# RGOR: De-noising of LiDAR point clouds with reflectance restoration in adverse weather

Seung-Jun Han, Dongjin Lee, Kyoung-Wook Min and Jeongdan Choi
*Autonomous Driving Intellgence Research Section*
*Electronics and Telecommunications Research Institute (ETRI)*
Daejeon, Republic of Korea
{joshua, robin2002, kwmin92 and jdchoi}@etri.re.kr

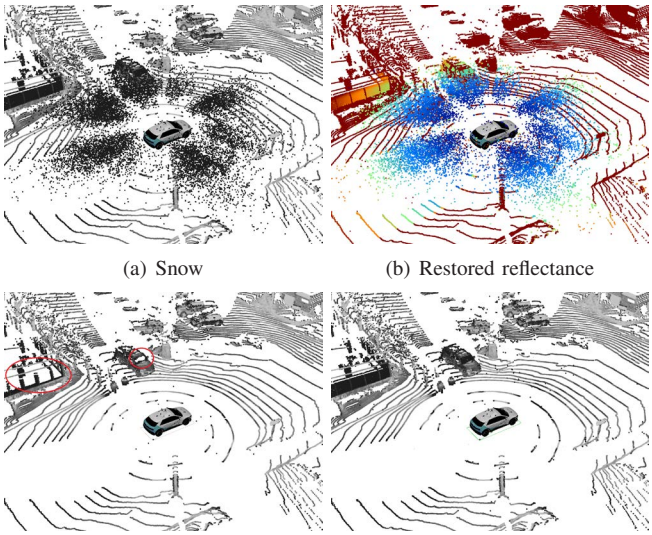
***Abstract*—Recently, LiDAR sensors have become indispensable in autonomous driving research. Despite continuous improvements in performance and price reductions, noise generated under adverse weather conditions remains a serious challenge. Most of the noise generated under such conditions is due to particles such as fog, rain, and snow. These particles are extremely fine; therefore, they have a very low reflectance compared to the targets that the laser should detect. In this study, we propose a method to distinguish particles by restoring the reflectance from LiDAR sensing data based on the reflectance characteristics of the particles. In addition, we propose a method to make additional judgments based on the geometrical shapes of adjacent particles to distinguish the particles more accurately. The proposed method is accurate enough to be compared to state-of-the-art deep learning methods. Moreover, the execution time is less than 2 ms on a single-core CPU, demonstrating a remarkable efficiency, being more than three times faster than that of methods performed on a GPU. Because noise removal is a preprocessing step, the proposed method is expected to allow more resources to be allocated to other, more important processes for autonomous driving.**

***Index Terms*—Point cloud filtering, De-noising LiDAR, outlier removal, adverse weather, self-driving, autonomous vehicle**

## I. Introduction

Light-detection and ranging (LiDAR) sensors are currently among the most prominent sensors used in autonomous vehicles. It collects information regarding the physical environment in the form of point clouds, which are then used for tasks such as localization and mapping, detection of moving objects, and obstacle avoidance during autonomous driving. Unfortunately, LiDAR sensors are susceptible to adverse weather conditions such as fog, rain, and snow, which can severely affect the quality of 3D point-cloud data. These airborne particles can interfere with the path of a laser beam, preventing it from reaching its target and resulting in noise in the detected data [1]. As research on autonomous driving matures, researchers face more frequent encounters with adverse weather conditions. Concurrently, issues related to noise in point clouds have gained prominence. Traditional noise reduction techniques for point clouds are not readily applicable to the rotating LiDAR systems primarily utilized in autonomous vehicles [2].

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIP) (No. 2020-0-00002, Development of standards SW platform based autonomous driving technology to solve social problems of mobility and safety for public transport marginalized communities)



(a) Snow (b) Restored reflectance

(c) De-noising with only reflectance (d) De-noising with geometry added

Fig. 1. Reflectance and Geometrical Outlier Removal (RGOR) algorithm: (a) Snowflakes create substantial noise. (b) The restoration of reflectance effectively distinguishes snow particles from targets. (c) When using only reflectance for noise removal, it can effectively eliminate noises. However, it might also eliminate objects with extremely low reflectance, such as glass. (d) Incorporating geometry information for noise removal enhances accuracy.

Consequently, various enhanced filtering techniques [2]–[6], which build upon existing methods, have been developed. Moreover, in alignment with current trends, research is actively exploring approaches based on deep learning [7]–[10].

The purpose of this study is to develop a lightweight and efficient method that can be directly applied to autonomous vehicles with limited power and computational capacity. The filter method for searching neighboring points, which improves upon existing algorithms, is excessively inefficient and unsuitable for use in actual autonomous vehicles. Additionally, deep learning methods encounter issues with heavy dependence on training data, and assigning point-wise labels to small particles, such as fog, rain, and snow, is extremely challenging.

In this study, we analyze LiDAR sensor optical characteristics for noise detection caused by particles and propose an efficient CPU algorithm: the Reflectance and Geometrical Outlier Removal (RGOR) technique. RGOR, illustrated in

Figure 1, leverages LiDAR laser reflections and nearby point cloud geometry. Initially, we reconstruct reflectance using 3D point and intensity data, which distinctly identifies particles. Subsequently, effective particle removal is achieved through reflectance alone, albeit also removing low-reflectivity entities such as glass. Finally, we enhance classification accuracy via geometric data. Our approach achieves 2 ms efficiency on a single-core CPU, maintaining accuracy comparable to cutting-edge deep learning techniques. This efficiency reserves additional GPU resources for crucial autonomous driving functions [11], [12]. Notably, the method is not reliant on learning, enabling adaptation to diverse sensors and environments.

The contributions of this study are as follows.

- We introduced a reflectance restoration method from the optical characteristics of the LiDAR sensor.
- We proposed a method to remove particles, which are outliers of LiDAR sensors, only with restored reflectance.
- We also proposed improving the classification accuracy method, including the distribution shape of 3D points.
- Our proposed method is straightforward; however, it outperforms state-of-the-art deep learning methods.

The rest of this paper covers the related work in Section II, outlines the optical characteristics of LiDAR, which motivated this algorithm, in Section III, presents the proposed algorithm in Section IV, discusses evaluation results in Section V, and concludes in Section VI.

## II. Related Works

Point cloud filtering improves the quality of 3D data, with numerous studies focused on 3D modeling and mapping [13]. Research on autonomous vehicles, however, is a relatively recent development. The direct application of traditional research is challenging due to the unique statistical characteristics of 3D point-cloud models and vehicle sensor data [2]. This section focuses on noise removal techniques for modern autonomous vehicles, categorized as statistics-based and deep learning-based, each with concise descriptions.

### A. Statistic-based Methods

The statistical approach uses the statistical characteristics of point-cloud distribution. Classically, there are radius outlier removal (ROR) and statistical outlier removal (SOR) [2] filters. Representative examples include dynamic radius outlier removal (DROR) [2] and dynamic statistical outlier removal (DSOR) [3], which have been expanded to autonomous vehicles by using ROR and SOR, respectively. The DROR is the first research result in this field and is based on the number of neighbors of each point. A $k$-d tree [14] is constructed, and outliers are classified by the number of neighbors obtained using the dynamic search radius, based on the fact that the density of the target points is inversely proportional to the distance. The search radius of each point was dynamically adjusted according to the distance.

DSOR also depends on nearest-neighbor search; however, it computes the mean and variance of relative distances based on a fixed number of neighbors for each point. The local density around each point is estimated and used to filter the outliers. Low-intensity outlier removal (LIOR) [4] is similar to ROR. Each point is predicted to be a candidate noise due to the low intensity. Subsequently, the predicted points are filtered using a noise intensity level that is lower than that of the target for the same distance. Dynamic light intensity outlier removal (DIOR) [5] integrates LIOR and DROR to improve the accuracy by dynamically adjusting the filtering radius. Dynamic distance-intensity outlier removal (DDIOR) [6] integrates LIOR and DSOR. The above statistical approach inevitably requires a long time to search for neighboring points, making real-time processing difficult. This may not be an appropriate approach for autonomous vehicles, which must be processed in real-time.

### B. Learning-based Methods

Recently, deep learning methods have gained traction in academia. Notably, WeatherNet [7] stands out as a representative algorithm. It employs LiLaNet [15], initially designed for LiDAR semantic segmentation, to create training data. This aids in reducing point-cloud noise and producing weather-specific semantic segmentation networks. Due to challenging noise annotation, the Self-supervised LiDAR de-snowing (SLiDE) [8] framework tackles noise using self-supervised learning. LiSnowNet [9] introduces a unique filtering technique with discrete wavelet transform (DWT) and inverse discrete wavelet transform (IDWT) layers. For training, the Winter Adverse Driving Dataset (WADS) [3] and Canadian Adverse Driving Conditions (CADC) datasets [16] were utilized solely for unsupervised learning. 4DenoiseNet [10] presents a noise classification methodology that employs temporally continuous point cloud data and a semantic segmentation network. A significant volume of virtual synthetic data was used to train the semantic segmentation network. However, deep learning methods face a vulnerability – their effectiveness hinges on the quality and quantity of the dataset. Noise annotation difficulty leads to limited data. To counteract this, virtual data synthesis and self-learning were dual strategies. Moreover, the complexity of LiDAR sensors and environments challenges generalization. Despite demanding learning, applying to different sensors or environments remains intricate.

## III. Motivation

The principle of LiDAR is that the sensor emits laser pulses toward a target area, and when these pulses encounter an object, they are reflected back to the sensor. By measuring the time taken for the return pulse, the LiDAR sensor computes the distance of the object [17]. LiDAR sensors typically measure two scalar values of the received power at the point of the strongest response: distance $r$, which is calculated as time, and the intensity $\mu$, which is the magnitude. A simple mathematical model [17] of the received power of this single laser beam is shown in

$$P(r) = KG(r)\beta(r)T(r). \tag{1}$$

Here, $K$ is a constant representing the performance of LiDAR. $G(r)$ is defined as the geometrical element $G(r) = O(r)/r^2$,

and $O(r) \in [0, 1]$ is the degree of overlap between the beam and laser detector. This means that the proximity-detection performance converges to 1 if it is over a certain distance. $\beta(r)$ represents the effect of backscattering and is affected by particle size and laser wavelength. The last term represents transmittance, given by $T(r) = \exp(-2\alpha r)$, where $\alpha$ stands for the extinction coefficient, typically a small value. The transmittance $T(r)$ approaches 1 for near distances. The impulse response $R(r)$ of a single beam, which is the output of an actual LiDAR sensor, can be represented as

$$R(r) = P(r)\rho\delta(r - r_t). \tag{2}$$

Where $\rho$ is the reflectance of the object, and $\delta$ is the Dirac delta function.

Figure 2(a) illustrates the received power when particles like snowflakes and raindrops are present in a single LiDAR laser beam. In general, LiDAR sensors produce the strongest or latest detected intensity using (1). ② was detected to be present close to the LiDAR sensor and has a large overlap with the beam. In the case of the last return, the target ④ attenuated by the sum of ① to ③ is detected. If the combined particle intensities surpass the target intensity, even the last return fails to detect the target. In this way, despite mixed detection outcomes, small particles inherently exhibit minimal reflectance. Consequently, determining the actual reflectance of particles enables differentiation through comparison with the target reflectance.

We propose an algorithm to distinguish between particles and targets by extending features from a single beam to the aggregate response of multiple beams. Figure 2(b) demonstrates the algorithm's principle applied to the responses of multiple laser beams. Because only particles closer than the target are detected, the probability of being a particle increases in the order of the closest particles among the results of multiple beams.

## IV. RGOR: Reflectance and Geometrical Outlier Removal

### A. Restoring Reflectance

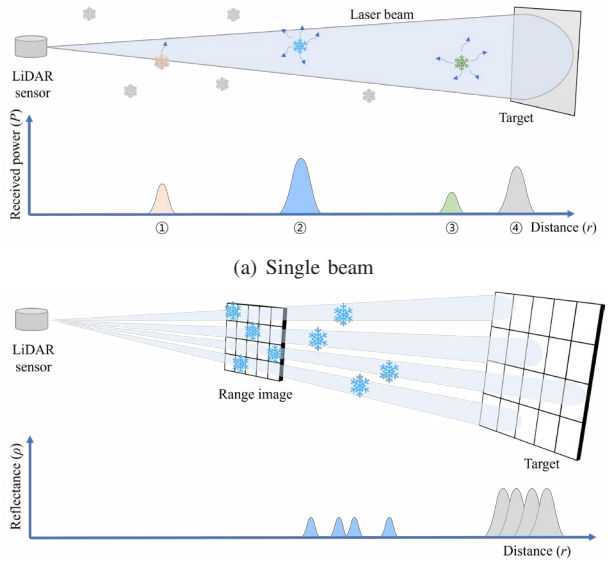
First, we explain how to obtain the reflectance of the original object from the intensity and distance information captured by LiDAR. As explained earlier in (1) and (2), the reflectance $\rho$ can be calculated as follows:

$$\rho = \frac{R(r)}{P(r)} = \frac{R(r)}{KG(r)\beta(r)T(r)}. \tag{3}$$

$R(r)$ represents the intensity measured at distance $r$. The scalar value of the measured intensity is denoted as $\mu$, and the distance $r$ is defined as $r = (x^2 + y^2 + z^2)^{1/2}$. Given the assumption of $K$ as a constant, $G(r) = 1/r^2$, and $T(r) = 1$ for a short distance, the reflectance simplifies as follows:

$$\rho \cong \frac{R(r)r^2}{K\beta(r)} = \frac{\mu r^2}{\gamma} = \frac{\mu\left(x^2 + y^2 + z^2\right)}{\gamma}. \tag{4}$$

Note, the measured intensity $\mu$ is typically a quantized positive value. It should be adjusted by adding a small constant and normalize it so that it is not ignored when the value is 0. $\gamma = K\beta(r)$ relates to the performance of the LiDAR sensor and the particle size and is an adaptive parameter that changes depending on the type of sensor and particles, such as fog, rain, and snow.




(a) Single beam

(b) Multiple beams

Fig. 2. Received power (intensities) and reflectance characteristics of LiDAR sensors: (a) received power of the single beam, (b) restored reflactances of multiple beams

The point cloud collected from the vehicle is constrained, particularly in the negative z-axis direction, extending only up to the road surface. Therefore, efficiency can be improved by introducing a weighting to the z-axis, as shown in the following equation

$$\hat{\rho} = \frac{\mu\left(x^2 + y^2 + \max(-\kappa z, z)^2\right)}{\gamma}. \tag{5}$$

Here, $\kappa$ is a weight, and it is set to a value approximately between 10 and 15, representing the ratio of the distance at which particles are detected (25m) to the height of the sensor above the road surface (2m). $\kappa$ is not a sensitive value, but it enhances the efficiency of road surface handling.

### B. Grouping and Sorting

This section outlines the process of grouping and sorting LiDAR responses to address multiple LiDAR beams simultaneously. The output of LiDAR encompasses azimuth, altitude, distance, and intensity information. Figure 3 illustrates sequentially aggregated LiDAR responses according to azimuth and elevation. This approach differentiates itself from the familiar range image technique [18] and requires minimal computations for generation, while also enabling response padding without gaps.

The simplest way to group LiDAR beams is by dividing them based on azimuth and altitude. In this study, aggregated images, as shown in Figure 3, are split into groups using 360

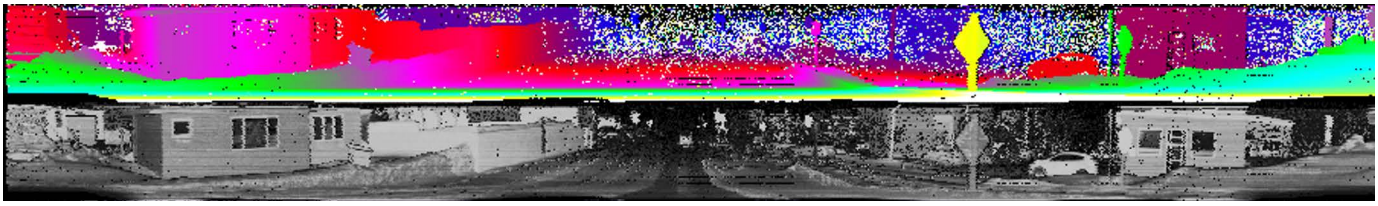


Fig. 3. Sequentially aggregated LiDAR responses for the first scene of WADS dataset sequence 11 [3]: These are clipped -90 to 90 degrees based on in front of the car. The fist row is distance and second row are intensity. For the best view, these are colorized, and histogram-equalized, respectively.

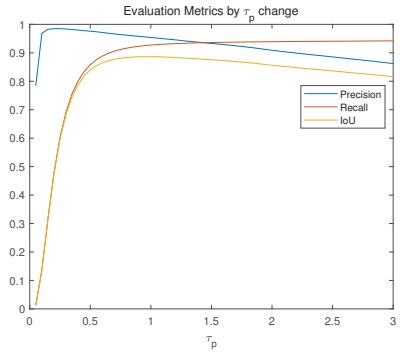



Fig. 4. Evaluation metrics analysis according to the variation in $\tau_p$. At $\tau_p = 1.45$, the precision, recall, and IoU of the training sequences reached 0.9350, 0.9356, and 0.8770, respectively.

azimuth and 16 elevation divisions. In the case of WADS, LiDAR has a resolution of $64 \times 1800$, so each group can contain up to 20 elements. The data within each group is sorted by distance to create $20 \times 16 \times 360$ data blocks.

The sorting process demanded the most computational time in the proposed algorithm. To address this, we employed a sorting network [19], which is a comparison network utilized for sorting fixed-sized inputs. It comprises wires and comparators, functioning without branches, making it suitable as a parallel-sorting algorithm. The comparator employs just `min` and `max` functions, enabling the definition of ascending and descending orders. The data structure for WADS allows for up to 20 points per block, and the optimal sorting network for 20 points requires 92 comparators [19].

### C. Filtering by Reflectance

As mentioned above, we have obtained a data block sorted by distance. Filtering is performed on each block. The block includes the distance from the sensor origin to the attached point, 3D coordinates, restored reflectance, and the index of the original point cloud for evaluation. We used two threshold values to determine whether it is a particle using the restored reflectance $\hat{\rho}$. These threshold values are defined as $\tau_p$ for obvious particles and $\tau_t$ for obvious targets. Their relationship is $\tau_p < \tau_t$.

If the reflectance within each block satisfies $\hat{\rho} < \tau_p$, it is classified as a particle. Once a particle is confirmed, nearby point clouds are also likely to be particles. This leads to a sequential evaluation where all point clouds satisfying $\hat{\rho} < \tau_p$ are classified as particles. However, during this sequential evaluation, if the condition $\tau_p \leq \hat{\rho} < \tau_t$ is met, it represents a boundary between particles and targets, necessitating a more thorough judgment. Further elaboration on this will be provided in the subsequent sections. Figure 4 illustrates the progression of evaluation metrics for the WADS dataset in response to varying $\tau_p$. The graph demonstrates comparable performance to the current state-of-the-art (SOTA) through sequential single thresholding of $\tau_p$.

### D. Filtering by Geometrical Information

In order to address the ambiguous classification of particles and targets, a principal component analysis (PCA) method is used to evaluate the geometric information of neighboring points [20]. PCA is using singular value decomposition (SVD) to analyze covariance characteristics. The covariance matrix is symmetric, and the singular values of the symmetric matrix coincide with its eigenvalues. The eigenvalue of a $3 \times 3$ symmetric matrix can be efficiently obtained in a closed form [21]. Figure 5 visually represents the relationship between the point distribution and SVD. The singular values, denoted as $\lambda_1$, $\lambda_2$, and $\lambda_3$, follow the order $\lambda_1 \leq \lambda_2 \leq \lambda_3$. Among the eigenvectors ($\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$), $\mathbf{v}_1$ corresponds to the surface normal, whereas $\mathbf{v}_3$ indicates the direction in which the data points are located. To compute SVD, it is necessary to gather the 3D coordinates of neighboring point clouds. For simplification, when the distance to the target point is $r$, all neighboring blocks within a distance $d$ encompass all points falling within that range. Since each block is already sorted by distance, the binary search method [14] is employed. Consequently, the actual boundary of the points forms a scalloped area, as illustrated in Figure 5. Given that the horizontal ratio of a block is $1^\circ$, the maximum distance between adjacent points can be determined as $d = r \arctan(1.5^\circ)$.

If the count of acquired neighboring points falls below the threshold value $\tau_c$, which is insufficient for conducting the SVD evaluation, these points are categorized as sparse and classified as particles. The measure of curvature used for shape evaluation, denoted as

$$\nu(\mathbf{p}_j) = \frac{\lambda_1}{\lambda_1 + \lambda_2 + \lambda_3}, \tag{6}$$

is derived from the singular value calculated through this process. Smaller values of (6) indicate flatter or more linear surfaces. When the value of the expression in (6) exceeds a certain threshold ($\nu(\mathbf{p}_j) > \tau_\nu$), points are identified as particles due to their scattering in 3D space. Additionally,

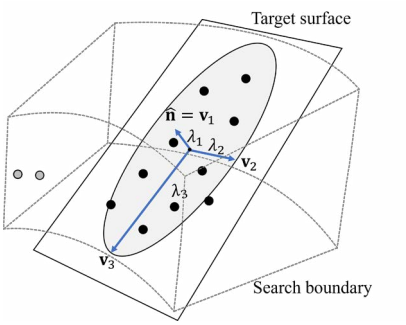



Fig. 5. Geometric filtering based on principal component analysis (PCA)

the angular difference between the largest eigenvector $\mathbf{v}_3$ (representing the direction of the target) and the point $\mathbf{p}_j$ can be determined using the cross product denoted as

$$\eta(\mathbf{p}_j) = \left\| \frac{\mathbf{p}_j}{\|\mathbf{p}_j\|} \times \mathbf{v}_3 \right\|. \tag{7}$$

If $\eta(\mathbf{p}j) < \tau_\eta$, the points are considered particles situated along the axis of the laser beam, as illustrated in Figure 2. Geometric filtering prevents the removal of objects with very low reflectivity, such as black vehicles or glass, enhancing accuracy (Figure 1).

## V. Experimental Results

### A. Evaluation Environment

The proposed RGOR algorithm was implemented using the C++ language with SIMD operations and validated on an Intel Xeon E-2286M mobile processor and NVIDIA RTX A5000 GPU environment, the same to our autonomous driving vehicle system [18]. To validate the algorithm, we evaluated it using the winter adverse driving dataset (WADS), which contains point-wise annotations for snowflakes, and synthetic virtual weather environments generated with weather simulators to assess various weather conditions. For evaluation, we used precision $\frac{TP}{TP+FP}$, recall $\frac{TP}{TP+FN}$, and intersection-over-union (IoU) $\frac{TP}{TP+FP+FN}$, which are commonly used in binary classification problems. Here, $TP$, $FP$, and $FN$ represent true positives, false positives, and false negatives, respectively. Precision signifies the ratio of correctly detected particles to all detected particles, recall indicates the ratio of detected particles to all particles that should be detected, and IoU measures the overlap between detected and actual particles. We divided the dataset in half for train and test data, then used the simplex method [14] to extract optimal parameters from the train data.

### B. Quantitative Evaluation

First, we used the winter adverse driving dataset (WADS) [3] to validate the algorithm. To the best of our knowledge, WADS is the only publicly available dataset that is point-wise labeled with particles, such as snow. WADS has 22 classes, the same as SemanticKITTI [25], with additional labels for `active-snow` and `accumulated-snow`. We defined `active-snow` as particles, which are outliers. Furthermore, we used weather simulators to authenticate diverse weather scenarios. The simulators employed fog_sim [22] for fog, snow_sim [24] for snow, and LISA [23] for rainy conditions.

Table I compares the evaluation metrics of existing state-of-the-art algorithms. Furthermore, the results have been visualized in Figure 6. LiDAR in autonomous vehicles typically operates at 10Hz, so all autonomous driving processes must be completed within 100ms. As evident from the results, statistical methods like DROR and DSOR exhibit high recall but remarkably low precision, leading to the removal of non-particle targets and proving impractical for real-time application due to slowness. LiSnowNet and 4DenoiseNet, employing deep learning approaches, demonstrate high performance in terms of precision, recall, and execution time, showcasing their applicability for real autonomous vehicles. However, LiSnowNet is trained on WADS, and 4DenoiseNet is trained on snow_sim data, leading to significant performance degradation when applied to different datasets. Our proposed approach demonstrates performance comparable to deep learning-based methods even when using only reflectance, and when incorporating geometric information, it shows better results in some cases. Notably, the execution speed is 3 to 10 times faster. Moreover, as it is not a training-based method, it effectively adapts to unseen data and other sensors.

## VI. Conclusions

In this study, a technique is proposed to effectively remove the noise caused by particles such as fog, rain, and snow that float in the air and act as noise in LiDAR sensors. This technique classifies the particles and the removes noise based on the fact that the particles have extremely little reflection compared to the target and are detected sequentially from the nearest particles. The proposed algorithm suggests a method to restore the reflection from LiDAR sensor input and sequentially classify the particles. Additionally, a method is proposed to increase the classification accuracy by using the distribution of neighboring points when classification is difficult with only reflection. The proposed technique was experimentally verified in various environments. Future research is needed on methods to automatically apply adaptive parameters, and it is suggested that there is value in researching deep-learning methods that use the restored reflectance proposed in this study as a feature in situations where a higher performance is required.

## References

[1] R. Roriz, J. Cabral, and T. Gomes, "Automotive lidar technology: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6282–6297, 2021.

[2] 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.

[3] 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.

[4] J.-I. Park, J. Park, and K.-S. Kim, "Fast and accurate desnowing algorithm for lidar point clouds," *IEEE Access*, vol. 8, pp. 160 202–160 212, 2020.

[5] R. Roriz, A. Campos, S. Pinto, and T. Gomes, "Dior: A hardware-assisted weather denoising solution for lidar point clouds," *IEEE Sensors Journal*, vol. 22, no. 2, pp. 1621–1628, 2021.

TABLE I
QUANTITATIVE EVALUATION

| Method | WADS [3] | | | Fog sim. [22] | | | Rain sim. [23] | | | Snow sim. [24] | | | Runtime [ms] |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Prec. | Recall | IoU | Prec. | Recall | IoU | Prec. | Recall | IoU | Prec. | Recall | IoU | |
| DROR [2] | 0.783 | 0.951 | 0.716 | 0.760 | 0.984 | 0.758 | 0.877 | 0.964 | 0.891 | 0.620 | 0.922 | 0.445 | 1031.823 |
| DSOR [3] | 0.531 | **0.981** | 0.524 | 0.710 | 0.981 | 0.771 | 0.885 | 0.944 | 0.877 | 0.573 | 0.952 | 0.432 | 883.712 |
| LiSnowNet [9] | 0.925 | 0.950 | 0.881 | 0.875 | 0.912 | 0.854 | 0.943 | 0.957 | 0.954 | 0.892 | 0.902 | 0.891 | 7.634 |
| 4DenoiseNet [10] | 0.917 | 0.977 | 0.901 | 0.951 | 0.965 | 0.947 | **0.989** | 0.990 | **0.990** | 0.985 | **0.983** | **0.976** | 44.301 |
| RGOR IV.C | 0.903 | 0.974 | 0.882 | 0.934 | 0.964 | 0.923 | 0.956 | 0.973 | 0.964 | 0.922 | 0.971 | 0.910 | **1.210** |
| RGOR IV.C+IV.D | **0.927** | 0.969 | **0.911** | **0.967** | **0.985** | **0.954** | 0.987 | **0.991** | 0.985 | **0.987** | 0.978 | 0.966 | 1.855 |

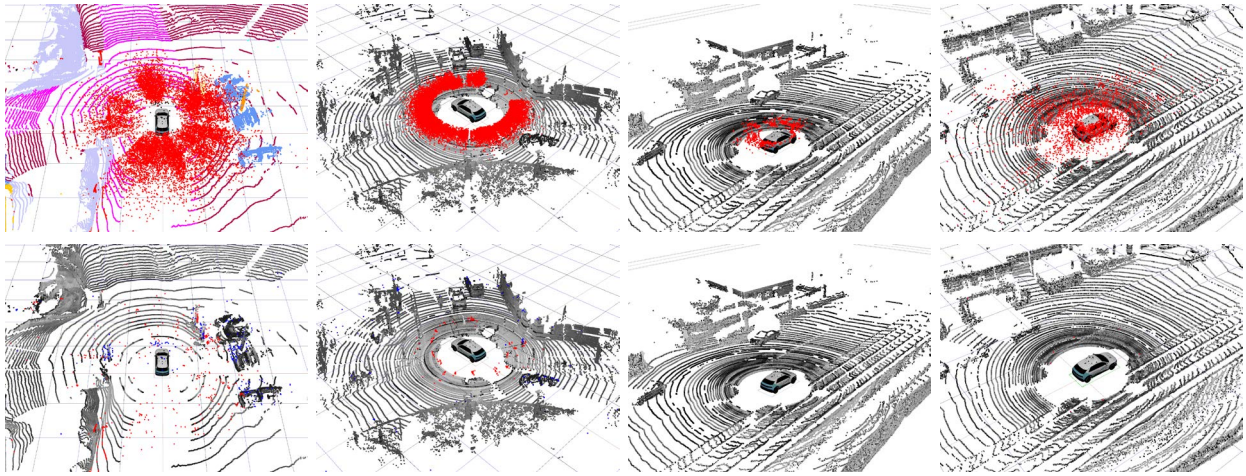


Fig. 6. Results from various adverse weather datasets: The first column represents WADS, while the second, third, and fourth columns correspond to data synthesized using the sim_fog, LISA, and sim_snow simulators, respectively. The labels of the datasets are color-coded in the top row. WADS contains various labels besides particles. Labeling particles is challenging, leading to numerous mislabeled particle data. The lower row displays the outcomes of the proposed algorithm. Red points indicate particles that were not removed, while blue points represent points that were incorrectly removed.


[6] 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.

[7] 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.

[8] G. Bae, B. Kim, S. Ahn, J. Min, and I. Shim, "Slide: Self-supervised lidar de-snowing through reconstruction difficulty," in *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIX*. Springer, 2022, pp. 283–300.

[9] 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.

[10] 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, 2023.

[11] J. Kang, S.-J. Han, N. Kim, and K.-W. Min, "Etli: Efficiently annotated traffic lidar dataset using incremental and suggestive annotation," *ETRI Journal*, vol. 43, no. 4, pp. 630–639, 2021.

[12] D. Choi, S.-J. Han, K.-W. Min, and J. Choi, "Pathgan: Local path planning with attentive generative adversarial networks," *ETRI Journal*, vol. 44, no. 6, pp. 1004–1019, 2022.

[13] 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.

[14] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, 2007.

[15] 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.

[16] M. Pitropov, D. E. Garcia, J. Rebello, M. Smart, C. Wang, K. Czarnecki, and S. Waslander, "Canadian adverse driving conditions dataset," *The International Journal of Robotics Research*, vol. 40, no. 4-5, pp. 681–690, 2021.

[17] M. Byeon and S. W. Yoon, "Analysis of automotive lidar sensor model considering scattering effects in regional rain environments," *IEEE Access*, vol. 8, pp. 102 669–102 679, 2020.

[18] S.-J. Han, J. Kang, K.-W. Min, and J. Choi, "Dilo: Direct light detection and ranging odometry based on spherical range images for autonomous driving," *ETRI journal*, vol. 43, no. 4, pp. 603–616, 2021.

[19] V. K. Valsalam and R. Miikkulainen, "Using symmetry and evolutionary search to minimize sorting networks," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 303–331, 2013.

[20] A. Nurunnabi, G. West, and D. Belton, "Outlier detection and robust normal-curvature estimation in mobile laser scanning 3d point cloud data," *Pattern Recognition*, vol. 48, no. 4, pp. 1404–1419, 2015.

[21] S. Sarabandi, A. Shabani, J. M. Porta, and F. Thomas, "On closed-form formulas for the 3-d nearest rotation matrix problem," *IEEE Transactions on Robotics*, vol. 36, no. 4, pp. 1333–1339, 2020.

[22] M. Hahner, C. Sakaridis, D. Dai, and L. Van Gool, "Fog Simulation on Real LiDAR Point Clouds for 3D Object Detection in Adverse Weather," in *IEEE International Conference on Computer Vision (ICCV)*, 2021.

[23] V. Kilic, D. Hegde, V. Sindagi, A. B. Cooper, M. A. Foster, and V. M. Patel, "Lidar light scattering augmentation (lisa): Physics-based simulation of adverse weather conditions for 3d object detection," *arXiv preprint arXiv:2107.07004*, 2021.

[24] 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 *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[25] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, "Towards 3d lidar-based semantic scene understanding of 3d point cloud sequences: The semantickitti dataset," *The International Journal on Robotics Research*, vol. 40, no. 8-9, pp. 959–967, 2021.