–26], aiming to create more diverse and challenging data scenarios. PC-GAN [24], one of the early attempts to apply GANs directly to point clouds, learns to generate point clouds using hierarchical Bayesian modeling and implicit generative models. Following this, Pointflow [27] proposes generating point clouds using normalizing flows, which can simulate a continuous distribution of point cloud data and excel in generating complex shapes. PointMixup [28] introduces a novel data augmentation approach by interpolating between examples through shortest-path linear interpolation, thereby creating new examples by optimal path function assignment. Polarmix [29] further enriches point cloud distribution and maintains fidelity using two cross-scan augmentation strategies, demonstrating a general approach to LiDAR point cloud data augmentation. These methods enhance data complexity and realism by understanding the underlying structure and characteristics of point clouds.

### 2.2. Adverse Weather LiDAR Data

Although data augmentation can address the issue of insufficient samples in datasets, these methods expand upon the existing data. Widely used datasets such as KITTI [8], H3D [11], nuScenes [9], and Waymo [10] are collected under clear weather conditions, making it challenging to obtain data under adverse weather conditions using data augmentation methods. Weather like fog and rainfall reduce visibility significantly and have a major impact on the safety of autonomous driving [30]. However, point cloud data for adverse weather conditions are scarce. Considering the physical principles of LiDAR signal interference, Ref. [31] starts with a model-based description of soft and hard LiDAR targets, deriving the fundamental factors that affect the weather performance of modern automotive LiDAR systems, and providing a theoretical foundation for subsequent point

cloud simulation. LIBRE [32] is collected using various brands of LiDAR indoors under different conditions (fog, rain, and strong light). Due to the expensive collection and annotation costs, methods [13–15] have been proposed to simulate adverse weather data based on physical simulation using clear weather data. Ref. [13] averages the effects of small scatterers and randomly places large particles to simulate real rainy scene point clouds. Ref. [14] proposes a physically valid fog simulation method that completely controls all parameters involved in the physical equations in the simulation, realistically simulating fog of any density. Following this, the work [15] introduces a methodology grounded in physics for replicating the effects of snowfall on LiDAR point clouds collected in clear weather conditions.

### 2.3. Three-Dimensional Object Detection

In recent years, the focus has increasingly shifted towards 3D object detection utilizing LiDAR, as it offers enhanced depth information compared to traditional imaging techniques. PointNets [33] and PointNets++ [34], the first networks to directly process point cloud data, have been proposed, utilizing Farthest Point Sampling (FPS) to downsample point sets while also preserving the rigid features of the targets. Meanwhile, VoxelNet [35] has been introduced, which divides the entire scene's point cloud into voxels, employing methods of random dropping or adding to ensure a uniform number of points within each voxel. Second [2] builds on VoxelNet, accelerating sparse 3D convolution to improve inference speed. PointPillars [5] changes voxels into vertical pillars, organizing the point cloud representation and encoding it using the PointNets network. CenterPoint [4] discards manually set anchors and uses heatmaps to locate targets. VoxelNet [36] seeks to minimize computational demands by directly detecting objects from voxel attributes, removing the necessity for elements like anchor intermediaries, conversion from sparse to dense, networks for proposing regions, and various other components associated with dense predictions, thus streamlining the detection workflow. Additionally, Refs. [37,38] apply the Detection Transformer (DETR) to LiDAR 3D detection tasks, generating predictions for each object query.

## 3. Realistic Simulation of Adverse Weather Conditions

LiDAR (Light Detection and Ranging) employs laser pulses to measure distances and create precise, detailed three-dimensional representations of the area around it. This technology operates on a fundamental method where a laser pulse is emitted and the duration for its return is recorded after reflection from a surface. By analyzing the return time and the reflected light, LiDAR systems can generate precise information about the shape and characteristics of surfaces within their field of view. During maritime navigation, adverse weather is frequently encountered. In such weather scenarios, visibility diminishes, impacting LiDAR sensors' functionality and resulting in imprecise distance measurements of objects. Unfortunately, the 3D LiDAR point clouds we collected for marine scenes were acquired under clear weather. Therefore, in this section, we simulate marine scenes under adverse weather conditions using physical modeling methods based on the principles of LiDAR sensors.

Section 3.1 outlines the optical model under clear weather conditions. Section 3.2 introduces the method for foggy simulation. Section 3.3 describes the method for rainy simulation.

### 3.1. Optical Model under Clear Weather

To mimic the impact of fog on a LiDAR point cloud captured in fair weather, one must first understand the optical system model of both the LiDAR sensor's transmitter and receiver [31].

The LiDAR system emits a pulsed signal and measures the time  $t$ , referred to as TOF (Time of Flight), it takes for the signal to travel from the emission to the target and back. This time measurement is then used to calculate the distance to the target. The product of time  $t$  and the speed of light  $c$  provides the distance  $ct/2$  traveled by the signal.

The received signal power  $P(R)$ , which varies with distance, is modeled as the temporal convolution of the transmitted signal power  $P_0$  and the time-varying environmental impulse response  $H$ :

$$P(R) = C_A \int_0^{2R/c} P_0 H(R - ct/2) dt \quad (1)$$

where  $C_A$  is a system constant independent of time and range.

The time-varying environmental impulse response  $H$  is defined as the product of the single impulse responses of the optical channel  $H_C$ , and the target  $H_T$ :

$$H(R) = H_C(R)H_T(R) \quad (2)$$

The single impulse response of the optical channel  $H_C$  is shown as Equation (3):

$$H_C(R) = \frac{T^2(R)}{R^2} \quad (3)$$

where  $T(R)$  denotes the total unidirectional transmission loss and is described as Equation (4):

$$T(R) = \exp\left(\int_{r=0}^R -\alpha(r) dr\right) \quad (4)$$

where  $\alpha(r)$  denotes the attenuation coefficient of spatial variation. In general, the attenuation coefficient is set to a homogeneous optical medium  $\alpha(r) = \alpha$ . Therefore, the total unidirectional transmission loss  $T(R)$  is defined as shown in Equation (5):

$$T(R) = \exp(-\alpha R) \quad (5)$$

The attenuation coefficient  $\alpha$  is influenced by the weather conditions during measurement and is inversely proportional to the visibility range of the scene. Specifically, as visibility decreases,  $\alpha$  increases. In clear weather conditions, we define  $\alpha = 0$ , resulting in a total unidirectional transmission loss  $T(R) = 1$ . Therefore, the total unidirectional transmission loss in clear weather is described as Equation (6):

$$H_C(R) = \frac{1}{R^2}. \quad (6)$$

Solid objects that effectively reflect LiDAR pulses, such as buildings, ships, or any other substantial structures with clear physical boundaries, are defined as hard targets [14]. The spatial impulse response  $H(R)$  describes the response of the laser LiDAR system to the reflection from the target. For hard targets, a simple model of the spatial impulse response at  $R_0$  is derived as Equation (7):

$$H_T(R) = H_T^{hard}(R) = \frac{\epsilon}{\pi} \delta(R - R_0) \quad (7)$$

With Lambertian reflectance properties, the differential reflectance is  $\beta_0 = \frac{\epsilon}{\pi}$ , where  $\epsilon \in (0, 1)$ .

By substituting Equations (6) and (7) into Equation (1), the received signal power with the given range of  $R_0$  conditions is obtained as Equation (8):

$$P_{clear}(R) = C_A \int_0^{2R/c} P_0 \frac{1}{R_0^2} \frac{\epsilon}{\pi} \delta(R - \frac{ct}{2} - R_0) dt = \frac{C_A P_0 \beta_0}{R_0^2} \int_0^{2R/c} \delta(R - \frac{ct}{2} - R_0) dt \quad (8)$$

### 3.2. Fog Simulation

In this section, we build the transition from clear weather to foggy. However, in the same 3D scene, hard targets are the invariant rigid bodies within the scene; hence, they exist in rainy or foggy conditions. However, due to the presence of water droplets or fog in the environment affecting the target response  $H_T$ , these are referred to as soft targets [14].

### 3.2.1. Foggy Weather Simulation Optical Model

In foggy weather conditions, along with the response from hard targets, there is also a response from fog-soft targets. The response from the fog-soft target  $H_T^{soft}$  is a Heaviside function of the form:

$$H_T^{soft}(R) = \beta U(R_0 - R) \quad (9)$$

where  $U$  denotes the Heaviside function. And  $\beta$  is the backscattering coefficient, which is set to a constant value under identical fog weather conditions.

