

# De-snowing Algorithm for Long Wavelength LiDAR

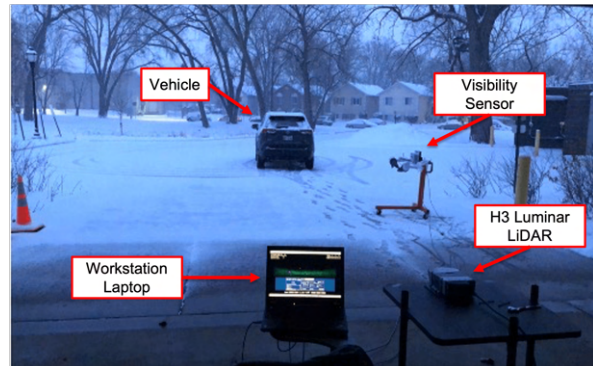
Bharat Jayaprakash<sup>†</sup>, Matthew Eagon<sup>†</sup>, Lu Zhan<sup>†</sup>, William F. Northrop<sup>†,\*</sup>

**Abstract**—Long wavelength light detection and ranging (LiDAR) sensors have emerged as an essential component for increasing the accuracy and range of perception of autonomous vehicles because they employ directed lasers with wavelengths longer than  $1\mu m$ . However, adverse weather conditions like fog, rain, and snow pose a major challenge. Longer wavelength lasers generally exhibit increased absorption by water-based ambient particles compared to those with shorter wavelengths. Suspended particles can also significantly scatter the incident laser beam and consequently reduce the sensor accuracy. Filtering out ambient particles is of paramount importance because an accurate representation of the surrounding environment is crucial to ensure safe navigation. Accurately distinguishing between snow particles, snow-covered surfaces, and other objects is challenging due to the varying intensities of the pulsed laser and the spatial orientation of highly reflective surfaces. Despite numerous research works addressing the problem of filtering out snow particles from LiDAR point clouds, there is a significant lack of documented research on long-wavelength LiDAR. Furthermore, many of the filters proposed in the literature are limited in terms of speed and accuracy which impedes their practical implementation in autonomous vehicles. In this paper, we propose a Network-Adjusted Reflectance Filter (NARF), a novel two-phase, convolutional neural network (CNN)-based filtering method for long-wavelength LiDAR that filters out 98.1% of snow particles. We use a custom experimental dataset obtained during a snow event to train and validate the proposed filter. The results shown here illustrate how the developed method outperforms the state-of-the-art geometric filters in terms of both speed and accuracy.

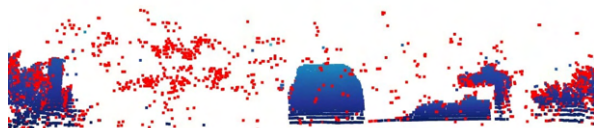
## I. INTRODUCTION

With the rapid evolution of the autonomous vehicle industry, the demand and capacity of Light Detection And Ranging (LiDAR) sensors have been advancing rapidly, while the equipment cost is dropping at a steady pace. The trend of massive LiDAR deployment in the near future is inevitable [1]. However, because LiDAR is a laser-based sensor, it is naturally subjected to particle light scattering physics. For vehicles, this means that ambient particles present during adverse weather, such as fog, snow, and rain droplets can potentially degrade accuracy. Contamination in LiDAR measurements due to snow particles remains a challenge.

It is crucial to tackle the degradation of LiDAR perception due to snow particles to achieve safer operation of automated vehicles. The understanding of automated LiDAR performance in snow is twofold. On one hand, historical studies have investigated the physical and optical properties of snow particles [2], which provides a guide for modeling an accurate perception environment for LiDAR. On the



(a) Experimental setup for custom dataset



(b) Point cloud view

Fig. 1: Spatial filters are typically less capable of detecting clustered snow

other hand, modeling LiDAR performance is indispensable to evaluating real-world measurement. Hence, the questions that need to be answered are: first, how to couple the snow statistics and the LiDAR performance model; and secondly, how to integrate the coupled models into the data processing.

Light scattering theory holds that scattering is affected by parameters including particle shape, size distribution, liquid water content, laser wavelength, etc. Mie theory deals with scattering where particle size is either of the same order as the laser wavelength or much larger. The larger the particles, the more the scattering diagram is peaked in the forward direction [3]. Automated LiDAR utilizes the particle back-scattered energy to differentiate observed objects and background noises. In [4], [5], rain droplet shape and size distribution are modeled, which are then used to characterize laser attenuation in a simulated environment. The correlation between LiDAR reflected intensity and rain intensity is evaluated through experimental studies [6]. Another experimental study quantifies the actual laser attenuation at different wavelengths (785 nm and 1550 nm) in fog [7]. Dust particles have a comparable size to snow particles. When dust is present, four levels of LiDAR return signal degradation are derived, observed, and characterized in [8].

