Seppänen, Alvari; Ojala, Risto; Tammi, Kari

Self-supervised multi-echo point cloud denoising in snowfall

Published in:
Pattern Recognition Letters

DOI:
10.1016/j.patrec.2024.07.007

Published: 01/09/2024

Document Version
Publisher's PDF, also known as Version of record

Pattern Recognition Letters 185 (2024) 52–58

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Self-supervised multi-echo point cloud denoising in snowfall

Alvari Seppänen *, Risto Ojala, Kari Tammi

Aalto University, Otakaari 4, Espoo, 02150, Uusimaa, Finland

ARTICLE INFO

Editor: Lorenzo Baraldi

MSC:
68T05
68T10
68T45
62M45

Keywords:
Self-supervised learning
Denoising
Snowfall
Point cloud

ABSTRACT

Snowfall can cause noise to light detection and ranging (LiDAR) data. This is a problem since it is used in many outdoor applications, e.g., autonomous driving. We propose the task of multi-echo denoising, where the goal is to pick the echo that represents the objects of interest and discard other echoes. Thus, the idea is to pick points from alternative echoes unavailable in standard strongest echo point clouds. Intuitively, we are trying to see through the snowfall. We propose a novel self-supervised deep learning method and the characteristics similarity regularization to achieve this goal. The characteristics similarity regularization utilizes noise characteristics to increase performance. The experiments with a real-world multi-echo snowfall dataset prove the efficacy of multi-echo denoising and superior performance to the baseline. Moreover, based on extensive experiments on a semi-synthetic dataset, our method achieves superior performance compared to the state-of-the-art in self-supervised snowfall denoising. Our work enables more reliable point cloud acquisition in snowfall. The code is available at https://github.com/alvariseppanen/SMEDen.

1. Introduction

The impact of snowfall conditions on light detection and ranging (LiDAR) sensor data can be enormous. Airborne snowflakes [1] cause unwanted reflections of the LIDAR signal, which causes cluttered and missing points. Different weather conditions cause different types of noise. This paper focuses on snowfall because it causes the most severe noise. The noise is a critical issue as point clouds are typically used for determining the accessible volume of the environment, for instance, in obstacle detection methods. Furthermore, it affects other downstream perception algorithms, namely object detection [2], which is a vital component, e.g., in automated driving and driving assistance systems. Moreover, accident rates for human drivers are notably higher in adverse weather conditions, as reported by the European Commission [3] and the US Department of Transportation [4]. Therefore, reliable perception data is crucial in such conditions.

Previous LiDAR snowfall and adverse weather denoising work has mainly used single-echo point clouds, which lack information due to noise occluding parts of the objects. This work proposes using multi-echo point clouds, picking the echo representing the objects of interest, and discarding the echoes representing airborne snowflakes (noise) or irrelevant artifacts caused by refractions (Fig. 1). Using raw photon count histograms is beneficial when denoising LiDAR data in fog [5]. However, they are unavailable in most of the shelf LiDARs. Therefore, we use the point cloud data format.

Our goal is to utilize information that is unavailable in single-echo approaches. That is, airborne particles might occlude objects in single-echo approaches, but in multi-echo approaches, they are visible as alternative echoes and thus provide valuable information. We present a novel self-supervised approach **SMED**en (**S**elf-supervised-**M**ulti-**E**cho-**D**enoising) to achieve this goal. Finally, we propose a characteristics similarity regularization method to improve convergence and accuracy, which utilizes typical snowfall noise characteristics, i.e., intensity and sparsity, to increase performance.

The contributions of this paper are summarized as follows:

- We propose and formulate the task of multi-echo denoising for LiDAR in snowfall.
- We propose a novel self-supervised approach SMEDen that learns this task, motivated by the fact that point-wise labels are laborious to obtain, especially for multi-echo point clouds. The approach uses novel neighborhood correlation and blind spot methods.
- We propose the characteristics similarity regularization for LiDAR snowfall denoising to boost the performance of self-supervised algorithms.