Therefore, the self-impulse response of targets in foggy conditions is composed of the superposition of hard responses  $H_T^{hard}$  and soft target responses  $H_T^{soft}$ .  $H_T^{hard}$  represents the reflective characteristics of hard targets at a specific distance  $R_0$ . And  $H_T^{soft}$  describes the attenuation of the signal by fog before the target distance  $R_0$ :

$$H_T^{fog}(R) = H_T^{hard}(R) + H_T^{soft}(R) = \beta U(R_0 - R) + \beta_0 \delta(R - R_0) \quad (10)$$

Based on Equations (3) and (5), the optical channel in foggy conditions is obtained as follows:

$$H_C^{fog}(R) = \frac{\exp(-2\alpha R)}{R^2} \quad (11)$$

As shown in Equation (2), the optical channel impulse response  $H_C^{fog}(R)$  and the target impulse response  $H_T^{fog}(R)$  together constitute the time-varying environmental impulse response, which is expressed as follows:

$$H^{fog}(R) = \frac{\exp(-2\alpha R_0)}{R^2} \times (\beta U(R_0 - R) + \beta_0 \delta(R - R_0)) \quad (12)$$

The received signal power in foggy conditions is more complex than in clear weather, but for simplification, it is represented as the combination of the impulse responses of hard and soft targets. The hard target response in fog is an attenuation of the original clear weather response. Consequently, the received signal power of hard targets in foggy conditions is the product of the total unidirectional transmission loss  $T^2(R)$  and the received signal power under clear weather conditions  $P_{clear}(R)$ :

$$P_{fog}^{hard}(R) = \exp(-2\alpha R_0) P_{clear}(R) \quad (13)$$

Given that the speed of light is  $c$  and the propagation time is  $t$ , the position of the soft target at this time is  $R - ct/2$ . Therefore, the received signal power in foggy conditions is as follows:

$$\begin{aligned} P_{fog}^{soft}(R) &= C_A \int_0^{2R/c} P_0 H_C^{soft}(R - ct/2) H_T^{soft}(R - ct/2) dt \\ &= C_A P_0 \beta \int_0^{2R/c} \frac{\exp(-2\alpha(R - ct/2))}{(R - ct/2)^2} U(R_0 - R + ct/2) dt \end{aligned} \quad (14)$$

We develop an algorithm as shown in Algorithm 1 to simulate LiDAR in foggy conditions, which transforms point clouds from clear weather into foggy weather point clouds. The input for the algorithm includes the coordinates  $x, y, z$ , intensity  $P_{clear}$ , attenuation coefficients  $\alpha$ , and  $\beta$ , backscattering coefficient  $\beta_0$ , and time  $t$  in clear weather. By calculating the received power of hard and soft targets in foggy conditions and adjusting the position of the point cloud, the algorithm effectively simulates how heavy fog affects the visibility and appearance of objects in LiDAR data.

**Algorithm 1** LiDAR simulation in foggy conditions

---

**Input:**  $x, y, z, P_{clear}, \alpha, \beta, \beta_0, t$

- 1: **for**  $dist$  in  $(50, 100, 150, \dots, 1800)$  **do**
- 2:    $R_0 \leftarrow \sqrt{x^2 + y^2 + z^2};$
- 3:    $C_A P_0 \leftarrow \frac{P_{clear} R_0^2}{\beta_0};$     $\backslash \backslash$  According to Equation (8).
- 4:    $P_{fog}^{hard} \leftarrow \exp(-2\alpha R_0) P_{clear};$     $\backslash \backslash$  According to Equation (13).
- 5:   **for**  $R$  in  $(0, 0.5, 1.0, \dots, R_0)$  **do**
- 6:      $P_R, R_R \leftarrow SIMPSON(I(R, R_0, \alpha, t));$     $\backslash \backslash$  According to Equation (14).
- 7:   **end for**
- 8:    $P_{tmp}, R_{tmp} \leftarrow \max(I_R);$
- 9:    $P_{fog}^{soft} \leftarrow C_A P_0 \beta P_{tmp};$
- 10:   **if**  $P_{fog}^{soft} > P_{fog}^{hard}$  **then**
- 11:      $s \leftarrow (R_{tmp} + dist) / R_0;$
- 12:      $x \leftarrow x * s;$
- 13:      $y \leftarrow y * s;$
- 14:      $z \leftarrow z * s;$
- 15:      $P \leftarrow P_{fog}^{soft};$     $\backslash \backslash$  Soft targets.
- 16:   **else**
- 17:      $P \leftarrow P_{fog}^{hard};$     $\backslash \backslash$  Hard targets
- 18:   **end if**
- 19:   **return**  $x, y, z, P$
- 20: **end for**

---

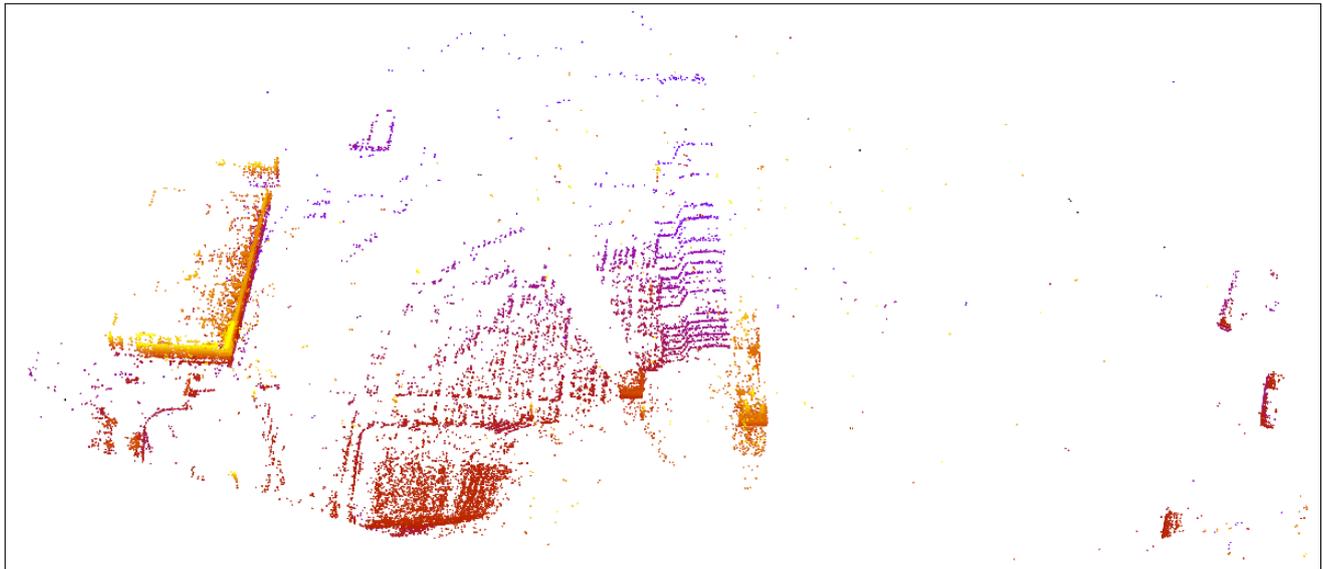
### 3.2.2. Foggy Weather Simulation Data Validation

Figure 3a,b depict 3D visualizations of the port scene when a sensor is placed in the same location during a foggy morning and a clear noon, respectively. Evidently, in Figure 3a, there is noise interference around the buildings due to the impact of the fog. The point cloud in Figure 3b exhibits less noise and clearer targets under clear weather conditions. The scanning distance is longer in clear weather than in foggy conditions because of the absence of particulate matter. Figure 3c simulates foggy weather data under clear weather data conditions. Figure 3c shows that when simulating more distant objects, the sparsity of the points resembles the sparsity of the actual point cloud in foggy conditions. The building area also simulates the noise distribution found in actual foggy weather collections.

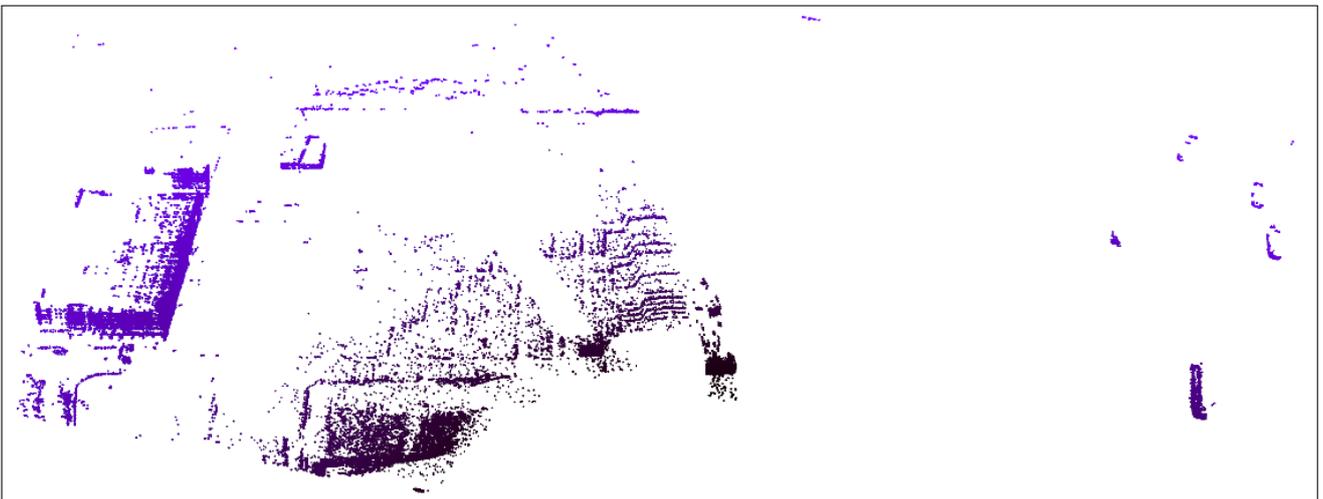
### 3.3. Rainy Simulation

#### 3.3.1. Rainy Weather Simulation Optical Model

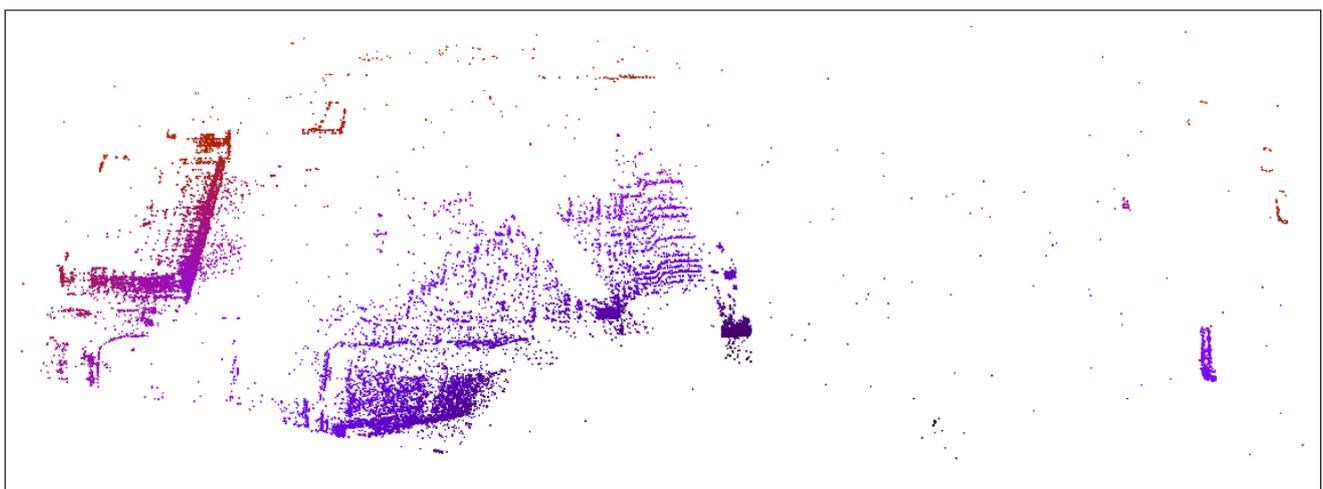
Raindrops cause scattering and attenuation of the laser signal. When the LiDAR laser beam passes through rain, a portion of the light energy emitted by the LiDAR sensor is absorbed or scattered by the raindrops. The laser beam needs to pass through more scattering media (such as water droplets), which leads to increased path loss, meaning greater energy loss of the laser during its propagation. Raindrops themselves may become a source of reflection for the laser beam, which can lead to the generation of, namely, false signals [12]. The laser beams emitted by LiDAR are capable of penetrating transparent materials such as glass and water, resulting in no reflective echoes. However, raindrops might contain impurities. Thus, to simplify, soft targets are not considered in our LiDAR simulation during rainy conditions.



(a) Collecting point clouds in foggy marine scenes.



(b) Collecting point clouds in clear weather marine scenes.



(c) Simulating point clouds in foggy marine scenes.

**Figure 3.** Display of marine scenes: (a) depicts the point cloud of a marine scene under foggy weather conditions, (b) shows the marine scene under clear weather conditions, (c) is a simulation of a marine scene in foggy weather.

The attenuation coefficient in rainy conditions, which is related to the rainfall rate, differs from the one in foggy conditions. Based on the data from LiDAR and rangefinders, we determine the theoretical relationship between the extinction coefficient and rainfall rate through an optimal fitting program, according to Lewandowski et al. [39]. Therefore, the extinction coefficient for rainy conditions is defined as follows:

$$\alpha = aRr^b \quad (15)$$

where  $Rr$  represents the rainfall rate in mm/h. And  $a$  and  $b$  are empirical coefficients.

The response of hard targets in rainy conditions is an attenuation of the original response in clear weather. The received power of the hard targets after refraction by rainwater is as shown in Equation (16):

$$P_{rain}^{hard}(R) = \exp(-2aRr^b R_0) P_{clear}(R) \quad (16)$$

Filgueira et al. [40] indicate that raindrops reduce the intensity of LiDAR echoes. Additionally, it is considered that rainfall introduces a certain amount of noise to the measured distance. This noise increases with the rate of rainfall, yet even at higher rainfall rates, the error remains within a small range. Due to the complexity involved in simulating the refraction and reflection of rainwater, we employ a normal distribution  $\mathcal{N}$  to approximate the deviation observed in the echo of the rays:

$$R = R_0 + \mathcal{N}(0.02R_0(1 - \exp(-Rr)^2)) \quad (17)$$

We develop an Algorithm 2 for simulating marine scene LiDAR in rainy conditions. As with foggy conditions, we use point cloud data from clear weather as input and transform them into rainy weather LiDAR point clouds through realistic physical simulation. In the rain simulation, since LiDAR rays can penetrate raindrops, we neglect soft targets and only consider hard targets. The algorithm's inputs include the coordinates  $x, y, z$ , backscattering coefficient  $\beta_0$ , intensity  $P_{clear}$ , and rainfall rate  $Rr$ , and empirical coefficients  $a$  and  $b$ . After obtaining the simulation rainy point cloud according to Algorithm 2, the points are downsampled to enhance the realism of the rainy weather simulation. Each point is represented by its  $x, y, z$  coordinates, and the distance  $R_0$  is calculated as  $R = \sqrt{x^2 + y^2 + z^2}$ . Based on the Gaussian,  $\text{pro} = \exp\left(-\frac{R^2}{2 \times \mu^2}\right)$ , where  $\mu = 600$  is the standard deviation. We compute the probability weight for each point, meaning that the further a point is from the origin, the lower the probability that it will be selected.

---

#### Algorithm 2 LiDAR rainy simulation.

---

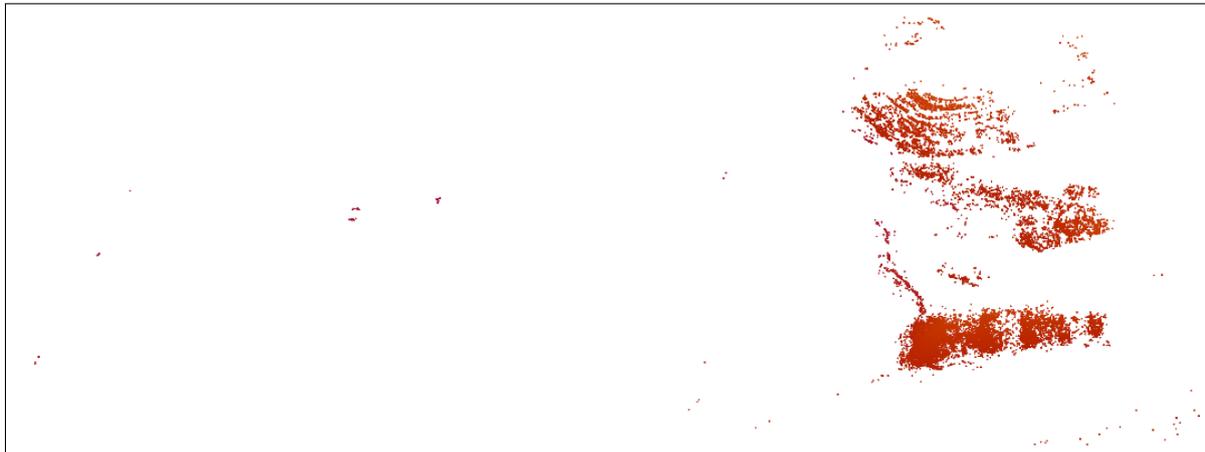
**Input:**  $x, y, z, P_{clear}, Rr, \beta_0, a, b, t$