To model the realistic LiDAR measurement in a snowy environment, we developed a reflectance filter based on the

<sup>†</sup>Department of Mechanical Engineering, University of Minnesota, MN

\*Corresponding author.

snow size distribution and Mie scattering theory. A neural network is developed to further improve the snow filter accuracy. The paper is organized as follows. The prevailing particle filters for LiDAR data are reviewed in Section II. The problem statement is formally defined in Section III. In Section IV-A, the experimental setup and data collection process are described. The rest of Section IV explains the construction of the components of the proposed Network-Adjusted Reflectance filter (NARF). Section V presents the experimental results using the NARF and compares the results to conventional filters.

## II. RELATED WORK

Conventional filters are available that can remove snow particles from LiDAR point clouds. These include spatial or distance-based filters. The Radius Outlier Removal (ROR) filter counts the number of neighbors for each point within a specified search radius and filters out points that have neighbors less than a specified minimum [9]. The ROR filter uses a k-d tree data structure to allow for efficient nearest neighbor search. The Statistical Outlier Removal (SOR) filter calculates the mean distance of each point to its nearby points (the number of neighbors is specified). The points in the point cloud that surpass the weighted sum of the mean distance and standard deviation are excluded [9]. The SOR also uses a k-d tree for efficient search operations. One of the primary drawbacks of the ROR and the SOR filters is that the filter parameters are fixed and not distance-dependent. This results in farther points having a higher probability of being classified as an outlier. To address this limitation, the authors of [10] developed the 4-parameter Dynamic Radius Outlier Removal (DROR) filter which implements a distance-dependent search radius for the nearest neighbor search. The farther the distance of the point, the greater the search radius of the filter. The authors reported that the filter exhibited precision and recall of greater than 90%. Subsequently, the authors of [11] developed the Low-Intensity Outlier Removal (LIOR) filter that leverages the intensity value of a point captured by the LiDAR sensor to classify points as snow and subsequently use a spatial filter to restore the few incorrectly classified snow points. Furthermore, the authors of [12] presented a de-snowing approach that first filtered out snow-caused noise using an intensity filter and then leveraged spatiotemporal information from the point clouds to restore the incorrectly filtered points.

One persistent drawback of spatial filters is their reliance on k-d trees for nearest neighbor searches, which can be computationally intensive. In the context of autonomous cars where real-time processing is crucial, de-snowing filters should be designed for high-speed execution. Spatial filters are also more likely to miss moving snow clusters (Figure 1b). Depending on the filter parameters, snow clusters that are farther than a particular distance may be falsely classified as hard targets. Moreover, these filters require empirical parameter tuning. A feature of the LIOR filter, discussed in Section V, is that the intensity model is developed specifically

for LiDAR of 865 nm wavelength and is unlikely to work for long wavelength LiDARs. Data-driven methods used for de-snowing LiDAR point clouds include WeatherNet [13] and LiSnowNet [14], both of which are constructed using convolutional neural networks (CNN). However, both these methodologies require projecting the point clouds onto a spherical coordinate system as range images before filtering, which can potentially be a contributing factor to slower real-time processing.

## III. PROBLEM DESCRIPTION

Within the context of de-snowing for long wavelength LiDAR outputs, a point cloud is defined as a set of 4-dimensional points  $\{P_i | i = 1, 2, \dots, n\}$ , where each point  $P_i \in \mathbb{R}^4$  contains the 3-dimensional Cartesian coordinates and the reflectance value of the point. The number of points in a point cloud,  $n$ , may vary depending on the complexity of the detected environment. Given an unordered set of points comprising a point cloud  $P$ , a snow filter should return a processed point cloud  $Q = \{Q_i | i = 1, 2, \dots, m\}$ , where  $Q_i \in \mathbb{R}^4$  are the non-snow points from  $P$ . The objective is to find  $Q$  with high accuracy and precision. To function in a practical setting, filters should ideally be input order invariant (or permutation invariant), and have low computational time complexity to enable fast real-time processing.

## IV. METHODOLOGY

### A. Data Collection and Annotation

A stationary experiment was conducted to study LiDAR performance during a natural snow event. The LiDAR sensor used in the work is a prototype product manufactured by Luminar Technologies Inc. with a 1.55  $\mu\text{m}$  wavelength. The sensor is capable of scanning  $120^\circ$  in the horizontal field of view, and  $30^\circ$  in the vertical direction. The system can acquire approximately  $880 \times 64$  data points at 10 Hz sampling frequency. As shown in Figure 1a, the LiDAR sensor was mounted on a portable table facing directly at the target vehicle which was approximately 10 meters away. Although the LiDAR ingress protection rating is IP67, it was placed indoors to avoid the accumulation of snow on the laser lens, which may significantly contaminate the measurements.

A visibility sensor was utilized to measure the environmental extinction coefficient with 1 Hz sampling frequency. For data quality control details of the LiDAR and visibility sensor, see [15]. A workstation laptop was used to perform data acquisition tasks including time synchronization, data logging, live data monitoring and sensor control.

The LiDAR sensor performed pre-programmed dual-scanning profiles and reported point clouds with 3-dimensional Cartesian coordinates and reflectance. The point cloud captured looks like a snapshot of the scene, as shown in Figure 7a. The reflectance value is referred to by the manufacturer as range-corrected directional reflectance (RCDR), which is defined as a ratio of the total received energy over

the total incident energy of each laser beam. The RCDR value is scaled to be one at any range if the reflector is a Lambertian (isotropic) target.

About 2837 LiDAR snapshots were collected during the snow event, among which two hundred consecutive snapshots of interest were selected for training and testing the neural network. To tackle the hurdle in labeling snow particles, we adopt a two-step approach to annotate the dataset. Firstly, a reflectance-based filter is utilized to initialize the detection of the snow particles. If RCDR is below or equal to 0.02, the LiDAR sample point is labeled as 1, otherwise is labeled as 0. The threshold of 0.02 is derived theoretically in the next section. The second step is to manually annotate the snow particles, particularly to remove the false positives on the hard targets and add the true negatives that are clearly snow particles.

### B. RCDR-based filtering

In [15], a formula is established to model the RCDR of 1.55  $\mu$  wavelength laser from ambient snow particle:

$$I \sim \frac{P_r}{P_i} \frac{L^2}{O(L)} = \left[ \frac{c_r}{2} A\eta \right] \times \beta_b \times e^{-2\beta_e} \quad (1)$$

where  $P_r$ ,  $P_i$  are the received and incident power respectively.  $L$  is the distance from receiving aperture to a certain snow particle. The term  $O(L)$  is the LiDAR geometric factor and the first term on the RHS is LiDAR system factor [16]. These two factors can be treated as constant once the LiDAR is deployed.  $\lambda$  is the laser wavelength of 1.55  $\mu$ m. Usually for a cluster of snow particles, back-scatter coefficient  $\beta_b$  and extinction coefficient  $\beta_e$  can be defined as:

$$\beta_b = \int_0^\infty Q_{bck}(x, m) \pi r^2 n(r) \quad (2)$$

$$\beta_e = \int_0^\infty Q_{ext}(x, m) \pi r^2 n(r) \quad (3)$$

where  $r$  is the radius of the snow particle if a sphere shape is assumed and  $n(r)$  is the number of particles with an equivalent radius of  $r$ .  $x$  is defined as a unit-less ratio  $x = \frac{2\pi r}{\lambda}$  and  $m$  is the refractive index of snow, which can be approximated as 1.33 [2].  $Q_{bck}$  and  $Q_{ext}$  are a function of snow particle size distribution. The analytical solution of  $Q_{bck}$  and  $Q_{ext}$  as a function of size parameter  $x$  can be computed from the software Mieplot [17]. It is noteworthy that the sharp and highly irregular ripple structure in Figure 2 originated in resonant electromagnetic modes of a sphere [18].

If the snow particle radius is known, it can be inserted into Figure 2a and 2b to interpolate the particle individual extinction coefficient and back-scatter coefficient. Then for an individual snow particle, the RCDR can be reduced to a Beer-Lambertian approach as:

$$I \sim Const. \times Q_{bck}(x, m) e^{-2Q_{ext}(x, m)} \quad (4)$$

To model the snow size distribution, we adopt the Gunn-Marshall distribution as [19]:

$$n(D_m) = n_m \exp(-\gamma_m D_m) \quad (5)$$

Noted that  $n_m$  and  $\gamma$  are defined by following results in [20]:

$$n_m = 2500 R_s^{-0.94} \quad (6)$$

$$\gamma = 22.9 R_s^{-0.54} \quad (7)$$

$R_s$  is the precipitation rate and  $D_m$  is the diameter of a water drop to which a snow particle melts. According to [19], the former can be interpolated by using the extinction coefficient as input. The visibility sensor measures the snow extinction coefficient at a fixed sampling spot. And assuming the homogeneity of snow distribution the precipitation rate of snow is estimated to be around 0.3 mm/hr.

In order to simulate the distribution of  $n(D_m)$ , we first sample snow particle diameter  $D_x$  from 0.01 mm to 1 mm uniformly. Based on the Equation5, an inverse transform sampling technique is performed.

$$r_{inv} = \frac{1}{-2\gamma} \log\left(\frac{D_x}{n_m}\right) \quad (8)$$

where  $r_{inv}$  is the simulated snow particle radius randomly sampled by following the Gunn-Marshall distribution. Insert  $r_{inv}$  back into Figure 2a and 2b to obtain  $Q_{ext}$  and  $Q_{bck}$  value (indicated as red dots). Then substitute  $Q_{ext}$  and  $Q_{bck}$  back to Equation4 to finally compute the simulated snow particle RCDR and histogram. The Constant in Equation4 can be set equal to one without losing generosity. Comparing the histogram of RCDR from simulation and the experimental data in Figure 3, a strong agreement is observed. More importantly, both simulated RCDR and experimental data present the same cut-off value at 0.02, which is then set to be the threshold for RCDR-based filtering.

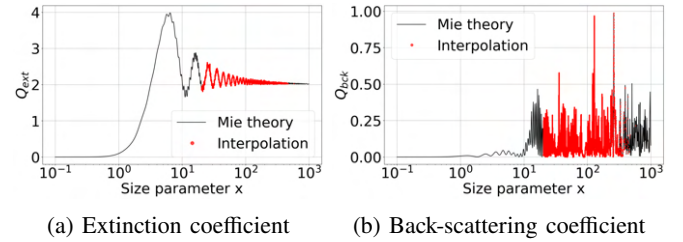


Fig. 2: Extinction and back-scattering coefficient of snow particle

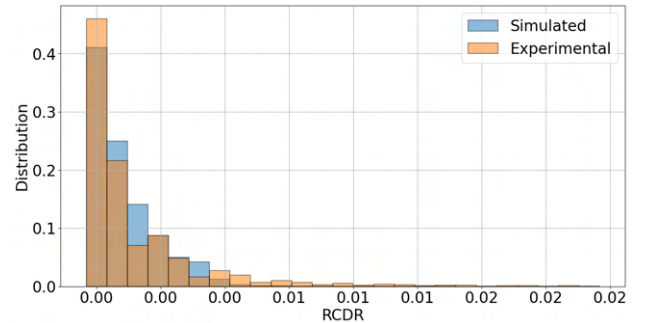


Fig. 3: Histogram of simulated and experimental RCDR values

It was observed that the RCDR filter exhibited varying classification capabilities based on the RCDR values of the points in the cloud. The filter has high accuracy above the

0.02 threshold, which means high confidence in predicting non-snow points. The relatively low accuracy of the RCDR filter below the threshold of 0.02 calls for a second filtering method specifically to re-classify the detected snow particles with the goal of correcting false positives.

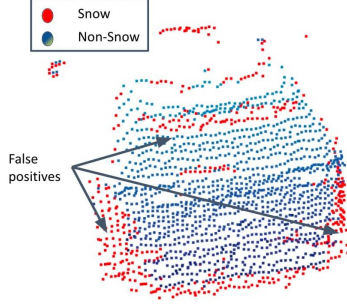


Fig. 4: Visualization of the RCDR filter's output. False positives were detected on the rear side of the vehicle due to the high angle of incidence of laser beams

Additionally, a notable observation from the output in Figure 4 is the nature of the false positives detected by the RCDR filter. Surfaces like the rear wheel arc of the vehicle are oriented in a way such that the angle between their normal and the line of sight from the lidar is high. Such spatial orientations result in the laser beams hitting the surface with high angles of incidence because of which the intensities of the beams returning to the sensor are low. This causes the RCDR filter to classify such points as snow. Hence, it is necessary to use a secondary filter that restores the false positive points back into the original point cloud.

### C. Restoring False Negatives

Inspired by the accuracy of WeatherNet [13], we implement a CNN-based framework called RestoreNet (Figure 5) that restores false positives detected by the RCDR filter back into the filtered point cloud.

Carrying forward the definitions from the authors of [21], a set of points in  $\mathbb{R}^n$  that make up a point cloud has the following important properties:

- 1) The points are unordered. Hence, any class of filter that is applied on such point clouds needs to be input order invariant, i.e., a change in the order of the input must not result in a different point-wise output.
- 2) Neighboring points form meaningful local representations. Points that are close to each other are more likely to share geometric and optical properties than points that are far from each other.

Considering the two properties, it was necessary to transform the input point cloud to a canonical order before allowing the network to learn local representations. The input transform block in Figure 5 predicts a Cartesian transformation matrix (labeled as T-Matrix in Figure 5a) that transforms the raw input cloud into a canonical order. The transformed point cloud is then fed into a series of RestoreBlocks that

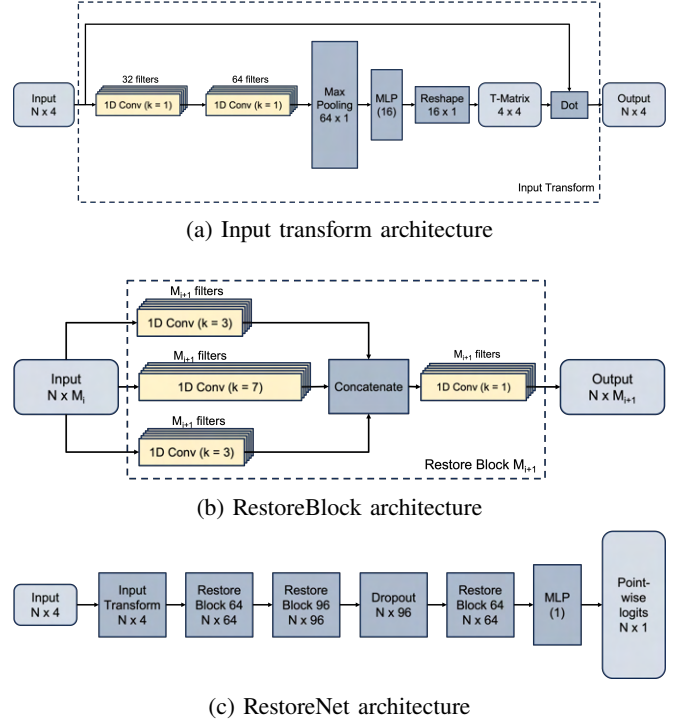


Fig. 5: Illustration of the individual components of RestoreNet inspired by WeatherNet architecture [13]

form the building blocks of RestoreNet. The RestoreBlocks consist of convolutional filters and transform input point sets from size  $N \times M_i$  to  $N \times M_{i+1}$ . The architectures of the RestoreBlock and RestoreNet are case-specific variants of LiLaBlock and LilaNet [22]. The complexities of the respective architectures are reduced given that the purpose of the RestoreNet is to correctly reclassify false positive snow particles. A dropout regularization layer is added after the second RestoreBlock to prevent overfitting. The output of the RestoreNet is a set of logits  $\{S_i | i = 1, 2, \dots, N\}$ , where  $S_i \in \mathbb{R}^{N \times 1}$  represents the probability that the  $i_{th}$  point being a snow particle.

### D. RestoreNet training

Network hyperparameters were optimized using grid search with 3-fold cross-validation. The dropout layer probability was set to 0.3. The batch size  $b$  for training was set to 10. The learning rate was scheduled with an exponential decay rate of 0.9 for every 150 epochs. The Adam solver is used for the training process, with the suggested default values  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\varepsilon = 10^{-8}$  [23]. The dataset is split into a training set (75%) and a test set (25%). The loss function used for forward pass evaluation was binary cross entropy as the network is a point-wise binary classifier. Since the dataset consists of point clouds with stationary objects and non-stationary snow particles, we implemented a data augmentation technique in which each point cloud was subjected to a 3-dimensional axis translation to prevent overfitting in the form of memorization of cartesian coordinates of specific objects. In this translation, the entire point cloud



was shifted in all directions by a randomly generated value with a predefined upper limit.

$$\begin{aligned} P' &= P + d \\ d &\sim U(-D, D) \end{aligned} \quad (9)$$

The translation can be mathematically represented by Equation 9, where  $P$  is the original point cloud,  $P'$  is the augmented point cloud, and  $d$  is a random number generated from a uniform distribution with upper and lower bounds  $-D$  and  $D$  respectively. The value of  $D$  for the training process was set to 5 meters. Additionally, the raw point clouds were padded with dummy points positioned at the origin in order to get all the point clouds to have the same number of points, which would ultimately aid in the training process. Enough number of dummy points were added to each point cloud such that it contains 22000 points.

---

**Algorithm 1:** Network-Adjusted Reflectance Filter

---

**Input:**  $P = \{[P_{ix}, P_{iy}, P_{iz}, P_{ir}] | i = 1, 2, \dots, N\}$ ,  
 $r_{thres}$   
**Output:**  $S, NS$

```

1  $NS \leftarrow \emptyset$  // Points classified as non-snow
2  $S \leftarrow \emptyset$  // Points classified as snow
3 for  $p$  in  $P$  do
4   if  $P_{ir} \leq r_{thres}$  then
5      $S \leftarrow S \cup \{p\}$ 
6   else
7      $NS \leftarrow NS \cup \{p\}$ 
8  $logits \leftarrow \text{Run RestoreNet}(P)$ 
9 for  $p$  in  $S$  do
10  if  $logits[p] < 0.5$  then
11     $S \leftarrow S \setminus \{p\}$  // Remove from S
12     $NS \leftarrow NS \cup \{p\}$  // Restore to NS

```

---

## V. RESULTS AND DISCUSSION

As described in Section IV-A, we obtained a custom dataset from an outdoor static test during a snowstorm event. A threshold-based RCDR filter was implemented on the dataset first. RestoreNet was trained on the dataset to restore false positives back into the filtered point cloud with hyperparameters as discussed in Section IV-D. Figure 6 shows the bin-wise accuracy of RestoreNet, where each bin is 0.025 units long in the output probability space and the bins span the whole output range:  $[0.0, 1.0]$ . It was observed that RestoreNet was least confident when it classified a point with a probability in the ranges  $[0.4, 0.475)$  and  $[0.525, 0.625)$ . However, it was observed that predictions in these ranges were predominantly for points that are not snow particles, for which the RCDR filter exhibited high accuracy. Figure 8 provides a visualization of the performance of the NARF with a visual comparison with the output of the RCDR filter. It can be observed that the false positives that were actually hard targets on the rear of the vehicle are restored back into the point cloud.

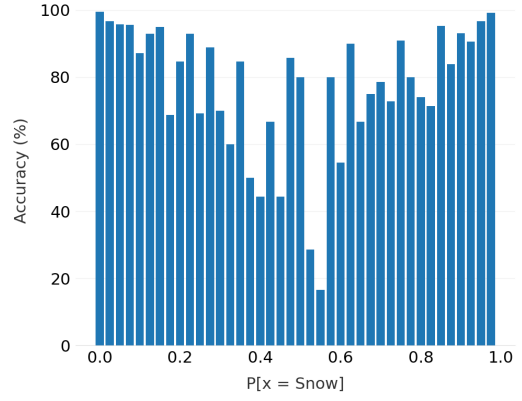


Fig. 6: Accuracy of RestoreNet for output bins

TABLE I: Performance Metrics Comparison

Metric	DROR [10]	LIOR [11]	RCDR	NARF ( <i>proposed</i> )
Accuracy	0.6910	0.9444	0.9517	0.9879
Precision	0.9105	0.8740	0.8767	0.9814
Recall	0.0421	0.9660	0.9887	0.9808
IOU	0.0419	0.8480	0.8680	0.9629
TNR	0.9980	0.9341	0.9342	0.9912
FPR	0.0019	0.1391	0.0658	0.0088
FNR	0.9578	0.0160	0.0112	0.0192
MFT	0.5946	0.3769	0.0005	0.1957

Input order invariance is an important constraint for the NARF. An input transform block, as shown in Figure 5a, was used to transform the input points to a canonical form. To test the robustness of the algorithm to changes in input point order, a random point cloud from the test dataset was chosen. The point cloud was randomly shuffled to four progressively more extensive degrees: initially, 25% of the points were shuffled, followed by 50%, followed by 75%, and finally, 100%. For each of these scenarios, and for the non-shuffled case, the accuracy of the NARF was calculated and is reported in Table II. The accuracy dips slightly as the extent of shuffling increases, and the accuracy falls by 3.07% on average when all points are shuffled, showing that the algorithm is fairly robust to changes in input point order.

TABLE II: Effect of shuffled input

Shuffle %	0%	25%	50%	75%	100%
Accuracy	0.9886	0.9809	0.9740	0.9675	0.9579

### A. Performance Metrics

Three preliminary performance metrics, four key performance metrics, and a time complexity metric are used for the purpose of comparing the different de-snowing methods. The preliminary performance metrics used are the True Negative Rate (TNR), the False Positive Rate (FPR), and the False Negative Rate (FNR). The key performance metrics that are used for the purpose of comparison are accuracy, precision, recall, and intersection-over-union (IoU), as formulated in

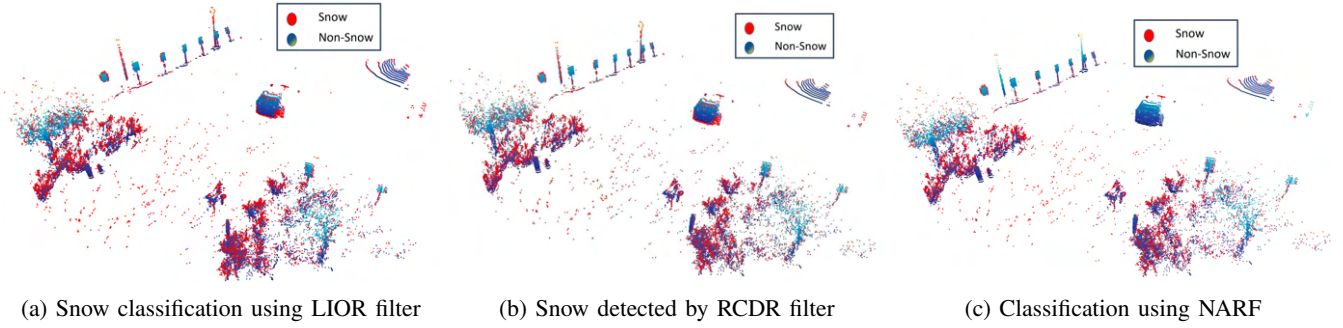


Fig. 7: Comparison of snow particle classification using different methods

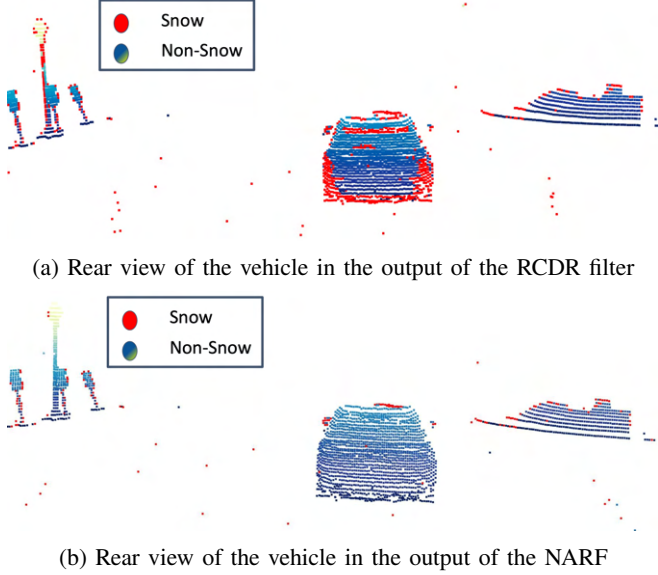


Fig. 8: Visualization of the capability of the RestoreNet in restoring false positives back into the point cloud

Equation 10.

$$\begin{aligned}
 Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\
 IoU &= \frac{TP}{TP + FP + FN} \\
 Precision &= \frac{TP}{TP + FP} \\
 Recall &= \frac{TP}{TP + FN}, \quad TNR = \frac{TN}{TN + FP} \\
 FPR &= \frac{FP}{FP + TN}, \quad FNR = \frac{FN}{FN + TP}
 \end{aligned} \tag{10}$$

Additionally, to compare the time complexity of the filtering methodologies, the Mean Filtering Time (MFT, in seconds) is used. MFT is the average execution time for the filtering process of the point clouds using each of the filtering methods. The compared filtering methodologies are implemented on the whole dataset.

### B. Comparison with state-of-the-art

The NARF, the baseline RCDR filter, and two state-of-the-art spatial filters (Low-Intensity Outlier Removal filter [11] and Dynamic Radius Outlier Removal filter [10]) were

compared with each other. Table I shows the notable distinctions in the de-snowing effectiveness of each of the compared filters. The parameters of the DROR and the LIOR filters were tuned using the grid search technique. The standalone RCDR filter and the NARF outperformed the DROR and the LIOR filters across multiple key metrics. The DROR filter exhibited a lower FPR and higher TNR than all the other filters. Therefore, it had higher precision than both the LIOR and the RCDR filter as it was able to extensively minimize the number of FPs. However, the DROR performed poorly in correctly detecting snow particles and hence, exhibited low recall and had a high FNR. Comparing the overall accuracy, the RCDR filter (95.17%) outperformed the LIOR filter (94.4%) and the DROR filter (69.10%), while the NARF (98.79%) fared the best. As expected, the NARF had a lower FPR and a higher TNR than the RCDR filter due to the corrective capabilities of RestoreNet. Consequently, it exhibits a 10.47% and 7.09% increase in precision from the RCDR filter and the DROR filter respectively. Finally, the NARF demonstrates a significantly lower MFT compared to DROR and LIOR. The RCDR has a significantly lower MFT than the NARF as RCDR filtering is the first step of the NARF. This reduction in processing time underscores the high-speed capabilities of the NARF, which is a critical factor for ensuring the real-time functionality of filters in autonomous vehicle applications.

## VI. CONCLUSIONS

In this work, we presented a novel two-step method for filtering snow from point cloud data collected using long wavelength LiDAR. Experimental results show the reflectance-based filter has a high true positive rate, while a CNN-based RestoreNet model is used to effectively identify and correct false positives. Together, the proposed NARF method filters snow while conserving points from solid objects, improving upon existing methods.

## ACKNOWLEDGMENT

This research was funded by the Minnesota Department of Transportation (MnDOT) under project number 00081748 and would not have been possible without the collaboration of Luminar Technologies Inc. and Vision System Intelligence (VSI) labs.

## REFERENCES

- [1] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, 2020.
- [2] S. Warren, "Optical properties of ice and snow," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 377, 04 2019.
- [3] "Light scattering by small particles. by h. c. van de hulst. new york (john wiley and sons), london (chapman and hall), 1957. pp. xiii, 470; 103 figs.; 46 tables. 96s," *Quarterly Journal of the Royal Meteorological Society*, vol. 84, no. 360, pp. 198–199, 1958.
- [4] M. Byeon and S. W. Yoon, "Analysis of automotive lidar sensor model considering scattering effects in regional rain environments," *IEEE Access*, vol. 8, pp. 102669–102679, 2020.
- [5] M. Berk, M. Dura, J. V. Rivero, O. Schubert, H.-M. Kroll, B. Buschardt, and D. Straub, "A stochastic physical simulation framework to quantify the effect of rainfall on automotive lidar," *SAE International Journal of Advances and Current Practices in Mobility*, vol. 1, no. 2019-01-0134, pp. 531–538, 2019.
- [6] S. Hasirlioglu, I. Doric, C. Lauerer, and T. Brandmeier, "Modeling and simulation of rain for the test of automotive sensor systems," in *IEEE Intelligent Vehicles Symposium (IV)*, pp. 286–291, IEEE, 2016.
- [7] I. I. Kim, B. McArthur, and E. J. Korevaar, "Comparison of laser beam propagation at 785 nm and 1550 nm in fog and haze for optical wireless communications," vol. 4214, pp. 26–37, 2001.
- [8] T. G. Phillips, N. Guenther, and P. R. McAree, "When the dust settles: The four behaviors of lidar in the presence of fine airborne particulates," *Journal of field robotics*, vol. 34, no. 5, pp. 985–1009, 2017.
- [9] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, (Shanghai, China), IEEE, May 9–13 2011.
- [10] 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)*, pp. 254–261, 2018.
- [11] 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.
- [12] 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)*, (Philadelphia, PA, USA), pp. 2359–2365, IEEE, May 2022.
- [13] 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, pp. 2514–2521, apr 2020.
- [14] M.-Y. Yu, R. Vasudevan, and M. Johnson-Roberson, "Lisnownet: Real-time snow removal for lidar point cloud," 2022.
- [15] L. Zhan and W. F. Northrop, "Impact of fog particles on 1.55  $\mu\text{m}$  automotive lidar sensor performance: An experimental study in an enclosed chamber," in *SAE technical paper series.*, vol. 1 of *SAE WCX Digital Summit*, (Warrendale, PA, USA), Society of Automotive Engineers.
- [16] C. Weitkamp, "Lidar, range-resolved optical remote sensing of the atmosphere," Berlin: Springer, 2005., vol. 102, 01 2005.
- [17] P. LAVEN, "Mieplot (a computer program for scattering of light from a sphere using mie theory & the debye series)," <http://www.philiplaven.com/mieplot.htm>, 2011.
- [18] C. F. Bohren and D. R. Huffman, "Absorption and scattering of light by small particles," Apr 1998.
- [19] G. Koh, "Physical and optical properties of falling snow," 1989.
- [20] R. S. Sekhon and R. C. Srivastava, "Snow size spectra and radar reflectivity," *Journal of Atmospheric Sciences*, vol. 27, no. 2, pp. 299–307, 1970.
- [21] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," 2017.
- [22] F. Piewak, P. Pinggera, M. Schäfer, D. Peter, B. Schwarz, N. Schneider, M. Enzweiler, D. Pfeiffer, and J. Zöllner, "Boosting LiDAR-Based Semantic Labeling by Cross-modal Training Data Generation," Munich, Germany, 2018, Proceedings, Part VI, pp. 497–513. 01 2019.
- [23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.