2. Related work

There are several snowfall and adverse weather denoising methods in the literature. Charron et al. [6] presented an algorithm called

* Corresponding author.
E-mail addresses: alvari.seppanen@aalto.fi (A. Seppänen), risto.j.ojala@aalto.fi (R. Ojala), kari.tammi@aalto.fi (K. Tammi).

https://doi.org/10.1016/j.patrec.2024.07.007
Received 21 September 2023; Received in revised form 1 July 2024; Accepted 5 July 2024
Available online 10 July 2024

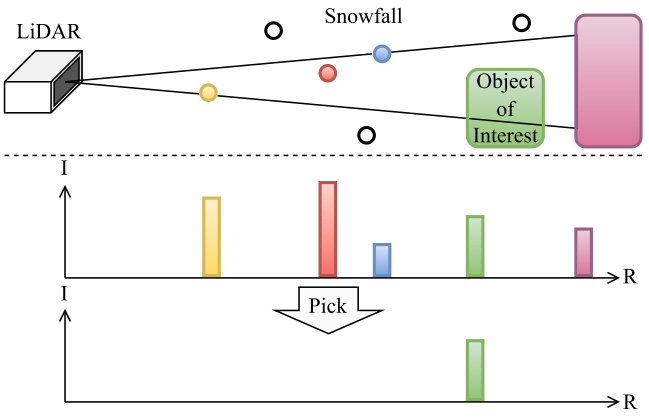

Fig. 1. In the concept of multi-echo denoising, multiple echoes are acquired for a single emitted pulse, and the echo caused by the object of interest is picked. Generally, only the strongest echo (red) is available in single-echo approaches. Thus, crucial information about the object of interest can be lost. I and R denote intensity and range, respectively.

dynamic radius outlier removal (DROR). It removes points whose distance from the neighboring points exceeds a threshold. Park et al. [7] discovered that snowfall-caused points typically have lower intensity than valid points. They developed low-intensity outlier removal (LIOR), which removes points based on the intensity value and DROR output.

Deep learning methods targeted explicitly to the adverse weather denoising task have also been presented in the literature, and they are typically projection-based as the projection provides a good trade-off between accuracy and computational cost. Heinzler et al. [8] proposed the WeatherNet architecture for denoising fog and rain. Seppänen et al. introduced 4DenoiseNet [9], which utilizes spatiotemporal information from consecutive point clouds. Bae et al. [10] developed SLiDE, which can be trained self-supervised by reconstruction difficulty. SLiDE can also be used to improve the labeling efficiency of supervised training for de-snowing. Yu et al. [11] used the fast Fourier and the discrete wavelet transforms to construct a loss function for training a network for snow segmentation. Our work is inspired by the network architecture of [9] and the self-supervised pipeline of [10] and builds the multi-echo denoising framework on them.

Parallels can be drawn between adverse weather denoising and deep image denoising, where camera images are restored, as these methods modify the original pixels and try to acquire a clean image [12,13]. This can be done in a supervised or self-supervised manner. Self-supervised methods typically utilize blind-spot to learn the noise [14,15]. Despite the success of deep image denoising, the methods do not directly suit the purpose of LiDAR adverse weather denoising as the data is sparse, and the task is to remove noise points and keep valid points unaltered. Some works have developed dense point cloud denoising methods [16, 17], which have an equivalent goal to deep image denoising. However, these methods are yet to be implemented on the adverse weather denoising task.

Preliminary work has been done on airborne particle classification using multi-echo point clouds. Stanislas et al. [18] studied supervised deep learning methods for classifying airborne particles using the multi-echo measurement as a feature. They predicted the class only for the first echo and used the alternative echo merely as a feature. This approach has the limitation of not utilizing alternative echoes to their fuller potential, as they can be used for viable substitute points to replace points of the strongest echo point cloud, which is the goal and contribution of our work. We are the first to present a method that picks substitutes from a multi-echo point cloud to replace invalid points of the strongest echo point cloud.

3. Methods

Multi-echo input formulation. A traditional single-echo point cloud is defined as $\mathbf{P}_s \in \mathbb{R}^{N_p \times N_c}$, where N_p and N_c denote the number of points and channels, respectively. A multi-echo point cloud is defined as $\mathbf{P}_m \in \mathbb{R}^{N_p \cdot N_e \times N_c}$, where N_e denotes the number of echoes per emitted laser pulse. In this work, point clouds are processed in an ordered format. We define multi-echo ordered format as a spherical projection of the point cloud $\Gamma : \mathbb{R}^{N_p \cdot N_e \times N_c} \rightarrow \mathbb{R}^{H \times W \times N_e \times N_c}$, where H and W stand for image dimensions of the projection. As echoes from the same laser pulse have the same azimuth and elevation angles, they are stacked on dimension N_e, and a vector along this dimension is also denoted as an echo group E_g. Finally, the multi-echo ordered point cloud is $\mathbf{P}_{meo} \in \mathbb{R}^{H \times W \times N_e \times N_c}$.
Multi-echo denoising task formulation. The intuition behind multi-echo denoising is to see through the noise caused by snowfall by recovering points that carry useful information from other echoes. Additionally, misleading or irrelevant points caused by other echoes must be discarded. To formulate this task, let $M(\mathbf{P}_{nm}) = \mathbf{P}_c$, where $\mathbf{P}_{nm} \in \mathbb{R}^{N_p \cdot N_e \times N_c}$ is the noisy multi-echo point cloud and $\mathbf{P}_c \in \mathbb{R}^{(N_p \cdot N_e - N_n - N_r) \times N_c}$ is the obtained clean point cloud. N_n denotes the number of removed noise points, and N_r denotes other irrelevant points, for instance, duplicate and artifact points caused by the refraction of the laser. The remaining points $\mathbf{P}_c$ include substitutes recovered from alternative echoes. Ultimately, the goal is to obtain $\mathbf{P}_c$, equivalent to a standard single-echo point cloud in clear weather. This work defines $M(\cdot)$ as a self-supervised neural network described in the following subsections.

3.1. SMEDen

Multi-echo neighbor encoder. The multi-echo ordered point cloud $\mathbf{P}_{meo}$ is the input to the multi-echo neighbor encoder, which is one of our contributions and the main difference to the work of Bae et al. [10]. It is illustrated in Fig. 3. This module processes $\mathbf{P}_{meo}$ into upper bound KNN sets using Euclidean distance to the reference point cloud, i.e., strongest echo point cloud. It encodes these values into a feature tensor $\mathbf{\Pi}$. The reference point cloud is the strongest point cloud, given that it is the default in clear weather. Thus, it is most likely to contain points of objects of interest. We formulate the multi-echo neighbor encoder as a set of convolutions. For simplicity, components of $\mathbf{P}_{meo}$ are denoted as: $\mathbf{P}_{xyz}$ – Cartesian coordinates, $\mathbf{P}_{\theta\phi}$ – azimuth and elevation coordinates, and $\mathbf{P}_r$ – range coordinates.

$$\begin{aligned}\mathbf{\Pi} &= \mathbf{w}_t * \mathbf{P}_{meo} \\ &= \mathbf{w}_t(\mathbf{P}_r[argmink_{\vec{p}_2}(\psi(\vec{p}_2, \vec{p}_3))] \\ &\oplus (\mathbf{P}_{\theta\phi}[\vec{p}_2] - \mathbf{P}_{\theta\phi}[argmink_{\vec{p}_2}(\psi(\vec{p}_2, \vec{p}_3))]))\end{aligned} \tag{1}$$

where $\mathbf{w}_t \in \mathbb{R}^{k \cdot N_e \times S_o}$ indicates trainable weights, $\vec{p}_2 = (h, w, 0)$, $h \in [\![0, H]\!]$, and $w \in [\![0, W]\!]$ indicates a pixel coordinate on the strongest echo point cloud. $argmink_{\vec{p}_2}(\cdot)$ returns indices of $k \times N_e$ strongest echo values which minimize Euclidian distance to multi-echo queries $\vec{p}_3 = (h, w, \hat{e}), \hat{e} \in [\![0, N_e]\!]$. $\oplus$ is the concatenation operation, and

$$\begin{aligned}\psi(\vec{p}_2, \vec{p}_3) &= \|\mathbf{P}_{xyz}[\vec{p}_2 + \Delta\vec{p}_2] - \mathbf{P}_{xyz}[\vec{p}_3]\|_2 \\ \Delta\vec{p}_2 &\in \mathbf{A}_c \qquad \forall\psi(\vec{p}_2, \vec{p}_3) < C_r\end{aligned} \tag{2}$$

where C_r is a fixed radius cutoff hyper-parameter defining the upper bound for the neighbor search, which ensures that only local points are considered. $\mathbf{A}_c$ defines the elements considered in the search. $\mathbf{P}_{\theta\phi}$ are encoded to preserve the 3D information because the grid positions of the neighbors $\mathbf{P}_{meo}$ are lost due to the nature of the KNN search. With this, the architecture can utilize the original 3D information for the predictions. The final output of the module is the activated features. Some of the experiments are conducted with a traditional single-echo dataset SnowyKITTI [9]. Therefore, SMEDen is made to be convertible to a single-echo mode SSEDen. When the model is in the single-echo mode, the difference is in the multi-echo neighbor encoder, where the

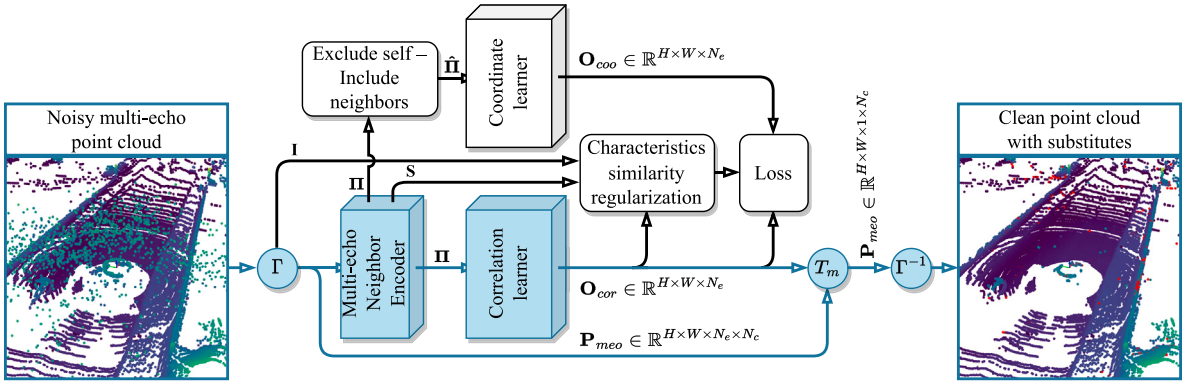

Fig. 2. The proposed self-supervised multi-echo denoising architecture (SMEDen). The multi-echo point cloud is first projected, and then the projected point cloud is processed by the Multi-echo Neighbor Encoder. Next, trainable points are masked in the Exclude self – Include neighbors – module. Then, the Coordinate learner processed this masked point cloud. Meanwhile, the Correlation learner processes the point cloud without masks. The loss is computed using the output of the networks and characteristics similarity regularization. Finally, a threshold T_m filters out the noisy points. White modules are used for training only.

neighbors are only searched for the available echo. In practice $N_e = 1$ and $\mathbf{w}_t$ convolves over single-echo point cloud $\mathbf{P}_s$. Next, the coordinate and correlation learner models process these features.

Coordinate learner. The coordinate learner predicts the coordinate of a point p_i given the neighboring points of p_i. To simplify the task and enable faster convergence, the three dimensions $p_i \in \mathbb{R}^3$ are reduced to one $r_i = \|p_i\|_2 \in \mathbb{R}$. The coordinate learner learns point coordinates by minimizing the absolute distance error to the actual coordinates.

Correlation learner. The correlation learner predicts the predictability of the coordinate of point p_i. The loss is formulated so that the correlation learner predicts a high value for p_i if its coordinate is challenging to predict to compensate for a high coordinate prediction error. During training, this network learns the correlation of points to their neighbors. We assume that highly correlating points correspond to objects of interest. During inference time, the points can be filtered based on the predicted value of the correlation learner.

Exclude self–Include neighbors. