

# Denoising Framework Based on Multi-frame Continuous Point Clouds for Autonomous Driving LiDAR in Snowy Weather

Xinyuan Yan, Junxing Yang, Xinyu Zhu, Yu Liang, and He Huang

**Abstract**— Adverse weather conditions are one of the long-tailed problems facing autonomous driving. Solving the problem of autonomous driving operation in adverse weather conditions is an important challenge for realizing advanced autonomous driving. In order to enhance the LiDAR perception capability in snowy weather for autonomous driving, this study proposes a denoising method for multi-frame continuous point clouds. The core concept of this method is to allow ordered objects (e.g., stationary objects on the ground) to strengthen each other, while allowing disordered objects (e.g., snow) to weaken each other. This is done by first selecting three consecutive frames of the point cloud as a denoising unit, and then removing the ground points from each frame of the point cloud. After that, the point clouds from the first two frames are used as the source point clouds, and the point cloud from the third frame is used as the target point cloud for point cloud registration. Finally, the Time Outlier Removal (TOR) filter proposed in this paper combined with Entropy Weight Method (EWM) is utilized for denoising. The experimental results show that the performance of the method proposed in this paper exceeds the existing methods. In addition, the method in this paper not only removes the disordered snowflakes in the air, but also removes some other disordered noise points (e.g., the ghosting of the stationary objects), which provides an advantageous guarantee for the realization of the automatic driving in snowy weather. The source code is available at <https://github.com/Naclzno/TOR>.

**Index Terms**— Autonomous driving, LiDAR, snowy weather, point cloud denoising, disorder.

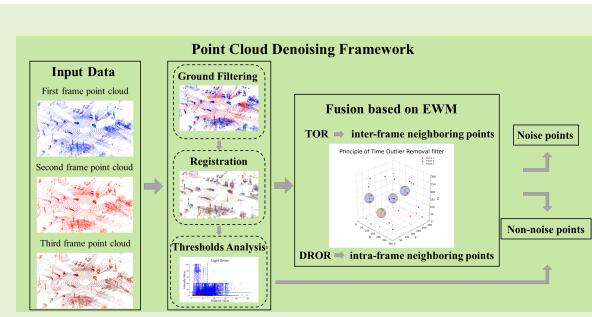
## I. INTRODUCTION

In recent years, autonomous driving technology has advanced remarkably, particularly in sunny weather conditions, and it is getting closer and closer to people's lives. However, autonomous driving technology in adverse weather faces great challenges, mainly because adverse weather greatly weakens the perception ability of autonomous driving. For example, in snowy weather, snowflakes can form noise points in the point cloud scanned by the LiDAR mounted on the autonomous vehicle, affecting the vehicle's perception of the surrounding environment.

In order to extend the use-cases of autonomous driving technology in snowy weather, researchers have proposed many different denoising methods for snow noise points in LiDAR point clouds, such as denoising by constructing Conditional

This research was funded by the National Natural Science Foundation of China (Grant Numbers 42201483), the China Postdoctoral Science Foundation (Grant Numbers 2022M710332) and Funding for Postdoctoral Research Activities in Beijing(2023-zz-140). The associate editor coordinating the review of this article and approving it for publication was xxx. (Corresponding author: He Huang.)

Xinyuan Yan, Junxing Yang, Xinyu Zhu, Yu Liang and He Huang are with the School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, Beijing 102616, China (e-mail: yan1075783878@126.com; yangjunxing@bucea.edu.cn; zxy155100@gmail.com; r98646084@gmail.com; huanghe@bucea.edu.cn).



Random Fields (CRF) using the physical properties of snow to derive confidence levels for snow [1]. These denoising methods are primarily categorized into two classes: traditional filter methods and deep learning methods. The most classical filters are ROR and SOR filters [2]. Both of these filters are designed based on the a prior knowledge that noise points are isolated. The ROR filter mainly searches for the number of neighbors of each point in the point cloud within a specified search radius, and identifies a point as noise if the number of neighbors is less than a specified value. The SOR filter iterates over each point in the point cloud and calculates the mean distances to its K nearest neighbors, then calculates the mean and standard deviation of the distances to all the points, and finally calculates the global threshold, and identifies the points whose mean of the distances to the K nearest neighbors is greater than the threshold as noise points. However, considering only the a prior knowledge that the noise points are isolated will lose many environmental features, so the DROR and DSOR filters consider the a prior knowledge that the point cloud collected by LiDAR has the characteristic of near-dense and far-sparse [3], [4]. DROR improves the search radius from a fixed radius set by the ROR to a dynamically set search radius, and DSOR improves the search radius from a fixed global threshold set by SOR to a dynamically set threshold. Experimental results show

that they can remove most of the noise points as well as retain most of the environmental features. Another filter LIOR is designed based on the prior knowledge that the LiDAR intensity of snow noise points is below a certain threshold [5]. The use of deep learning to remove snow noise points from LiDAR point clouds is not yet mature enough, mainly because snowy weather datasets are not rich enough and it is too complicated to manually label snow noise points point by point. 4DenoiseNet is a recently proposed neural network model for removing snow noise points, which was trained and evaluated on SnowyKITTI, a dataset that simulates real-life snowfall [6]. It can be trained well on SnowyKITTI, removing most of the snow noise points. But the dataset is changed, its denoising ability will weaken, so its generalization ability is not as strong as traditional methods.

In this paper, we design a snow denoising framework for LiDAR of autonomous driving applications, which is based on the prior knowledge that snow noise points have disorder. The motivation of our research is that we want to strengthen the connections between ordered objects(e.g., stationary objects on the ground) and weaken the connections between disordered objects(e.g., snow). To this end, we break the traditional single-frame point cloud denoising idea and introduce the multi-frame continuous point cloud denoising idea. The specific approach is done by first selecting three consecutive frames of the point cloud as a denoising unit, and then removing the ground points from each frame of the point cloud. After that, using the point clouds in the first two frames as the source point clouds and the point cloud in the third frame as the target point cloud for point clouds registration. Finally, the Time Outlier Removal (TOR) filter proposed in this paper combined with Entropy Weight Method (EWM) is utilized for denoising [7]. The contributions of this paper are as follows:

- 1) We introduce the prior knowledge that snow noise points have disorder, and propose a framework for denoising multi-frame continuous point clouds in snowy weather for autonomous driving LiDAR. In the denoising framework, we use ground filtering, point cloud registration, thresholds analysis, and EWM.
- 2) In the denoising framework, we design TOR filter based on the disorder of noise points.
- 3) In order to retain more dynamic non-snow point objects, we fused three metrics on neighboring points based on the EWM approach, which allows our method to remove not only dynamic airborne snow points, but also disordered noise points such as moving vehicles and their trailing shadow, and ghosting of stationary objects on the ground.

## II. RELATED WORK

### A. A Generalized Study of Point Cloud Denoising

Past research has explored a variety of methods for 3D point cloud denoising, which can usually be categorized into the following five classes [8]: (i) Statistical-based denoising methods [9], [10], [11], such as point cloud denoising based on principal component analysis [9] or Bayesian estimation [11];

(ii) Neighborhood-based denoising methods [12], [13], which can be achieved by using the neighborhood information of each point to compute similarity between the points to achieve point cloud denoising, bilateral filtering [12] is a representative method; (iii) projection-based denoising methods [14], which can be interpreted as projecting the point cloud in multiple views or using different projection strategies to realize point cloud denoising; (iv) voxel-based denoising methods [15], which rasterize the point cloud and denoise the points within the voxels, and then used them as the basic units; (v) other denoising methods [16], [17], [18], [19], including signal processing-based denoising methods [16], partial differential equation-based denoising methods [17], and denoising methods that integrate multiple methods [18].

However, with the development of deep learning techniques, some studies have begun to explore the use of deep learning techniques for point cloud denoising. For example, WeatherNet [20], published in 2020, provides a convolutional neural network (CNN)-based point cloud denoising method, which is specifically designed to deal with noise in LiDAR point cloud data induced by severe weather (e.g., fog, rain, etc.). In fact, WeatherNet is essentially an efficient variant of LiLaNet [21], a point-by-point, multi-class semantic labeling neural network for semi-dense LiDAR data presented at ECCV in 2018.

### B. Point Cloud Denoising Study in Snowy Weather

Currently, LiDAR denoising methods in snowy weather are generally categorized into two main types, one is the traditional use of filters for denoising, and the other is the use of deep learning methods for denoising. Traditional filters are generally based on a single-frame point cloud for denoising, which can only utilize spatial information and cannot utilize temporal information well. Therefore we want to design a filter that can utilize both spatial and temporal information. As for the methods of removing snow noise points in point clouds using deep learning is only recently proposed and is not mature enough. There are two main reasons: first, the publicly available adverse weather datasets are still immature. Specifically, the point cloud data in snowy conditions is not sufficiently rich, and manual annotation of noise is particularly challenging. The second is that snowfall varies from place to place, which leads to the fact that the denoising network trained on one dataset cannot show good denoising ability on other datasets. The following sections will explain the mainstream methods of removing snow noise points by filters and deep learning in detail.

The traditional generalized PCL-based filters are ROR filter and SOR filter [2]. They are designed based on the prior knowledge that noise points are isolated. However, they are unable to retain environmental feature points at medium and long distances. After that, considering the property that the density of point cloud data collected by LiDAR is higher near the sensor and becomes sparser with increasing distance from the sensor, the researchers have advanced the original ROR and SOR filters to create their enhanced counterparts, termed DROR [3] and DSOR [4] filters, respectively. These two filters are able to retain the environmental feature points

at medium and long distances better. But they only consider the distance information and not the intensity information. Therefore, researchers proposed the LIOR filter based on the a priori knowledge that the intensity value of snow noise points is below a certain threshold [5]. But essentially LIOR filter only considers intensity information on the basis of ROR filter. For this reason, researchers proposed the DDIOR filter by combining the LIOR filter with the DSOR filter [24]; and the DIOR filter by combining the LIOR filter with the DROR filter [25]. We proposed the LIDSOR filter based on DSOR filter in a previous paper [26] based on the intensity and spatial characteristics of snow noise points. In addition, in order to improve the denoising efficiency, researchers proposed PCAAR filter by combining principal component analysis technique with DROR filter [22]; FCSOR filter by combining voxel subsampling method and SOR filter [23]. The principles of the ROR, SOR, DROR, DSOR, and LIOR filters are as described in the introduction, so we will not elaborate them further. The principles of each of the other filters are detailed below.

The PCAAR filter is proposed by a team in 2021, which is actually a dimensionality reduction of the input original three-dimensional point cloud data using the PCA technique, extracting the two principal components while discarding the eigenvector corresponding to the smallest eigenvalue, and projecting the three-dimensional point cloud data into a two-dimensional space, which reduces the computational complexity. Then the AROR filter is utilized to perform adaptive radius outlier removal on the downscaled 2D point cloud data. Finally the filtered first principal component and second principal component are then recovered as 3D point cloud data. AROR filter is actually essentially DROR filter.

The DDIOR filter is proposed by a team in 2022, which for the first time utilizes distance threshold and intensity threshold to preprocess the data of the input raw point cloud, retaining data above the threshold and performing subsequent dynamic filtering on data below the threshold. It is essentially an improvement based on the DSOR filter, which integrates dynamic distance coefficient and dynamic intensity coefficient in the process of setting dynamic threshold.

We proposed a new filtering technique for ISPRS called LIDSOR in 2023 [24]. It is also essentially an improvement based on the DSOR filter. In the article, we analyze the intensity characteristics and spatial distribution characteristics of snow noise points and fit a gamma distribution curve to characterize the spatial distribution of snow noise points. Finally, a LIDSOR filter is designed based on the intensity threshold and distance threshold.

Unlike various filters, there are not many methods to remove snow noise points using deep learning. 4DenoiseNet is a deep learning algorithm released in 2023 for removing snow noise points [6]. It is actually improved from the point cloud semantic segmentation neural network [25]. Unlike other methods, this algorithm utilizes the time dimension. It then uses the LiDAR snowfall simulation algorithm [26] presented at CVPR in 2022 to create the SnowyKITTI dataset on which 4DenoiseNet is trained. Although the snowfall simulation algorithm can automatically generate the labels of the points and avoid the tediousness of manually labeling the dataset, there is still a gap between the simulated dataset and the real dataset. How to quickly label the real dataset is a challenge faced by the current method of removing snow noise points using supervised learning. There are also unsupervised, self-supervised denoising methods that do not require labeling. For example, LiSnowNet is an unsupervised denoising algorithm based on deep convolutional neural networks released on IROS in 2022, specifically designed to handle LiDAR point cloud data damaged by snow [27]. SLIDE is a self-supervised learning framework released at ECCV in 2022, specifically designed to remove snow noise points from LiDAR point clouds [28]. However, all of these methods can only be adapted to specific snowfall scenarios, and the generalization ability is not strong. In contrast, the denoising method proposed in this paper does not require labeled data for training, and its generalization ability far surpasses these methods.

### III. METHODOLOGY

Our innovation is inspired by prior knowledge that objects are ordered in multi-frame continuous point clouds, while snow is disordered. For this reason, we break the idea of

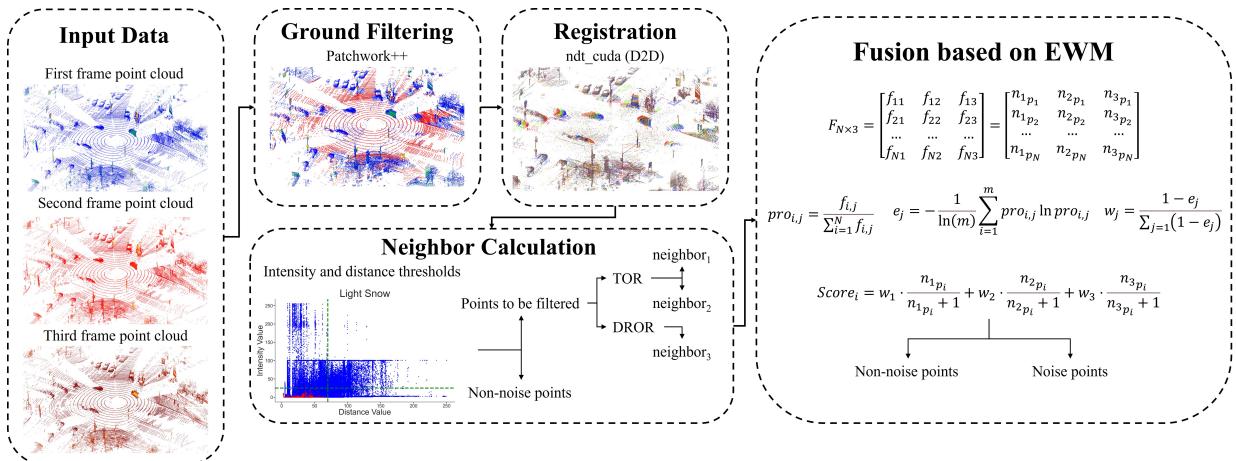


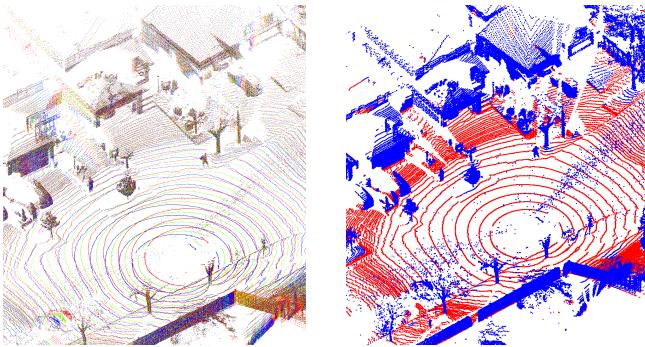
Fig. 1. Point cloud denoising framework.

single-frame point cloud denoising of traditional methods and introduce the idea of multi-frame point cloud denoising. In this paper, we choose three consecutive frames of point clouds as a denoising unit. In each denoising unit, the last frame point cloud has the greatest influence on the current perception of the autonomous vehicle, so we choose the last frame point cloud for denoising.

The point cloud denoising framework of this paper is shown in Figure 1. The input point cloud data are organized into multiple denoising units, and each unit includes three consecutive frames of point clouds. First, ground filtering is performed on each frame of the point cloud in every denoising unit to remove ground points, followed by the registration process for these three frames of point clouds. The registered point cloud is initially classified according to the intensity threshold and distance threshold. Subsequently, the inter-frame neighboring points of each point in the third frame point cloud within a specified search radius are calculated using the TOR filter proposed in this paper. In order to maximize the retention of dynamic objects while removing snow noise points, a modified DROR filter will be used to compute the intra-frame neighboring points of each point in the third frame point cloud within the dynamic search radius. Finally, the three kinds of neighboring points are objectively weighted and combined to determine whether they are snow noise points or not.

### A. Ground Filtering

During the experiment, we observed that without applying the ground filtering algorithm to each frame of the point cloud, the registration result, as depicted in Figure 2 (a), is suboptimal. This figure clearly shows that the ground points are not effectively registered, which in turn adversely affects the subsequent classification of snow noise points. Owing to these observations, we have selected the Patchwork++ ground filtering algorithm [29], introduced in 2022, for our study. Figure 2 (b) illustrates the result of a single-frame point cloud after applying this ground filtering.



(a) Reason for ground filtering (red, green and blue represent point clouds from different frames)  
 (b) Result of ground filtering (red represents ground points, blue represents non-ground points)

Fig. 2. Ground filtering.

### B. Point Clouds Registration

In the denoising framework proposed in this paper, the point clouds registration algorithm [30] is one of the key algorithms to realize our idea. When three frames of point clouds are simply superimposed together, it does not realize the original idea: ordered objects strengthen the connection with each other and disordered objects weaken the connection with each other. Figure 3 shows the result of simply superimposing the three-frame point clouds. From the figure it can be noticed that there is an overall offset between the point clouds of different frames.

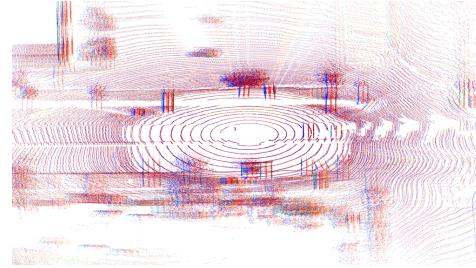


Fig. 3. Simply superimposed point clouds (blue, red, and brown represent point clouds from different frames).

To realize the initial idea, the CUDA-accelerated D2D-NDT algorithm from the VGICP paper published in 2021 is chosen for point clouds registration in this paper [31]. Figure 4 shows the registration results of the point clouds after removing the ground points. In Figure 4, we denote the point clouds from different frames with red, green, and blue colors, respectively. Ordered points refer to the points that make up an ordered object, such as the buildings around the road in the figure. Their points in different frames are superimposed together after point cloud registration and appear in brown color. Disordered points are points that make up an unordered object, such as the snow points in the middle of the road in the figure. Their points in different frames cannot be registered together due to their own disorder and appear in one of three colors: red, green, or blue. This approach fulfills the initial concept of this paper and establishes a solid basis for the filters that are designed later.

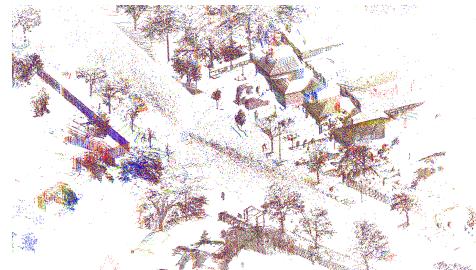


Fig. 4. Result of point clouds registration.

### C. TOR Filter

After ground filtering, only non-ground points are retained in each frame of the point cloud, and then three consecutive frames of the point clouds are registered, using the last frame

as the target, to obtain the registered point cloud, which will be the initial point cloud input to the TOR filter. The principle of the TOR filter design is based on the fact that stationary objects above the ground are capable of being registered, but dynamic objects cannot be registered such as snowflakes. Therefore, in this paper, the points in the third frame of the registered point cloud are used as the reference points for radius search in this paper. Within the specified search radius, the number of neighbors from the other two frames of point clouds are queried, and if it is less than a certain number, it is judged as a noise point, otherwise it is a non-noise point. The principle is shown in Figure 5, where the red sphere represents that the reference point is a noise point, and the blue sphere represents that the reference point is a non-noise point (at this time, the number of neighbors of the other two frames is set to 1). The pseudo-code is shown in Algorithm 1.

#### Algorithm 1 Time Outlier Removal (TOR) Filter

##### Input:

Point cloud after ground filtering and registration  $P = p_i$ ,  $i = 1, 2, \dots, N$ ;  
 $p_i = (x_i, y_i, z_i, intensity_i, label_i, time_i)$   
Minimum number of neighbors  $n$   
Search radius  $r$   
Distance threshold  $dt$   
Intensity threshold  $it$

##### Output:

Noise points  $ep$   
Non-noise points  $en$

```

for  $p_i \in P$  do
    Calculate:  $distance_i = \sqrt{x_i^2 + y_i^2 + z_i^2}$ 
    if  $distance_i < dt$  and  $intensity_i < it$  then
        Classified as the initial filtering result  $O$ 
    else if  $time_i == 3$  then
        Classified as non-noise points  $en$ 
    end if
end for

for  $p_i \in O$  do
     $O$  is set to kd-tree data structure
    for  $time_i == 3$  do
        neighbors = radiusSearch( $p_i, r$ )
        if  $neighbor.time == 1$  then
             $+ + neighbor_1$ 
        else if  $neighbor.time == 2$  then
             $+ + neighbor_2$ 
        end if
        if  $neighbor_1 \geq n$  and  $neighbor_2 \geq n$  then
            Classified as non-noise points  $en$ 
        else
            Classified as noise points  $ep$ 
        end if
    end for
end for

```

The input point cloud in the algorithm has six fields ( $x$ ,  $y$ ,  $z$ , intensity, label, time) for each point. The first three

Principle of Time Outlier Removal filter

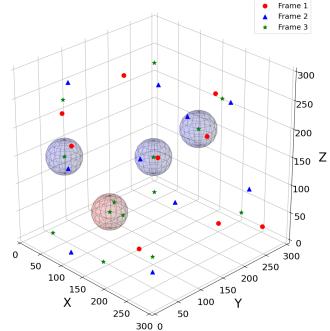


Fig. 5. TOR filter principle.

fields represent the spatial coordinates of the point relative to the origin, while the intensity field represents the laser intensity. The label field is a manual annotation field used to mark whether a point cloud is noise. In this field, non-noise points is set to 0, while the noise points is set to 1. This setting facilitates the intensity-distance characterization of the noise points, as well as the quantitative evaluation of the classification results of the filters. The time field is used to label the frames from which the points originate during the registration process of the point clouds. 1 for the first frame, and 2 and 3 for the second and third frames, respectively.

The TOR filter first uses a distance threshold and an intensity threshold in Algorithm 1 to classify the input raw point cloud into two classes: the initial filtering result and non-noise points. This is because after the analysis of distance and intensity characterization of the snow noise points, it is known that the snow noise points are distributed within a certain range, and if it exceeds this range there are no noise points caused by snowflakes, so there is no need to filter all the points [32]. This also helps to reduce the amount of computation and increase the efficiency of the algorithm.

After that the TOR filter uses a fixed search radius to query the number of neighbors in the other two frames. However, according to previous studies, it is known that the density of snow noise points is higher near the sensor and becomes sparser with increasing distance from the sensor, so it is reasonable to use a dynamic search radius to retain more environmental feature points. However, the point clouds were previously registered, which led to a change in the characteristics of the point clouds, and the stationary objects were well registered to each other, so there is no need to set the search radius dynamically, and in addition, this dynamic setting of the search radius will increase the computational complexity of the filter.

#### D. EWM-based Fusion

According to the design principle of TOR filter, we inevitably face a problem: when the autonomous vehicle is traveling in snowy weather, the dynamic objects include not only snowflakes, but also other moving vehicles and so on. But actually there is an essential difference between dynamic snowflakes and dynamic vehicles: snowflakes are isolated

points, while vehicles points are aggregated. So in order to retain more dynamic non-snow objects, we first apply the TOR filter to the points in the third frame of the registered point cloud, and get the neighbors for the other two frames, respectively ( $neighbor_1$  and  $neighbor_2$ ). After that, we improve the DROR filter, and search for neighbors on the third frame of the point cloud separately, and get the neighbors for the third frame ( $neighbor_3$ ). In this way, we get the neighbors of the third frame. Regarding the neighbors we get three metrics, the first two metrics serve to measure the degree of disorder of the points between different frames, while the last metric serves to measure the degree of isolation of the points between single frames.

We are inspired by the work of Boyang Li et al. who used the Entropy Weight Method (EWM) to fuse two metrics, 3D spatial neighboring points and W-T spatial neighboring points, to compute confidence scores as a way of determining whether points in a point cloud are snow noise points [7]. Therefore, this paper introduces the entropy weight method (EWM) to calculate the weights of the three neighboring metrics. Entropy weight method is a weight calculation model used for comprehensive evaluation of multiple metrics. The result of weight calculation is based on data distribution. It is not affected by subjective factors.

In our method,  $neighbor_1$ ,  $neighbor_2$  and  $neighbor_3$  will be fused by weighted average method in order to calculate the confidence score, and the points which are greater than the set score threshold will be categorized as non-noise points, otherwise they will be classed as noise points.

The following are the detailed steps for calculating the score for each point using the EWM method. Firstly, the corresponding  $neighbor_1$ ,  $neighbor_2$  and  $neighbor_3$  of the third frame points are arranged as a matrix  $F_{N \times 3}$  with  $N$  rows and 3 columns as shown in Equation (1). Where  $N$  rows represent  $N$  points in the third frame point cloud.

$$F_{N \times 3} = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ \dots & \dots & \dots \\ f_{N1} & f_{N2} & f_{N3} \end{bmatrix} = \begin{bmatrix} n_{1p_1} & n_{2p_1} & n_{3p_1} \\ n_{1p_2} & n_{2p_2} & n_{3p_2} \\ \dots & \dots & \dots \\ n_{1p_N} & n_{2p_N} & n_{3p_N} \end{bmatrix} \quad (1)$$

After that, the weight of each element of the matrix in its column  $pro_{i,j}$  is calculated as shown in Equation (2). Then calculate the entropy value  $e_j$  of each column as shown in Equation (3). Finally, the weight  $w_j$  of each column is calculated based on Equation (4), that is, the weights of the three neighbors indicators are calculated.

$$pro_{i,j} = \frac{f_{i,j}}{\sum_{i=1}^N f_{i,j}} \quad (2)$$

$$e_j = -\frac{1}{\ln(m)} \sum_{i=1}^m pro_{i,j} \ln pro_{i,j} \quad (3)$$

$$w_j = \frac{1 - e_j}{\sum_{j=1}^3 (1 - e_j)} \quad (4)$$

After calculating  $w_1$ ,  $w_2$ , and  $w_3$ , the weighted  $Score_i$  of point  $p_i$  is calculated by normalizing  $n_{1p_i}$ ,  $n_{2p_i}$ , and

$n_{3p_i}$  in Equation (5). The physical meaning of  $Score_i$  is the confidence score for determining whether the point  $p_i$  is a snow point or not relative to a certain threshold value. Based on the results of several experiments, we set the threshold to 0.27, which adequately meets the needs of most snowfall cases.

$$Score_i = w_1 \cdot \frac{n_{1p_i}}{n_{1p_i} + 1} + w_2 \cdot \frac{n_{2p_i}}{n_{2p_i} + 1} + w_3 \cdot \frac{n_{3p_i}}{n_{3p_i} + 1} \quad (5)$$

## IV. EXPERIMENT AND EVALUATION

### A. Dataset

The dataset used in this paper is the Boreas public dataset released by the University of Toronto in 2023 [33]. The Boreas dataset was collected by the survey vehicle as it repeatedly traversed multiple routes through Toronto, Canada, over multiple seasons and in multiple weather conditions. The LiDAR mounted on the survey vehicle is the 128-channel Velodyne Alpha Prime lidar, which has a horizontal resolution of 0.2 degrees.

The route we chose was Glen Shields, as it was the only route where point cloud data was collected in snowy weather. There are five sequences of snowy weather data in Boreas dataset, and we chose the 2021-1-26-11-22 sequence as a realistic scenario for the road during heavy snow, whereas the 2021-11-28 sequence was for medium snow, and the 2020-12-01 sequence for light snow. We did not choose other two sequences because the 2021-1-26-10-59 sequence was not scanned completely and the 2021-04-22 sequence was too special for the weather and road conditions. In addition, we selected the sequence 2021-04-08-12-44 to provide a realistic scene of the road in sunny weather for comparison with the in snowy weather. So in this paper, four sequences in the Boreas dataset are selected for heavy, medium, and light snow, as well as for sunny weather.

To evaluate the denoising performance of various methods, point cloud data collected under three different snowfall conditions—light, medium, and heavy—require point-by-point labeling to distinguish noise points from non-noise points. In order to simplify the workload of point cloud data labeling, we selected different road scenes in heavy, medium and light snow. The road scene selected in heavy snow weather is urban highway, the road scene selected in medium snow is residential road, and the road scene selected in light snow is intersection. The labeled point cloud data are shown in Figures 6-8.

During the labeling process for the point cloud data, we encountered challenges in accurately and comprehensively classifying each point as either noise or non-noise. For example, it is difficult to distinguish electrical wires and leaves from nearby snow noise points. In addition, the autonomous vehicle does not need to meticulously distinguish every point when sensing the surrounding environment. To reduce the workload of manually labeling noise points and to cater to the advanced perception requirements of autonomous driving, we have classified the noise points into the following three classes: airborne snow points, trailing shadows of moving vehicles, and ghosting of stationary objects on the ground. Figure 9

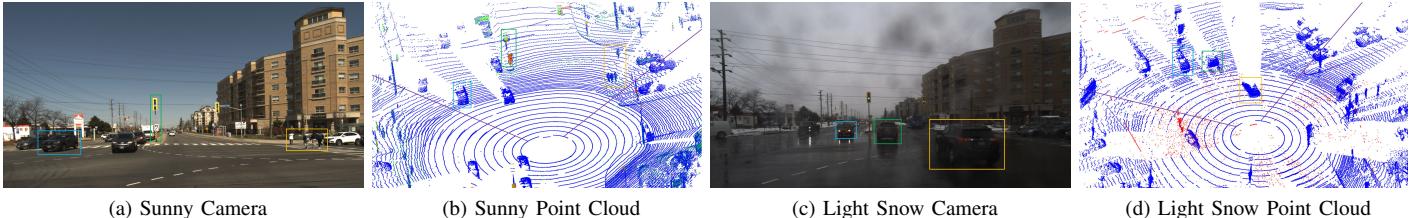


Fig. 6. Intersection.

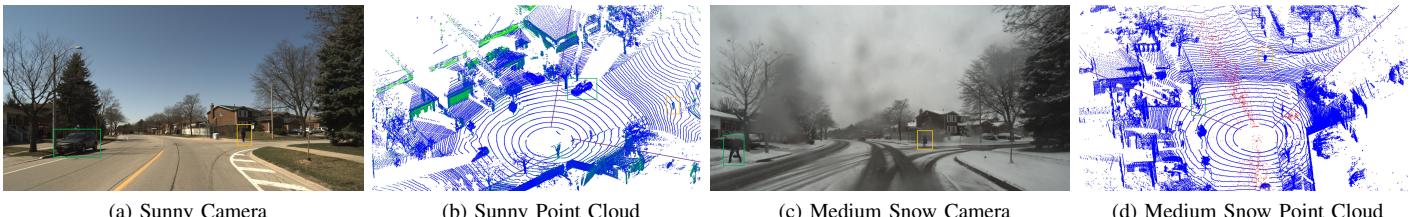


Fig. 7. Residential road.

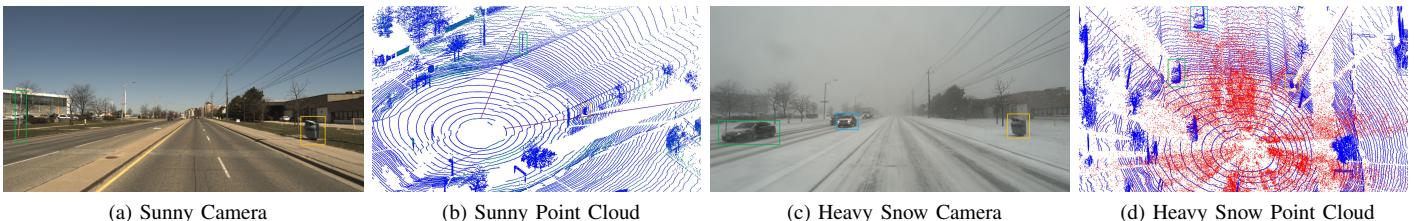


Fig. 8. Urban highway (the purple-lined area in the point cloud represents the camera's field of view in Figures 6 to 8).

illustrates the aforementioned three categories of noise points, using a frame of point cloud data collected from an urban highway scenario in light snow. In the actual labeling process, we focus on the airborne noise points, which are mainly airborne snow point, front end of the vehicle, electrical wire, and trailing shadows of moving vehicles.

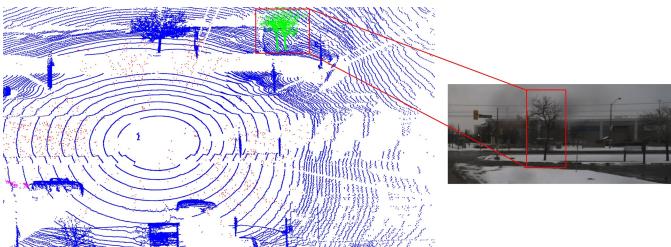


Fig. 9. Noise points in snowy weather point clouds (red points indicate airborne snow, purple points represent the trailing shadow of a moving vehicle, and green points depict the ghosting of a stationary tree on the ground).

### B. Evaluation Metrics

The quantitative evaluation metrics used in the experiment are: *accuracy*, *error*, *precision*, *recall*, *F score* and *time*. The equations for the first five quantitative metrics are as follows:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$\text{error} = \frac{FP + FN}{TP + TN + FP + FN} \quad (7)$$

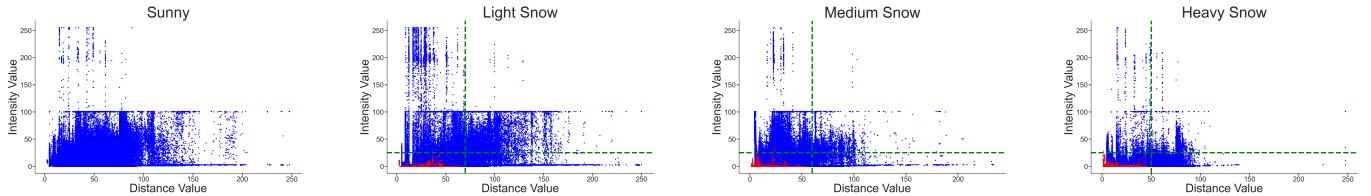
$$\text{precision} = \frac{TP}{TP + FP} \quad (8)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (9)$$

$$F \text{ score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (10)$$

where True Positive (TP) represents the number of noise points correctly identified as noise points; True Negative (TN) represents the number of non-noise points correctly identified as non-noise points; False Positive (FP) represents the number of non-noise points incorrectly identified as noise points; and False Negative (FN) represents the number of noise points incorrectly identified as non-noise points.

A good filtering method should have the following quantitative metrics: larger *accuracy*, smaller *error*, larger *precision*, larger *recall*, and larger *F score*. Among them, the larger the *precision* represents the stronger ability to retain environmental features; the larger the *recall* represents the stronger denoising ability. In order to quantitatively evaluate the filtering method proposed in this paper with other filtering methods, we will also perform similar evaluations for other methods, with every three consecutive frames of point cloud



**Fig. 10.** Intensity and distance analysis of point clouds under four weather conditions (blue for non-noise points, red for noise points, green dashed line for threshold).

as a denoising unit. The *accuracy*, *error*, *precision*, *recall* and *F score* of the other methods take the average of the three frames of the point cloud, while time is the sum of the three frames of the point cloud processing time. The TOR(EWM) filter's processing time includes the total time, ground filtering time, point cloud registration time, and denoising time.

### C. Noise Points Distance and Intensity Thresholds Analysis

Previous studies have identified that airborne snow points typically exhibit characteristics such as high density, low intensity, close range and fast decay, which means that there is an objective intensity threshold and distance threshold for airborne snow points [32]. In this paper, the intensity and distance of point clouds under four weather conditions are analyzed, and the results are shown in Figure 10.

From Figure 10, it can be seen that with the increase of snowfall, the range of LiDAR scanning gradually decreases, and the distance threshold of noise points gradually decreases. Considering that this paper not only labels the airborne snow points as noise points, but also labels other kinds of noise points, a marginal number of noise points exceeding the specified thresholds can be tolerated in the process of selecting intensity and distance thresholds. The values of the specific intensity thresholds and distance thresholds are shown in Table 1.

**TABLE I**  
INTENSITY AND DISTANCE THRESHOLDS

	Light Snow	Medium Snow	Heavy Snow
Intensity Threshold	25	25	25
Distance Threshold	70	60	50

### D. Comparison Experiments

We will conduct experiments using seven filters and a learning-based method, the seven filters include ROR, SOR, DROR, DSOR, DDIOR, LIDSOR and TOR (EWM). The learning-based method is LiSnowNet. The experiments will be conducted under three weather conditions: light snow, medium snow and heavy snow. The task of point cloud denoising can be viewed as a simplified form of the point cloud semantic segmentation task. It only requires each point in the point cloud to be simply classified [34] as a noise or non-noise point, i.e., a binary classification problem. Figures 11-13 demonstrate the classification results of the filtering method. In

the classification results of the filtering method, we can see not only the points identified as non-noise by the filtering method, but also the points identified as noise by the filtering method. This helps us to comprehensively evaluate the performance of the filtering method in terms of both denoising ability and retention of environmental features. The parameters of the filters used in the experiment are shown in Table 2. The LiSnowNet model is trained on the Boreas dataset with a batch size of 30. The training process spans 30 epochs, utilizing the Adam optimizer with an initial learning rate of 0.001. The learning rate is updated at the end of each epoch with a learning decay of 0.926.

**TABLE II**  
FILTERS PARAMETERS SETTING

Filters	Parameters	Value
ROR	Minimum number of neighbors	3
	Search radius	0.3
SOR	Number of neighbors	12
	Standard deviation	0.1
DROR	Minimum number of neighbors	3
	Multiplication factor	1.75
	Horizontal angular resolution of the lidar	0.2
	Minimum search radius	0.12
DSOR	Minimum number of nearest neighbors	12
	Multiplication factor for standard deviation	0.1
	Multiplication factor for range	0.07
DDIOR	Number of nearest neighbors	12
	Dynamic distance coefficient	0.2
LIDSOR	Minimum number of nearest neighbors	12
	Multiplication factor for standard deviation	0.12
	Multiplication factor for range	0.12
	Intensity threshod	
	Distance threshod	
TOR(EWM)	Multiplication factor	2.5
	Horizontal angular resolution of the lidar	0.2
	Minimum search radius	0.12
	Distance threshod	
	Intensity threshod	
	Search radius	0.1
	Score threshold for ewm	0.27

**1) Qualitative Evaluation:** The experimental road scene we selected is an intersection in light snow, as depicted in Figure 11 (a-b). The purple box in Figure 11 (b) indicates the airborne snow points, the red box indicates the trailing shadows of the moving vehicles, the green box indicates the electrical wires, and the yellow box indicates the front ends of the vehicles scanned by LiDAR. The classification results in light snow are shown in Figure 11 (c-j). The denoising capabilities of the ROR and SOR filters are moderate. However, as observed in Figure 11 (c-d), these filters incorrectly classify all the non-noise points located at a distance from the LiDAR as

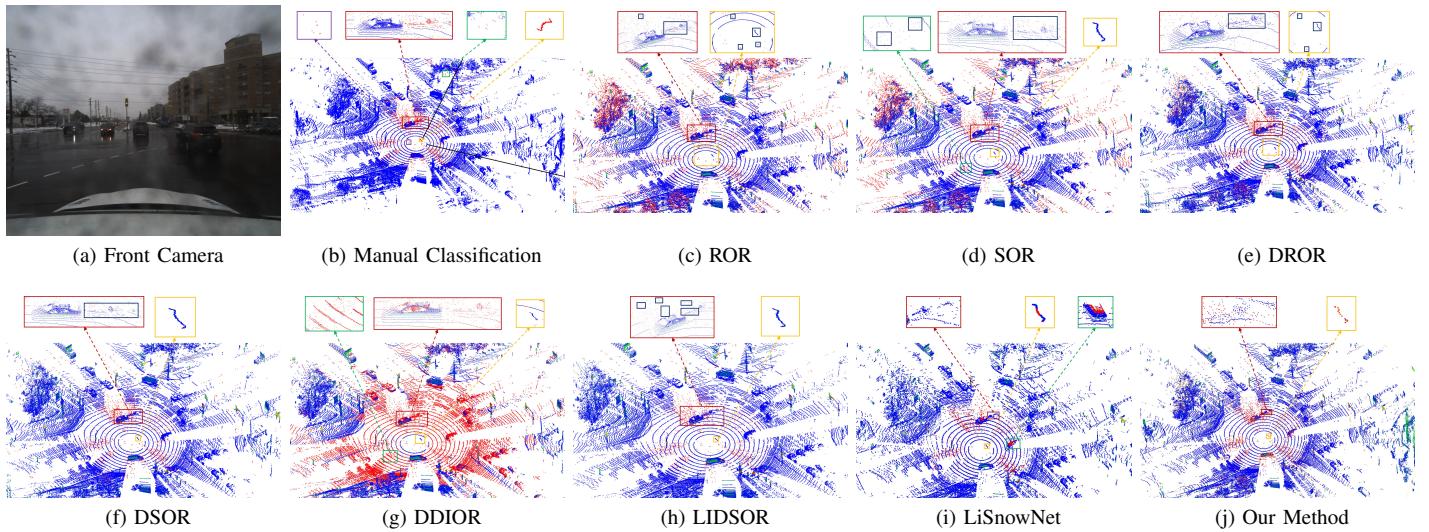


Fig. 11. Classification results in light snow.

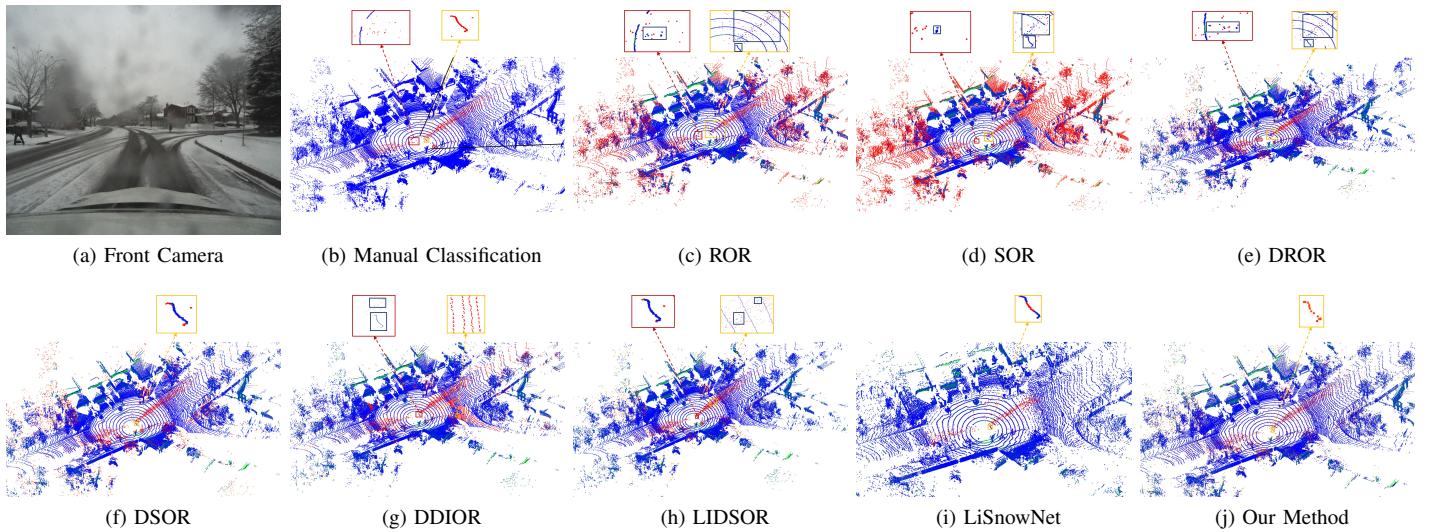


Fig. 12. Classification results in medium snow.

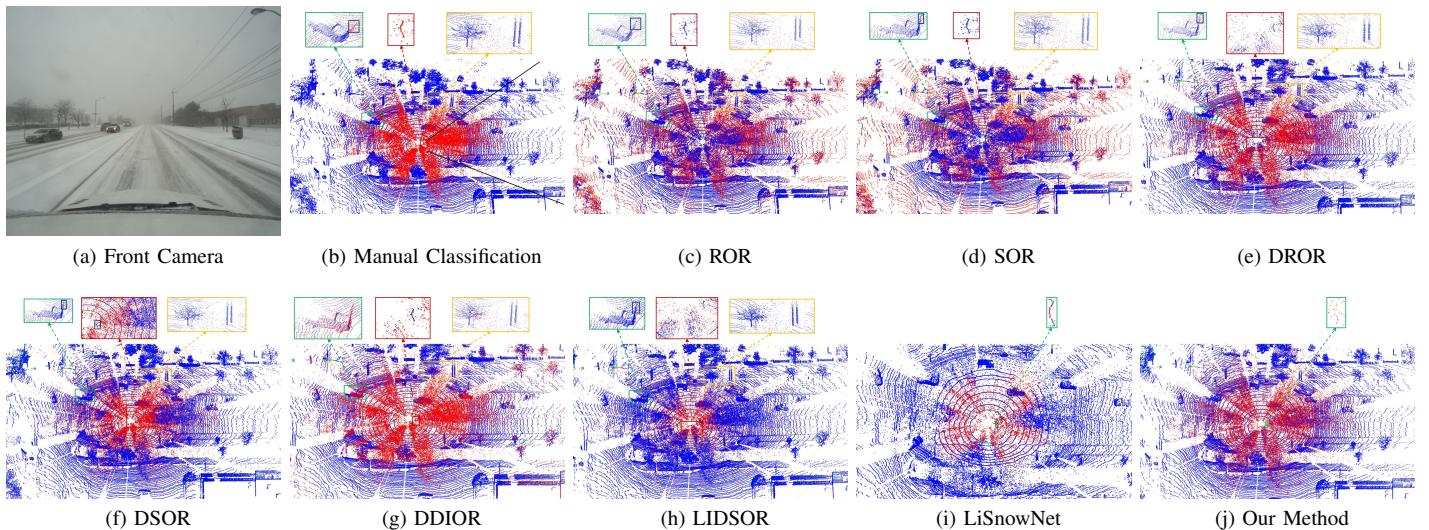


Fig. 13. Classification results in heavy snow (in Figures 11 to 13, blue represents non-noise points, red represents noise points; the closer the filter classification results are to the manual classification results, the better the performance of the filters).

noise points, resulting in a significant loss of environmental features. This is because these two filters do not consider the characteristics of the point cloud that the density of the point cloud decreases with increasing distance from the sensor. From the boxes in Figure 11 (c-d), it can be noticed that both of them fail to identify the trailing shadows and the front end of vehicles as noise points, and there are still some airborne snow noise points that are incorrectly identified as non-noise points. In order to enhance the ability to retain environmental features of the ROR and SOR filters, the DROR and DSOR filters were respectively improved. The DROR adjusted the fixed search radius of the ROR to a dynamic search radius, while the DSOR adjusted the fixed threshold of the SOR to a dynamic threshold. From Figure 11 (e-f), it can be observed that most of the non-noise points at long distances are correctly identified, and DSOR is better than DROR. however, they are also unable to identify the trailing shadows and the front end of a moving vehicle as noise points, and there are also some airborne snow points that are identified as non-noise points. As seen in Figure 11 (g), DDIOR incorrectly identifies most of the ground point as noise points (as shown in the green box). While it correctly identifies the trailing shadow of a moving vehicle as noise points, it also incorrectly identifies a portion of the vehicle as noise points. Also, it fails to identify the front end of the vehicle as noise points. LIDSOR correctly identifies most of the noise points and preserves most of the environmental features, yet fails to identify the trailing shadows and front end of the vehicle as noise points, as can be seen from the box in Figure 11 (h). From Figure 11 (i), it can be found that although LiSnowNet is able to identify some points of the front end as noise points, it is not able to identify the trailing shadows of the vehicle as noise points, and additionally incorrectly identifies some points of the vehicle as noise points. The experimental result of our method is shown in Figure 11 (j). From Figure 11 (j), it can be found that our method not only identifies the front end of the vehicle as noise points, but also correctly identifies most of the airborne snow points as noise points. Although in Figure 11 (j), the TOR(EWM) filter does not identify all the trailing shadows of the moving vehicle as noise points. It identifies more trailing shadows as noise points and is even able to remove the dynamic vehicle when the score threshold is set differently, as detailed in extended applications.

The road scenario chosen for the experiment in medium snow is a residential road, as shown in Figure 12 (a-b). The red box in Figure 12 (b) indicates the airborne snow points, and the yellow box indicates the front end of a vehicle. The classification results in medium snow are shown in Figure 12 (c-j). Both the ROR and SOR filters fall short in their ability to preserve environmental features at a distance. Comparing Figure 12 (c) and (d), it can be seen that SOR is less capable of retaining environmental features than ROR. The boxes in these figures further reveal that the ROR filter misclassifies a large number of snow noise points as non-noise points. This implies that SOR has better denoising ability than ROR. The DROR and DSOR filters have improved environmental feature retention ability compared to the ROR and SOR filters. Also comparing the boxes in Figure 12 (e) and (f), it can

be observed that the denoising ability of the DSOR filter surpasses that of the DROR filter. The DDIOR filter identifies some of the ground points as noise points as shown in the yellow box in Figure 12 (g). The LIDSOR filter is unable to identify the front end of the vehicle as noise points as shown in the red box in Figure 12 (h). Also from the yellow box it is found that it incorrectly identifies some of the snow noise points as non-noise points. LiSnowNet is able to identify some points of the front end as noise points, as shown in the yellow box in Figure 12 (i). The experimental result of the TOR (EWM) filter is shown in Figure 12 (j). It correctly identifies most of the airborne snow points, and it can be noticed from the yellow box that it correctly identifies the front end of vehicle as noise points as well. In addition, the TOR(EWM) filter even identifies all of the airborne snow points as noise points by simply adjusting the score threshold, as detailed in extended applications.

The road scene chosen for the experiment in heavy snow is an urban highway, as shown in Figure 13 (a-b). From Figure 13 (b), the air is so densely filled with snow in heavy snow that it poses a significant challenge for any filter. In addition, unlike the light and medium snow, the urban highway scene in heavy snow demonstrates for the first time the ghosting problem of stationary objects on the ground, as shown in the yellow box. The classification results in heavy snow are shown in Figure 13 (c-j). The ROR and SOR filters in heavy snow have the weakest denoising ability compared to the experimental results in medium and light snow. They are also unable to identify the trailing shadows of moving vehicles, the front end of the vehicle and the ghosting of stationary objects on the ground as noise points, as shown by the boxes in Figure 13 (c-d). DROR and DSOR also retain more environmental features compared to both ROR and SOR filters, as in the previous experiments. Comparing Figure 13 (e) and (f), it can be seen that the DROR filter have better denoising ability than the DSOR filter, but they also cannot solve the problems of the front end of the vehicle and the ghosting of stationary objects on the ground. From Figure 13 (g), it is evident that the DDIOR filter exhibits the strongest denoising capabilities among the filters evaluated. It identifies most all of the noise points correctly. And, it correctly identifies the trailing shadow of the moving vehicle as noise points, but identifies a part of the vehicle as noise points, as shown in the green box. The LIDSOR filter and LiSnowNet are unable to identify most of the snow points in the air as noise points, as shown in Figure 13(h-i). The experimental result of the TOR(EWM) filter proposed in this paper are shown in Figure 13 (j). It removes most of the noise points while retaining most of the environmental feature points and identifies the front end of vehicle as noise points as shown in the green box. Moreover, by adjusting the score threshold, the TOR(EWM) filter not only identifies all noise points correctly, but even identifies the ghosting of a stationary object on the ground as noise points, as detailed in extended applications.

*2) Quantitative Evaluation:* The evaluation metric results are presented in Table 3. Comparing different snowfall scenarios, we find that most of the methods perform better in the heavy snow than in the medium and light snow. This is mainly due to

the fact that the density of snow noise points in the heavy snow is much higher than that in the light and medium snow, and the snow noise points are more uniformly distributed, which allows the filtering methods to better utilize the correlation between the snow noise points to identify the noise points. The reason for the poor performance of the LIDSOR filter in heavy snow is that as the density of snow noise points around the sensor increases, it only relies on the points within the distance threshold to identify the noise points, and cannot utilize the points that are beyond the distance threshold for analysis. The reason for LiSnowNet's poor performance in heavy snow is a common problem with deep learning methods, which can only be adapted to specific snowfall scenarios. When the snowfall increases and the density of snow noise points gradually increases, the performance of LiSnowNet decreases.

Comparing the different methods, we find that the ROR and SOR filters do not perform well in light to medium snow, mainly because they do not take into account the characteristic of point clouds being denser near and sparser far, and thus fail to retain environmental feature points at medium and long distances. However, the DROR and DSOR filters, which take into account the characteristic of point clouds being denser near and sparser far, have stable performance in different snowfall scenarios, while the DDIOR filter has a lower performance because it identifies some of the ground points as noise points in the medium and light snow. However, as the density of snow noise points increases, the DDIOR filter is able to better utilize the correlation between the noise points for analysis, so it shows good performance in heavy snow.

Overall, the ROR and SOR filters fail to denoise LiDAR point clouds effectively due to their inability to account for the point clouds' characteristic of being denser near and sparser far. The DDIOR, LIDSOR, and LiSnowNet methods are limited to specific snowfall scenarios. In contrast, the DROR, DSOR, and TOR(EWM) filters show adaptability to various snow conditions, with the TOR(EWM) filter proposed in this study demonstrating the strongest performance.

In terms of denoising efficiency, LiSnowNet has the shortest processing time, partly because it is an unsupervised learning method based on range view. Additionally, it benefits from acceleration using CUDA, which also contributes to the shorter point cloud registration time in the TOR(EWM) filter. If we compare the denoising time between seven filters, the denoising time of TOR(EWM) filter proposed in this paper is much less than other filters. If the total processing time between the seven filters is compared, the time of the proposed method in this paper is also almost the shortest. The time complexity of our proposed TOR filter is  $O(n \log n)$ .

### E. Extended Applications of Our Method

The TOR(EWM) filter designed in this paper exhibit different characteristics as the score threshold is varied. Notably, it can preserve stationary ground objects while eliminating dynamic vehicles, enhancing its denoising capability, and removing the ghosting of stationary objects on the ground. The extended applications of this innovative approach are delineated in the subsequent sections.

**1) Dynamic Vehicles Removal:** The result of point clouds registration from three consecutive frames, obtained after removing ground points from each frame in light snow, is shown in Figure 14 (a). The points from different frames are indicated by red, green and blue colors in Figure 14 (a). In addition, the blue box in the figure indicates a stationary object on the ground, which can be registered, so it is presented in brown color; while the red box in the figure indicates a moving vehicle, which cannot be registered, so it is presented in red, green, and blue colors sequentially. When the score threshold is set to 0.33, the TOR(EWM) filter removes the moving vehicle and retains the stationary object on the ground, as shown in Figure 14 (b).

**2) Enhanced Denoising Capability:** In the comparison experiments of the filters in medium snow, the TOR(EWM) filter identifies most of the airborne snow points as noise points, but there are still some airborne snow points that are incorrectly

TABLE III  
QUANTITATIVE EVALUATION RESULTS

	ROR	SOR	DROR	DSOR	DDIOR	LIDSOR	LiSnowNet	TOR(EWM)
Light Snow	Accuracy	0.74	0.76	0.97	0.98	0.68	0.99	<b>0.97</b>
	Error	0.26	0.24	0.03	0.02	0.32	0.01	<b>0.03</b>
	Precision	0.06	0.06	0.39	0.64	0.06	0.85	<b>0.61</b>
	Recall	0.71	0.74	0.67	0.65	0.86	0.61	<b>0.78</b>
	F score	0.11	0.12	0.49	0.64	0.11	0.7	<b>0.68</b>
	Time(ms)	612.6	936.01	1113.4	899.37	917.3	758.2	316.09
<b>631.9(102.28/58.89/470.73)</b>								
Medium Snow	Accuracy	0.88	0.8	0.97	0.97	0.95	0.99	<b>0.95</b>
	Error	0.12	0.2	0.03	0.03	0.05	0.01	<b>0.05</b>
	Precision	0.1	0.07	0.34	0.34	0.2	0.62	<b>0.46</b>
	Recall	0.79	0.89	0.79	0.89	0.76	0.79	<b>0.87</b>
	F score	0.18	0.13	0.47	0.48	0.32	0.69	<b>0.6</b>
	Time(ms)	782.01	1276.18	1521.23	1159.17	1155.16	1085.78	298.80
<b>612.91(129.75/43.68/439.48)</b>								
Heavy Snow	Accuracy	0.76	0.73	0.93	0.89	0.86	0.83	<b>0.88</b>
	Error	0.24	0.27	0.07	0.11	0.14	0.17	<b>0.12</b>
	Precision	0.49	0.43	0.92	0.96	0.64	0.99	<b>0.97</b>
	Recall	0.51	0.46	0.77	0.56	0.95	0.27	<b>0.74</b>
	F score	0.5	0.45	0.84	0.71	0.76	0.43	<b>0.84</b>
	Time(ms)	762.53	1178.65	1144.26	1145.67	1141.53	957.33	316.89
<b>798.49(123.34/69.60/605.55)</b>								

Note: The table shows the total processing time of the TOR(EWM) method, listed outside the brackets. Within the brackets, the individual processing times are presented in the following order: ground filtering, point cloud registration, and denoising.

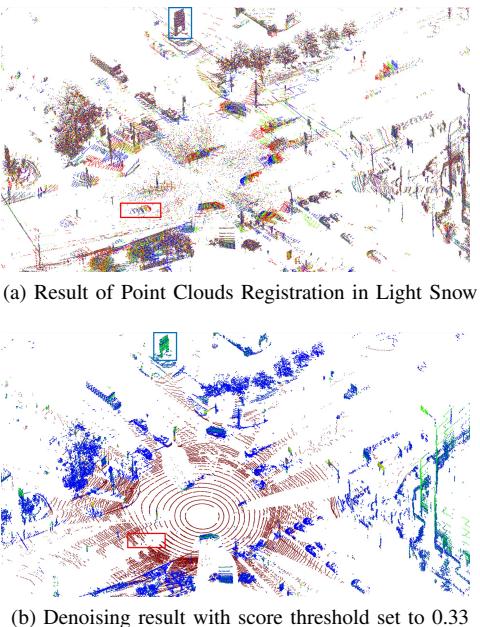


Fig. 14. Dynamic vehicles removal.

identified as non-noise points, as shown by the purple boxes in Figure 15 (a). However, with the score threshold set to 0.45, the TOR(EWM) filter removes almost all of the noise points without losing too much of the environmental features, as shown in Figure 15 (b).

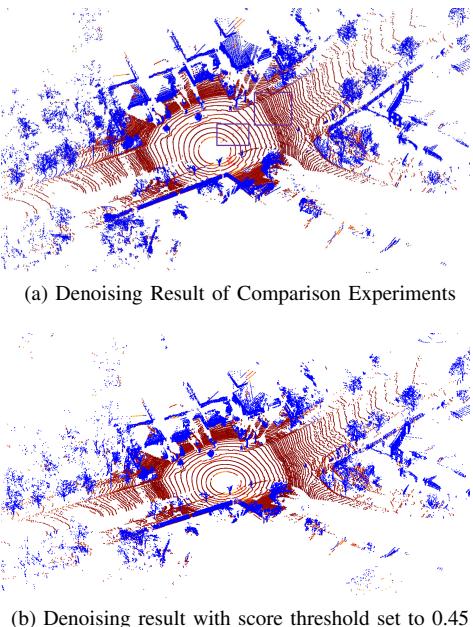


Fig. 15. Enhanced denoising capability.

**3) Removal of Ghosting from Stationary Objects on the Ground:** The registration result for three consecutive frames of heavy snow point clouds is shown in Figure 16 (a). From the figure, it is observable that the ghosting of stationary objects on the ground does not appear continuously across three consecutive frames of point clouds. Consequently, in the registration result, ghosting appears in red, as highlighted in the purple box. The filter developed in this paper considers the number

of neighbors across three frames of point clouds for denoising. With the score threshold set to 0.45, the TOR(EWM) filter effectively removes the ghosting of stationary ground object, as illustrated in Figure 16 (b).

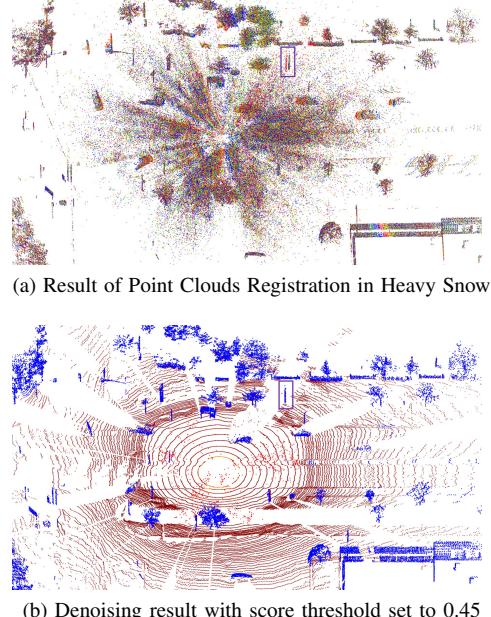


Fig. 16. Removal of ghosting from stationary objects on the ground.

## V. CONCLUSION

To extend the application of autonomous driving technology in snowy weather and enhance LiDAR perception under such condition, this paper proposed a point cloud denoising method for LiDAR in snowy weather. We took into consideration the prior knowledge that snow noise points exhibit disorder across multiple consecutive frames of point clouds. Based on this, we deviated from the traditional single-frame denoising approach and designed the TOR filter specifically for denoising multiple consecutive frames of point clouds. Finally, in order to preserve dynamic non-snow objects, we combine the TOR filter with the EWM method. The experimental results show that the method proposed in this paper performs well in denoising ability as well as environmental feature retention in heavy, medium and light snow, and the processing time is the shortest. Moreover, our method enables ordered objects to reinforce each other while causing disordered objects to mutually weaken. As a result, the method is effective not only in removing airborne snow points but also in eliminating dynamic non-snow noise points. This includes trailing shadows from moving vehicles, ghosting from stationary objects on the ground, and even the moving vehicles. This greatly expands the usage cases of our method, which can not only serve more advanced perception tasks of autonomous driving, such as 3D object detection, but also tasks that require the removal of dynamic objects, such as SLAM. The denoising method proposed in this paper can effectively remove noise points from LiDAR point clouds in snowy weather, which is crucial for the advancement of autonomous driving technology in severe weather.

## REFERENCES

- [1] W. Wang, T. Yang, Y. Du, and Y. Liu, "Snow removal for lidar point clouds with spatio-temporal conditional random fields," *IEEE Robotics and Automation Letters*, 2023.
- [2] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1–4.
- [3] N. Charron, S. Phillips, and S. L. Waslander, "De-noising of lidar point clouds corrupted by snowfall," in *2018 15th Conference on Computer and Robot Vision (CRV)*. IEEE, 2018, pp. 254–261.
- [4] A. Kurup and J. Bos, "Dsor: A scalable statistical filter for removing falling snow from lidar point clouds in severe winter weather," *arXiv preprint arXiv:2109.07078*, 2021.
- [5] J.-I. Park, J. Park, and K.-S. Kim, "Fast and accurate desnowing algorithm for lidar point clouds," *IEEE Access*, vol. 8, pp. 160202–160212, 2020.
- [6] A. Seppänen, R. Ojala, and K. Tammi, "4denoisenet: Adverse weather denoising from adjacent point clouds," *IEEE Robotics and Automation Letters*, vol. 8, no. 1, pp. 456–463, 2022.
- [7] B. Li, J. Li, G. Chen, H. Wu, and K. Huang, "De-snowing lidar point clouds with intensity and spatial-temporal features," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2359–2365.
- [8] X.-F. Han, J. S. Jin, M.-J. Wang, W. Jiang, L. Gao, and L. Xiao, "A review of algorithms for filtering the 3d point cloud," *Signal Processing: Image Communication*, vol. 57, pp. 103–112, 2017.
- [9] E. A. L. Narváez and N. E. L. Narváez, "Point cloud denoising using robust principal component analysis," in *International Conference on Computer Graphics Theory and Applications*, vol. 2. SCITEPRESS, 2006, pp. 51–58.
- [10] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Point set surfaces," in *Proceedings Visualization, 2001. VIS'01*. IEEE, 2001, pp. 21–29.
- [11] P. Jenke, M. Wand, M. Bokeloh, A. Schilling, and W. Straßer, "Bayesian point cloud reconstruction," in *Computer graphics forum*, vol. 25, no. 3. Wiley Online Library, 2006, pp. 379–388.
- [12] N. A. I. M. Rosli and A. Ramli, "Mapping bootstrap error for bilateral smoothing on point set," in *AIP conference proceedings*, vol. 1605, no. 1. American Institute of Physics, 2014, pp. 149–154.
- [13] B. Huhle, T. Schairer, P. Jenke, and W. Straßer, "Robust non-local denoising of colored depth data," in *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2008, pp. 1–7.
- [14] B. Brown, "Statistical uses of the spatial median," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 45, no. 1, pp. 25–30, 1983.
- [15] H. Balta, J. Velagic, W. Bosschaerts, G. De Cubber, and B. Siciliano, "Fast statistical outlier removal based method for large 3d point clouds of outdoor environments," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 348–353, 2018.
- [16] G. Taubin, "A signal processing approach to fair surface design," in *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995, pp. 351–358.
- [17] U. Clarenz, M. Rumpf, and A. Telea, "Fairing of point based surfaces," in *Proceedings Computer Graphics International, 2004*. IEEE, 2004, pp. 600–603.
- [18] S. Liu, K.-C. Chan, and C. C. Wang, "Iterative consolidation of unorganized point clouds," *IEEE computer graphics and applications*, vol. 32, no. 3, pp. 70–83, 2011.
- [19] J. Liu, G. Wang, W. Ye, C. Jiang, J. Han, Z. Liu, G. Zhang, D. Du, and H. Wang, "Diffflow3d: Toward robust uncertainty-aware scene flow estimation with diffusion model," *arXiv preprint arXiv:2311.17456*, 2023.
- [20] R. Heinzler, F. Piewak, P. Schindler, and W. Stork, "Cnn-based lidar point cloud de-noising in adverse weather," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2514–2521, 2020.
- [21] F. Piewak, P. Pinggera, M. Schafer, D. Peter, B. Schwarz, N. Schneider, M. Enzweiler, D. Pfeiffer, and M. Zollner, "Boosting lidar-based semantic labeling by cross-modal training data generation," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.
- [22] Y. Duan, C. Yang, and H. Li, "Low-complexity adaptive radius outlier removal filter based on pca for lidar point cloud denoising," *Applied Optics*, vol. 60, no. 20, pp. E1–E7, 2021.
- [23] H. Balta, J. Velagic, W. Bosschaerts, G. De Cubber, and B. Siciliano, "Fast statistical outlier removal based method for large 3d point clouds of outdoor environments," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 348–353, 2018.
- [24] H. Huang, X. Yan, J. Yang, Y. Cao, and X. Zhang, "Lidsor: A filter for removing rain and snow noise points from lidar point clouds in rainy and snowy weather," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLVIII-1/W2-2023, pp. 733–740, 2023. [Online]. Available: <https://isprs-archives.copernicus.org/articles/XLVIII-1-W2-2023/733/2023/>
- [25] T. Cortinhal, G. Tzelepis, and E. Erdal Aksoy, "Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds," in *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II* 15. Springer, 2020, pp. 207–222.
- [26] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, F. Yu, D. Dai, and L. Van Gool, "Lidar snowfall simulation for robust 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16364–16374.
- [27] M.-Y. Yu, R. Vasudevan, and M. Johnson-Roberson, "Lisnownet: Real-time snow removal for lidar point clouds," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 6820–6826.
- [28] G. Bae, B. Kim, S. Ahn, J. Min, and I. Shim, "Slide: Self-supervised lidar de-snowing through reconstruction difficulty," in *European Conference on Computer Vision*. Springer, 2022, pp. 283–300.
- [29] S. Lee, H. Lim, and H. Myung, "Patchwork++: Fast and robust ground segmentation solving partial under-segmentation using 3d point cloud," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13276–13283.
- [30] J. Liu, G. Wang, Z. Liu, C. Jiang, M. Pollefeys, and H. Wang, "Reformer: An efficient projection-aware transformer network for large-scale point cloud registration," *arXiv preprint arXiv:2303.12384*, 2023.
- [31] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized gicp for fast and accurate 3d point cloud registration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 11054–11059.
- [32] W. Wang, X. You, L. Chen, J. Tian, F. Tang, and L. Zhang, "A scalable and accurate de-snowing algorithm for lidar point clouds in winter," *Remote Sensing*, vol. 14, no. 6, p. 1468, 2022.
- [33] K. Burnett, D. J. Yoon, Y. Wu, A. Z. Li, H. Zhang, S. Lu, J. Qian, W.-K. Tseng, A. Lambert, K. Y. Leung *et al.*, "Boreas: A multi-season autonomous driving dataset," *The International Journal of Robotics Research*, vol. 42, no. 1-2, pp. 33–42, 2023.
- [34] B. E. Aissou, A. B. Aissa, A. Dairi, F. Harrou, A. Wichmann, and M. Kada, "Building roof superstructures classification from imbalanced and low density airborne lidar point cloud," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 14960–14976, 2021.



**Xinyuan Yan** is currently pursuing his master's degree at the School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture. His research focuses on the challenges faced by autonomous driving perception systems under adverse weather conditions. His areas of expertise include point cloud denoising, point cloud semantic segmentation, and 3D occupancy prediction.

**Junxing Yang** received his PhD degree from Wuhan University. His research interests are photogrammetry and remote sensing, 3D reconstruction, image stitching and scene understanding.



**Xinyu Zhu** is pursuing the bachelor degree at the School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture, China. Her major is surveying and mapping engineering.



**Yu Liang** is currently pursuing her master's degree at the School of Geomatics and Urban Spatial Informatics, Beijing University of Civil Engineering and Architecture. Her research direction is point cloud denoising.



**He Huang** received his B.S. degree from Wuhan University, China, in 2000 and his M.S. and Ph.D. degrees from Sungkyunkwan University, South Korea, in 2004 and 2010, respectively. Since 2010, he has been a lecturer and associate professor at Beijing University of Civil Engineering and Architecture, China. His research interests are in autonomous driving, high-precision navigation maps, and visual navigation and positioning.