- 1:  $R_0 \leftarrow \sqrt{x^2 + y^2 + z^2}$ ;
  - 2:  $C_A P_0 \leftarrow \frac{P_{clear} R_0^2}{\beta_0}$ ; \ \ According to Equation (8).
  - 3:  $P_{rain}^{hard} \leftarrow \exp(-2aRr^b R_0) P_{clear}$ ; \ \ According to Equation (16).
  - 4:  $R \leftarrow R_0 + \mathcal{N}(0.02R_0(1 - \exp(-Rr)^2))$ ; \ \ According to Equation (17).
  - 5:  $s \leftarrow R/R_0$ ;
  - 6:  $x \leftarrow x * s$ ;
  - 7:  $y \leftarrow y * s$ ;
  - 8:  $z \leftarrow z * s$ ;
  - 9:  $P \leftarrow P_{rain}^{hard}$ ;
  - 10: **return**  $x, y, z, P$
- 

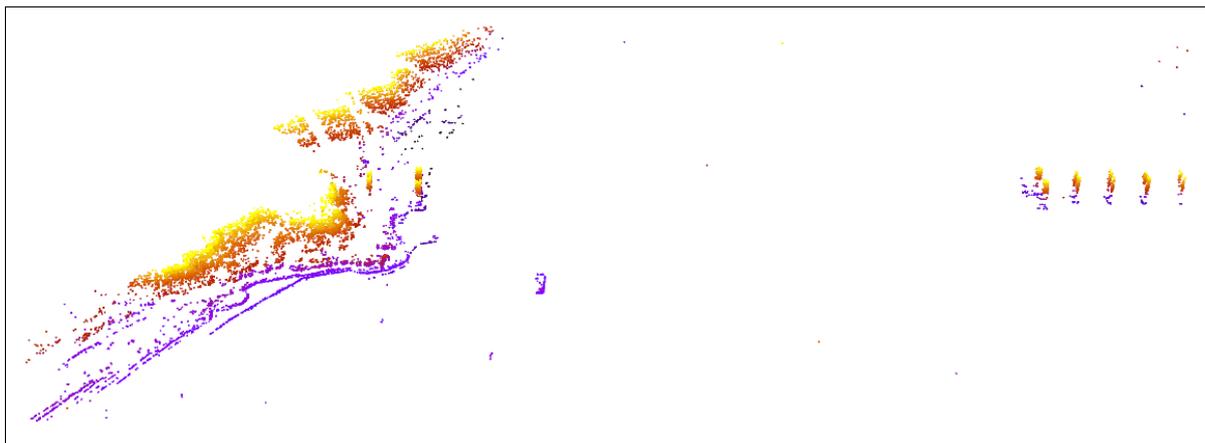
#### 3.3.2. Rainy Weather Simulation Data Validation

Due to data collection constraints, we did not obtain data for the same scene in both rainy and clear conditions. Fortunately, we managed to collect data on different island scenes in both weather conditions. Figure 4a shows a visualized point cloud of an island

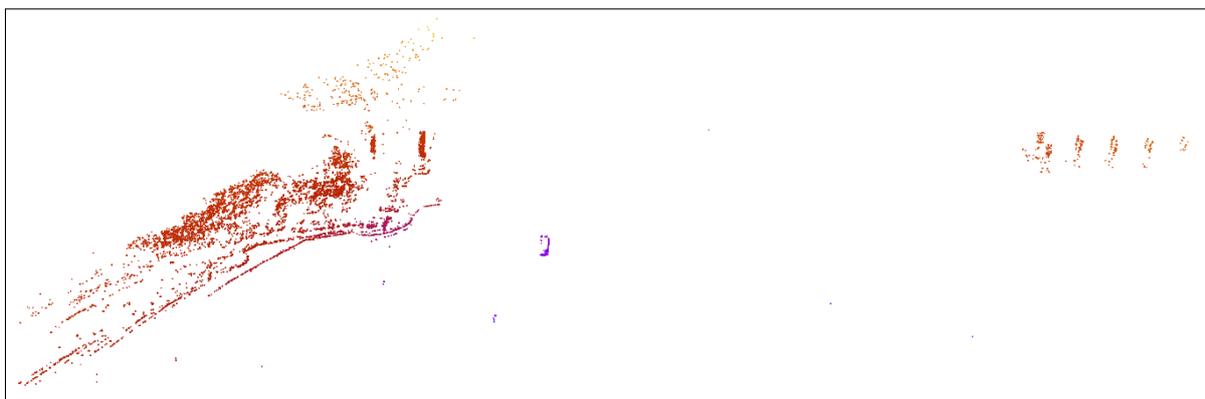
during rainy weather, and Figure 4b depicts the island scene under clear weather conditions. Fog contains impurities, resulting in noise during data collection. In rainy weather, the laser beams emitted by the LiDAR can penetrate through raindrops, but the scattering and refraction by the raindrops lead to a reduction in point collection. Following this principle, a rainy scene simulated using data from clear weather conditions is shown in Figure 4c.



(a) Collecting point clouds in rainy marine scenes.



(b) Collecting point clouds in clear weather marine scenes.



(c) Simulating point clouds in rainy marine scenes.

**Figure 4.** Example of a marine scene simulation in rainy weather: (a) shows a marine scene collected on a rainy day, (b) shows a marine scene captured in clear weather, and (c) is a simulated rainy scene derived from (b).

## 4. Ship Detector

The shipborne LiDAR sensors enable the acquisition of maritime point cloud scenes within a range of 1 nautical mile. In voxel-based target detection, as the detection range and target size increase, the voxel size is correspondingly enlarged. However, this results in the loss of appearance details during feature extraction, and a significant loss of height information during the compression of 3D information to 2D. In this section, we introduce the overall framework of our proposed Ship Detection and provide detailed information about the components of the detector.

Section 4.1 outlines the overall network structure of the Ship Detector. Section 4.2 describes the dual-branch sparse 3D feature extraction module used for extracting 3D features. Section 4.3 introduces the multi-scale 2D CNN module for extracting 2D features. Section 4.4 describes the settings of the ship anchors.

### 4.1. Network Architecture of the Ship Detector

The architecture of the Ship Detector is illustrated in Figure 2. Our detector is composed of three segments: a dual-branch sparse 3D feature extraction module, a multi-scale 2D CNN, and a detection head. The dual-branch sparse 3D feature extraction module processes the raw point cloud by converting it into voxel and coordinates, and then applies sparse 3D convolution to transform the point cloud into feature maps. Subsequently, the multi-scale 2D CNN extracts multi-scale features from these feature maps. Finally, the detection head is used for classification, regression, and direction prediction.

### 4.2. Dual-Branch Sparse 3D Feature Net

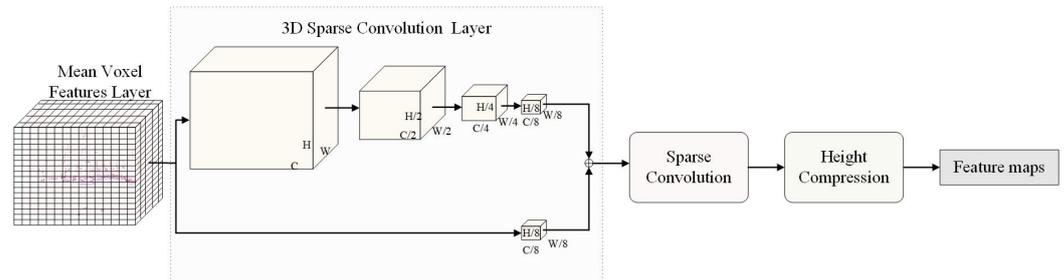
The dual-branch sparse 3D feature extraction net, as shown in Figure 5, is utilized to extract pointwise features from the raw input point clouds and subsequently fuse these into feature maps. In Figure 5, the input point cloud features are  $(x, y, z, r)$ , representing the coordinates and reflectivity of the point cloud. The output feature maps have dimensions of  $[128 \times H/16, W/8, L/8]$ , where  $H$ ,  $W$ , and  $L$  respectively denote the height, width, and length of the voxelized point cloud. It comprises three primary modules: the mean voxel feature module, the 3D sparse convolution module, and the height compression module. Each of these modules is further detailed in the following section.

We execute mean voxel feature extraction on the input point clouds via a mean voxel features layer, which segments the scene's point clouds into voxel representations. The cropping of the input marine scene is based on real-world surface dimensions along the XYZ axes, specifically within the range of  $[-320 \text{ m}, 480 \text{ m}] \times [0 \text{ m}, 320 \text{ m}] \times [-11 \text{ m}, 17 \text{ m}]$ . Simultaneously, voxel dimensions of  $[0.5 \text{ m} \times 0.4 \text{ m} \times 0.7 \text{ m}]$  are employed for spatial cropping. We sum up the features (i.e., the  $xyz$  coordinates) of each point within each voxel, and then calculate the average.

Vehicle detectors execute the pointwise feature by downsampling the voxel representation of the point cloud from  $[C \times H \times W]$  to  $[C/2 \times H/2 \times W/2]$  and then to  $[C/4 \times H/4 \times W/4]$  and finally to  $[C/8 \times H/8 \times W/8]$  [2,4]. During the downsampling process, crucial contour information is often lost, especially for large objects, such as ships, where the loss of height information becomes significantly pronounced across successive downsampling stages. To address these challenges, two convolutional layers with different scale lengths are employed for point-wise feature extraction. Initially, iterative pointwise feature extraction via four convolutional layers with a sampling scale of  $[1, 2, 2, 2]$  is applied. Additionally, the input mean voxel features are downsampled once with a scale of 8. This dual-branch scaling approach accommodates information across different 3D scales, capturing a wider array of input points and securing a larger receptive field, thereby enhancing the comprehension of global contextual information. Consequently, the dual-branch 3D feature extraction network more effectively preserves the features of the original input points.

We perform feature fusion on the features, containing pointwise features and corresponding coordinates, obtained from the convolutional network using stacking. The

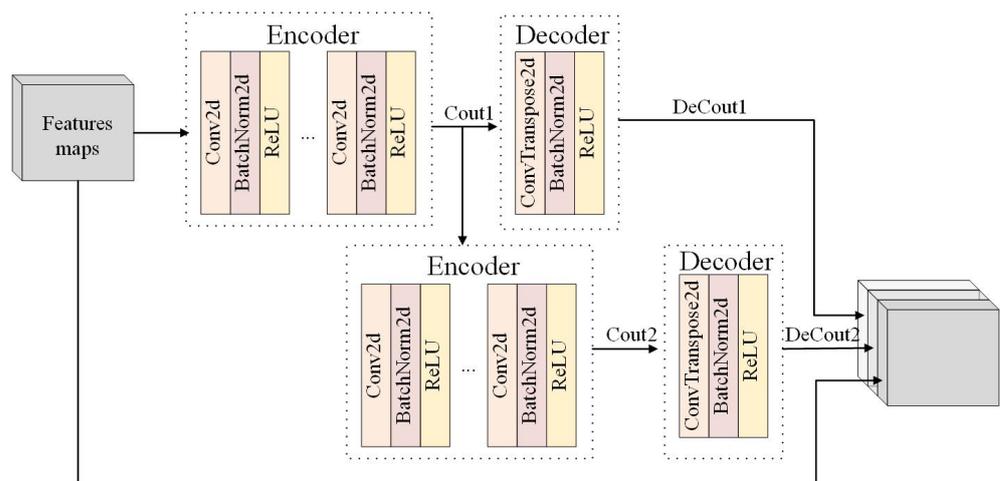
stacking features are then compressed into feature maps by a height compression method, which is fed as input to the subsequent 2D convolutional module.



**Figure 5.** The structure of the dual-branch sparse 3D feature extraction module. The network processes the raw point cloud data as input. It employs the mean voxel features layer to transform the marine point cloud into mean voxel features and their corresponding coordinates. Subsequently, these voxel features undergo pointwise feature extraction through 3D sparse convolution. A feature map is then produced by the height compression module.

#### 4.3. Multi-Scale 2D CNN Module

We employ a multi-scale 2D CNN to construct the feature extraction layer of the Region Proposal Network (RPN) [41]. Contrary to other detectors, the 2D features, used in our RPN for predicting categories, regression, and direction, are distinctive and are not only derived from the extraction process applied to feature maps but also include the feature maps themselves. As illustrated in Figure 6, the 2D feature extraction process comprises three main parts. The input is feature maps with dimensions  $[128 \times H/16, W/8, L/8]$ , and the output is a 2D feature with dimensions  $[768, W/8, L/8]$ , where  $H$ ,  $W$ , and  $L$  respectively denote the height, width, and length of the voxelized point cloud. In the first part, the feature map goes through one encoding and one decoding. In the second part, it experiences two encodings and one decoding. In the third part, the feature map undergoes no encoding or decoding and is used directly. The encoder consists of convolutional layers, BatchNorm, and ReLU layers. The decoder is composed of inverse convolution layers, BatchNorm, and ReLU layers.



**Figure 6.** The structure of the multi-scale 2D CNN module. The encoder layers, consisting of convolution, batch normalization, and ReLU, progressively compress the data, encoding the spatial hierarchy. Decoders then upsample these encoded features, aiming to preserve critical spatial information. The feature maps and the features after encoding and decoding are combined as output.

#### 4.4. Ship Anchor Setting

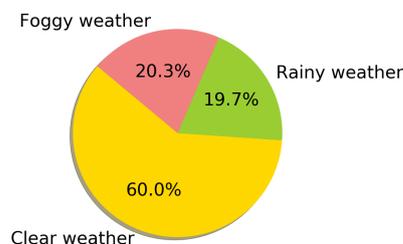
Upon statistical analysis of all labeled ships in the training dataset, we observe that their dimensions are similar. We use fixed-size anchors that are determined by averaging the dimensions and center positions of all ground truths in the ship's training set, including rotations of 0 and 90 degrees. The anchor dimensions used for cargo ships are  $w = 93 * l = 45 * h = 16$  m, for engineering ships,  $w = 54 * l = 18 * h = 20$  m, for tourist boats,  $w = 81 * l = 30 * h = 21$  m, and for speedboats,  $w = 25 * l = 5 * h = 6$  m. Our network utilizes the same strategy as Second [2] for setting box encoding and loss.

### 5. Experiments

In this section, we conduct extensive experiments and analysis to validate the effectiveness of our ship detection network under both clear and adverse weather conditions. This section is structured into five segments. Sections 5.1 and 5.2 respectively introduce the marine scene 3D point cloud dataset and the evaluation metrics for object detection. Section 5.3 presents the network parameter settings details of the Ship Detector. Section 5.4 displays the detection results under clear and adverse weather conditions. Finally, in Section 5.5, we conduct ablation studies to discuss the effectiveness of the detector. Our inference computational environment is a 1080Ti GPU.

#### 5.1. Ship Point Cloud Dataset

Owing to the mounting angle of the sensor, the collected data encompass marine scenes within a range extending 1 nautical mile in the positive direction of the Y axis. The clear weather conditions datasets include four types of ships: 'Cargo Ship', 'Engineering Vessel', 'Tour Boat', and 'Speedboat'. They are divided into 6964 frames for training and 1425 frames for testing. Clear weather point clouds are converted into adverse weather point clouds using the simulation methods for foggy and rainy conditions described in Section 4. The proportions of training data for clear, rainy, and foggy conditions in the adverse weather dataset are shown in Figure 7.



**Figure 7.** Pie chart of the proportion of clear weather, rainy weather, and foggy weather in the training data.

#### 5.2. Evaluation Metrics

For the 3D ship object detection experiments, the average precision (AP) of the LiDAR point clouds taken as the 'Cargo ships' benchmark is set at 0.7. We redefine the evaluation metrics for object detection. The detectors are evaluated at three difficulty levels: easy, moderate, and hard. The difficulty assessment is based on the object occlusion and truncation statuses in the 3D results. A truncation value less than 0.15 is defined as 'Easy', values from 0.15 to 0.3 are 'Moderate', and those exceeding 0.3 are 'Hard'.

#### 5.3. Network Architecture

We employ scene point clouds as input; however, the extensive detection range leads to GPU memory insufficiency. Consequently, we define the scene range as  $[-320 \text{ m}, 480 \text{ m}] \times [0 \text{ m}, 320 \text{ m}] \times [-11 \text{ m}, 17 \text{ m}]$ . The voxel dimensions are specified as  $[0.5 \text{ m} \times 0.4 \text{ m} \times 0.7 \text{ m}]$ , resulting in input dimensions along the XYZ axes of  $[1600, 400, 41]$ . The

input features are of size  $[N, 4]$ , where  $N$  represents the number of points, and 4 corresponds to the input features  $xyzr$ , with  $r$  denoting reflectivity. The raw data initially employ a mean voxel features layer for preliminary selection.

Each voxel is configured to contain exactly 32 points; surplus points are randomly discarded, while insufficient numbers are supplemented by random duplication. Subsequently, an average is calculated over these 32 points. The 3D sparse convolution layer processes data through two branches: one undergoes successive downsampling stages, resulting in dimensions of  $[1600, 400, 41]$ ,  $[800, 200, 21]$ ,  $[400, 100, 11]$ , and  $[200, 50, 5]$  with corresponding feature counts of 16, 32, 64, and 64, respectively. The alternate branch is subject to a single downsampling stage, yielding dimensions of  $[200, 50, 5]$  and a feature count of 16. The outputs of both branches are sparsely concatenated, forming a composite feature map with dimensions of  $[200, 50, 5]$  and an aggregate feature count of 64. This concatenated feature map is further refined through sparse convolution and height compression steps, producing a final feature map with dimensions of  $[batch, 256, 50, 200]$ , where '256' denotes the total number of features, spatially resolved to  $[50, 200]$ .

In the multi-scale 2D feature extraction process, three branches are incorporated. The first branch processes an initial feature map of dimensions  $[batch, 256, 50, 200]$  encoding it to reduce its size to  $[batch, 128, 50, 200]$ . This is subsequently passed through a decoder, restoring the size to  $[batch, 256, 50, 200]$ . The second branch starts with an encoded feature map of  $[batch, 128, 50, 200]$  and further encodes it to  $[batch, 256, 50, 200]$ , which is then decoded back to a size of  $[batch, 256, 50, 200]$ . The third branch utilizes the compressed feature map of  $[batch, 256, 50, 200]$ . By stacking the outputs from all three branches, a composite feature map with dimensions  $[batch, 768, 50, 200]$  is formed, serving as input for subsequent prediction tasks.

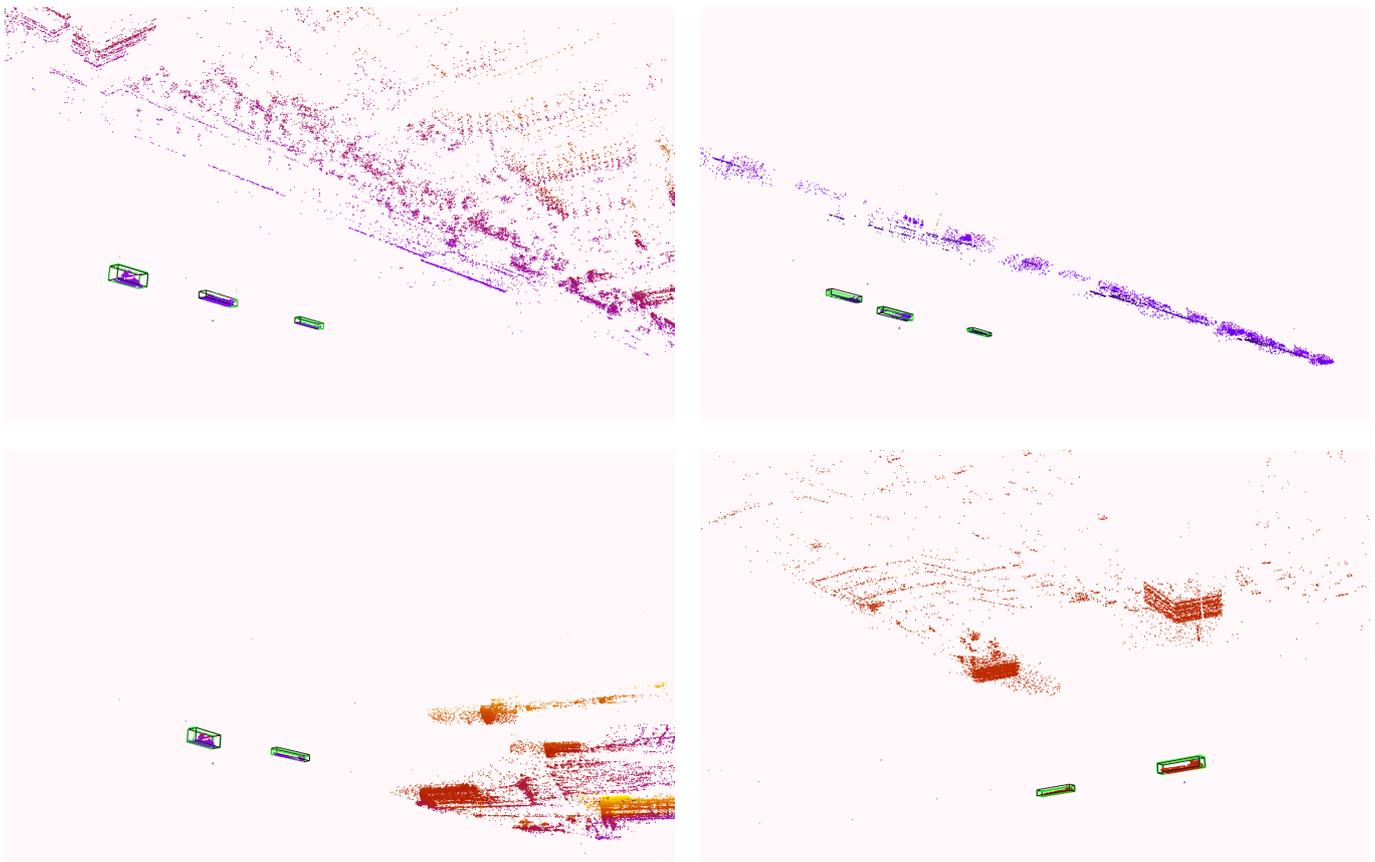
#### 5.4. Evaluation Using the Ship Point Cloud Dataset

To our knowledge, few studies are focusing on ship detection in marine scenes using LiDAR point cloud data. As our Ship Detector utilizes a voxel-based method for feature extraction, Second [3], a typical representative of voxel-based methods for detecting small objects, is chosen, and its parameters are adjusted to serve as a benchmark. Second uses 3D sparse convolutional layers to extract 3D features from voxelized data, which are then passed into a 2D CNN network for further feature extraction. We also select some representative small object detection networks for experimental comparison. Not all networks can detect ships effectively, as CIA-SSD [42], VirConv [43], and TED [44] show poor performance. Second [2] and PointRcnn [3] employ a voxel size of  $v_D = 0.5 * v_H = 0.4 * v_W = 0.7$  m. PointPillar [5] use a voxel size of  $v_D = 0.5 * v_H = 0.4 * v_W = 28$  m. Figure 8 shows the 3D visualization of our detector on point cloud data.

Table 1 and Table 2 respectively present the 3D detection results using datasets from clear weather and adverse weather conditions as the training data. The metrics for 'Tour Boat' are noticeably lower than those for 'Cargo Ship'. The poor performance of 'Tour Boat' is attributed to two factors. First, as shown in Figure 1a,b, the target labels in the dataset are defined based on their use. Despite all being labeled as 'Tour Boat', there is a significant difference in appearance among these boats. Second, there are fewer samples, with only about 22.6% of the ships in the training set being tour boats. In contrast, 'Cargo Ship' makes up 73.9% of the data, and their appearances are generally similar, resulting in relatively higher performance metrics. Therefore, a diverse set of samples helps the model learn better.

The two tables demonstrate that the Ship Detector excels in detecting the 'Cargo Ship' category across both types of training datasets. Ships, significantly larger than vehicles, particularly in height, pose unique challenges in detection. PointPillar, which performs suboptimally with both datasets, divides space into pillars to extract point cloud features, consequently losing crucial height information. In contrast, the Ship Detector employs voxel-based convolution to segment space into finer grids. Despite increasing the voxel dimensions to match the ship sizes, this approach reduces spatial resolution. To overcome

this limitation, the Ship Detector utilizes a dual-branch strategy to extract 3D features, thus broadening the receptive field. During 2D CNN convolutions, it retains 3D features for use in the detection head, ensuring that 3D information is preserved throughout the target localization process. The Ship Detector and CenterPoint show improved performance when trained on the adverse weather dataset compared to the clear weather dataset. The data used to build the validation set are collected from the real world, and due to the inherent noise in the sensors, the collected data will include noise in the scenes. Simulating foggy and rainy conditions enhances data diversity, helping the detector differentiate between foreground and background information. Data diversity does not always improve model performance. If diversity introduces too much irrelevant or noisy data, it can actually impair the detector’s ability to effectively learn the most critical features, as occurs with Second, PointRcnn, and PointPillar.



**Figure 8.** Visualization of ship detection results using our network: black cubes represent the ground truth 3D boxes, while green cubes indicate the detection results.

**Table 1.** Comparison of 3D detection performance trained on clear weather conditions datasets. The detection categories are ‘Cargo Ship’ and ‘Tour Boat’.

Method	Cargo Ship			Tour Boat		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Second	68.31	67.60	65.37	17.06	15.76	15.70
PointRcnn	62.29	61.63	60.22	22.18	18.26	18.21
PointPillar	31.09	30.87	30.74	18.18	16.69	16.65
CenterPoint	52.11	51.16	51.01	20.12	19.74	19.70
Ship Detector	73.4	71.01	68.67	30.89	27.91	27.91

**Table 2.** Comparison of 3D detection performance trained on adverse weather conditions datasets. The detection categories are ‘Cargo Ship’ and ‘Tour Boat’.

Method	Cargo Ship			Tour Boat		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Second	62.65	60.19	59.74	17.94	14.85	14.85
PointRcnn	60.39	58.27	58.65	22.58	16.88	16.55
PointPillar	31.02	29.55	29.18	29.15	23.02	22.01
CenterPoint	53.11	52.34	52.11	19.02	19.09	19.07
Ship Detector	76.27	73.71	73.37	23.11	19.09	19.07

### 5.5. Ablation Studies

In this section, we perform thorough ablation studies to assess how each component contributes to our detector’s overall performance.

#### 5.5.1. Effects of Dimensional Sparse Convolution with Different Branches

To determine the effects of 3D sparse convolutions across different scales, a comparison is made between the single-branch sparse convolution and the dual-branch sparse convolution employed in our detection system. The single-branch approach involves multiple downsampling stages of 3D sparse convolution, with each downsampling operation reducing the resolution to half of its previous state. Conversely, the dual-branch approach combines multiple downsampling stages of 3D sparse convolution with an additional branch that downsamples to one-eighth of the original resolution. To ensure a fair comparison, subsequent layers in both architectures employ a multi-scale 2D CNN. The results, as detailed in Table 3, demonstrate that our dual-branch configuration significantly outperforms the single-branch approach. The single-branch model tends to focus on either local features or global context exclusively. In contrast, the multi-branch strategy allows the model to concurrently consider both local details and global context, which is crucial for understanding complex scenes and effectively processing objects of varying sizes.

**Table 3.** Comparison of single-branch with dual-branches. Testing 3D detection results on an adverse weather dataset. The detection categories are ‘Cargo Ship’ and ‘Tour Boat’.

Method	Cargo Ship			Tour Boat		
	Easy	Moderate	Hard	Easy	Moderate	Hard
single-branch	62.65	60.19	59.74	17.94	14.85	14.85
dual-branches	67.84	63.54	63.08	19.84	15.76	15.76

#### 5.5.2. Effects of Features Maps

To validate the impact of directly using feature maps for prediction, as discussed in Section 4.3, we design an experiment employing the network structure from Second and the multi-scale 2D CNN module from this study. For fairness, a single-branch 3D Feature Net is used in the 3D sparse convolution stage for extracting 3D sparse features from the point cloud. The results, as indicated in Table 4, clearly demonstrate that incorporating Feature Maps significantly outperforms the network configuration without Feature Maps. The integration of 3D original features with encoded features in the network enhances feature diversity. This diversity aids the model in capturing more subtle and intricate geometric features, which might be lost during the encoding process.

**Table 4.** The experimental results compared with and without the inclusion of feature maps. The detection categories are ‘Cargo Ship’ and ‘Tour Boat’.

Method	Cargo Ship			Tour Boat		
	Easy	Moderate	Hard	Easy	Moderate	Hard
without feature maps	62.65	60.19	59.74	17.94	14.85	14.85
with feature maps	76.48	74.13	73.69	22.07	17.69	17.66

## 6. Discussion

Collecting LiDAR data in adverse weather significantly impacts sensor performance, reducing target visibility and accuracy due to light scattering and occlusion, thereby affecting object detection capabilities. Considering the substantial expense associated with data gathering and labeling, physically simulating clear weather data into adverse conditions emerges as a viable solution to the scarcity of data. This work enriches training data by simulating adverse weather point cloud data from existing clear weather data through physical simulation methods. The particles in fog scatter and absorb the light from LiDAR sensors, reducing the sensors’ ability to penetrate fog, leading to decreased visibility and accuracy. The shipborne LiDAR sensors can scan up to 1 nautical mile, with soft targets present throughout the scene. Unlike fog, LiDAR beams can penetrate raindrops. Therefore, rainy scenes do not produce many soft targets, but the refraction and scattering of raindrops reduce the number of point cloud samples.

To the best of our knowledge, we introduce pioneering work on ship detection in marine scenarios, utilizing a dual-branch sparse network for initial feature extraction, followed by a multi-scale 2D CNN module to preserve information across varying heights. Car and ship detection differ not only in dimensions but also in target shapes. Car targets are relatively regular, while ships can vary greatly in appearance due to their purpose or cargo state, as demonstrated in Figures 1b with sightseeing boats that differ significantly in design. This diversity leads to low detection metrics for ‘Tour Boat’ in Tables 1 and 2, indicating the need for extensive data training. Detectors usually segment the entire point cloud scene into voxels for feature extraction via single-channel conventional convolution. Our detector uses dual-channel convolution, extracting features at multiple scales to provide a diverse receptive field. After multiple extractions, features of different scales are sparsely concatenated. As a standard for small object detection algorithms, features are compressed into feature maps for 2D CNN convolution. Table 2 shows dual-branches conventional convolution performs better than single-branch. 2D CNNs refine features extracted by initial layers, processing spatial information crucial for distinguishing objects and their orientations or states, thereby enhancing task accuracy like object detection. Small target detection might employ one or two encoding-decoding cycles. Encoders generate high-dimensional feature maps by capturing spatial hierarchies, while decoders upsample these maps, enhancing detection by restoring details lost during encoding. However, information loss occurs during this process. This paper introduces an additional channel to preserve original feature maps, combining all features through stacking for subsequent processing. Table 3 highlights the advantages of preserving original feature maps.

In the marine point clouds dataset, the range of XYZ axes is  $[-1800 \text{ m}, 1800 \text{ m}] * [0 \text{ m}, 1800 \text{ m}] * [-20 \text{ m}, 50 \text{ m}]$ , but the preset range for input detection is  $[-320 \text{ m}, 480 \text{ m}] * [0 \text{ m}, 320 \text{ m}] * [-11 \text{ m}, 17 \text{ m}]$ . This is due to GPU limitations, and this is already the maximum detection range. Our detector utilizes voxel-based convolution for feature extraction. When processing larger input ranges, it is necessary to increase the voxel size, which brings adverse effects such as reducing the spatial resolution of the point cloud, losing important fine structural information, causing geometric distortion of object edges and shapes, and making it difficult for the model to distinguish between multiple small objects that are closely packed together. Additionally, due to the sparsity of the point cloud, targets collected from distant ships typically only consist of a few dozen points, and most marine areas are devoid of points. This requires the detector to have both a large receptive

field to handle expansive scenes and high resolution to capture detailed information, which is a significant challenge under limited GPU resources. Therefore, it is necessary to develop feature extraction methods that are better suited for expansive scenes.

## 7. Conclusions

The work enhances training data diversity by simulating clear weather data into adverse weather conditions using physical simulation methods. We introduce an innovative Ship Detector designed for marine 3D scenes. 3D point cloud data offers depth information and is not influenced by lighting conditions, presenting significant advantages over image detection. The object detectors are designed for small terrestrial targets; our detector fills the gap in marine scene 3D detection. Our detector, like those for small targets, segments scenes into voxels and uses conventional convolution and CNN convolution for feature extraction. To retain more height and detail information, we employ a dual-branch sparse convolution layer for conventional convolution, with multi-scale convolution maintaining a larger receptive field. Additionally, we use a multi-scale 2D CNN module that extracts features by both encoding and decoding methods and preserves feature map characteristics to address the loss of detailed information during the encoding phase.

**Author Contributions:** Author Contributions: Conceptualization, Q.Z.; methodology, Q.Z. and L.W.; software, Q.Z. and Z.Z.; formal analysis L.W. and H.M.; investigation, L.W. and C.Y.; resources L.W. and H.M.; data curation, Q.Z. and L.W.; writing—original draft preparation, Q.Z. and L.W.; writing—review and editing Q.Z. and L.W.; visualization Q.Z.; supervision, Z.Z. and C.Y.; funding acquisition, Z.Z. and H.M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (No. 62173103) and the Fundamental Research Funds for the Central Universities of China (No. 3072022JC0402, No. 3072022JC0403).

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Benedek, C. 3D people surveillance on range data sequences of a rotating Lidar. *Pattern Recognit. Lett.* **2014**, *50*, 149–158. [[CrossRef](#)]
2. Yan, Y.; Mao, Y.; Li, B. Second: Sparsely embedded convolutional detection. *Sensors* **2018**, *18*, 3337. [[CrossRef](#)]
3. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. *arXiv* **2019**, arXiv:1812.04244.
4. Yin, T.; Zhou, X.; Krahenbuhl, P. Center-based 3d object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 11784–11793.
5. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. Pointpillars: Fast encoders for object detection from point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12697–12705.
6. Bijelic, M.; Gruber, T.; Mannan, F.; Kraus, F.; Ritter, W.; Dietmayer, K.; Heide, F. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11682–11692.
7. Pitropov, M.; Garcia, D.E.; Rebello, J.; Smart, M.; Wang, C.; Czarnecki, K.; Waslander, S. Canadian adverse driving conditions dataset. *Int. J. Robot. Res.* **2021**, *40*, 681–690. [[CrossRef](#)]
8. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The kitti vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.
9. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. Nuscen: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11621–11631.
10. Sun, P.; Kretschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2446–2454.
11. Patil, A.; Malla, S.; Gang, H.; Chen, Y.T. The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9552–9557.

12. Goodin, C.; Carruth, D.; Doude, M.; Hudson, C. Predicting the Influence of Rain on LIDAR in ADAS. *Electronics* **2019**, *8*, 89. [[CrossRef](#)]
13. Kilic, V.; Hegde, D.; Sindagi, V.; Cooper, A.B.; Foster, M.A.; Patel, V.M. Lidar light scattering augmentation (lisa): Physics-based simulation of adverse weather conditions for 3d object detection. *arXiv* **2021**, arXiv:2107.07004.
14. Hahner, M.; Sakaridis, C.; Dai, D.; Van Gool, L. Fog simulation on real LiDAR point clouds for 3D object detection in adverse weather. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 15283–15292.
15. Hahner, M.; Sakaridis, C.; Bijelic, M.; Heide, F.; Yu, F.; Dai, D.; Van Gool, L. LiDAR snowfall simulation for robust 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16364–16374.
16. Zhang, W.; Wu, Y.; Tian, X.; Bao, W.; Yu, T.; Yang, J. Application Research of ship overload identification algorithm based on lidar point cloud. In Proceedings of the 2022 2nd International Conference on Electrical Engineering and Mechatronics Technology (ICEEMT), Hangzhou, China, 1–3 July 2022; pp. 377–381.
17. Lu, X.; Li, Y.; Xie, M. Preliminary study for motion pose of inshore ships based on point cloud: Estimation of ship berthing angle. *Measurement* **2023**, *214*, 112836. [[CrossRef](#)]
18. Kuang, H.; Wang, B.; An, J.; Zhang, M.; Zhang, Z. Voxel-FPN: Multi-Scale Voxel Feature Aggregation for 3D Object Detection from LIDAR Point Clouds. *Sensors* **2020**, *20*, 704. [[CrossRef](#)] [[PubMed](#)]
19. Zhan, J.; Liu, T.; Li, R.; Zhang, J.; Zhang, Z.; Chen, Y. Real-Aug: Realistic Scene Synthesis for LiDAR Augmentation in 3D Object Detection. *arXiv* **2023**, arXiv:2305.12853.
20. Huang, J.; Zhu, P.; Geng, M.; Ran, J.; Zhou, X.; Xing, C.; Wan, P.; Ji, X. Range scaling global u-net for perceptual image enhancement on mobile devices. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
21. Guo, C.; Li, C.; Guo, J.; Loy, C.C.; Hou, J.; Kwong, S.; Cong, R. Zero-reference deep curve estimation for low-light image enhancement. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1780–1789.
22. Tassano, M.; Delon, J.; Veit, T. Fastdvdnet: Towards real-time deep video denoising without flow estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 1354–1363.
23. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [[CrossRef](#)]
24. Li, C.L.; Zaheer, M.; Zhang, Y.; Poczos, B.; Salakhutdinov, R. Point cloud gan. *arXiv* **2018**, arXiv:1810.05795.
25. Shu, D.W.; Park, S.W.; Kwon, J. 3d point cloud generative adversarial network based on tree structured graph convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 3859–3868.
26. Wang, B.; Lan, J.; Gao, J. MSG-Point-GAN: Multi-Scale Gradient Point GAN for Point Cloud Generation. *Symmetry* **2023**, *15*, 730. [[CrossRef](#)]
27. Yang, G.; Huang, X.; Hao, Z.; Liu, M.Y.; Belongie, S.; Hariharan, B. Pointflow: 3d point cloud generation with continuous normalizing flows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 4541–4550.
28. Chen, Y.; Hu, V.T.; Gavves, E.; Mensink, T.; Mettes, P.; Yang, P.; Snoek, C.G. Pointmixup: Augmentation for point clouds. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part III 16; Springer: Cham, Switzerland, 2020; pp. 330–345.
29. Xiao, A.; Huang, J.; Guan, D.; Cui, K.; Lu, S.; Shao, L. Polarmix: A general data augmentation technique for lidar point clouds. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 11035–11048.
30. Mehra, A.; Mandal, M.; Narang, P.; Chamola, V. ReViewNet: A fast and resource optimized network for enabling safe autonomous driving in hazy weather conditions. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4256–4266. [[CrossRef](#)]
31. Rasshofer, R.H.; Spies, M.; Spies, H. Influences of weather phenomena on automotive laser radar systems. *Adv. Radio Sci.* **2011**, *9*, 49–60. [[CrossRef](#)]
32. Carballo, A.; Lambert, J.; Monrroy, A.; Wong, D.; Narksri, P.; Kitsukawa, Y.; Takeuchi, E.; Kato, S.; Takeda, K. LIBRE: The multiple 3D LiDAR dataset. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium (IV), IEEE, Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1094–1101.
33. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
34. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *arXiv* **2017**, arXiv:1706.02413.
35. Zhou, Y.; Tuzel, O. Voxelnet: End-to-end learning for point cloud based 3d object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4490–4499.
36. Chen, Y.; Liu, J.; Zhang, X.; Qi, X.; Jia, J. Voxelnext: Fully sparse voxelnet for 3d object detection and tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 21674–21683.

37. Misra, I.; Girdhar, R.; Joulin, A. An end-to-end transformer model for 3d object detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 2906–2917.
38. Erabati, G.K.; Araujo, H. Li3DeTr: A LiDAR based 3D Detection Transformer. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 4250–4259.
39. Lewandowski, P.A.; Eichinger, W.E.; Kruger, A.; Krajewski, W.F. Lidar-based estimation of small-scale rainfall: Empirical evidence. *J. Atmos. Ocean. Technol.* **2009**, *26*, 656–664. [[CrossRef](#)]
40. Filgueira, A.; González-Jorge, H.; Lagüela, S.; Díaz-Vilariño, L.; Arias, P. Quantifying the influence of rain in LiDAR performance. *Measurement* **2017**, *95*, 143–148. [[CrossRef](#)]
41. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1137–1149. [[CrossRef](#)]
42. Zheng, W.; Tang, W.; Chen, S.; Jiang, L.; Fu, C. CIA-SSD: Confident IoU-Aware Single-Stage Object Detector From Point Cloud. *arXiv* **2020**, arXiv:2012.03015.
43. Wu, H.; Wen, C.; Shi, S.; Li, X.; Wang, C. Virtual sparse convolution for multimodal 3d object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 21653–21662.
44. Wu, H.; Wen, C.; Li, W.; Li, X.; Yang, R.; Wang, C. Transformation-equivariant 3d object detection for autonomous driving. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 2795–2802. [[CrossRef](#)]

**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.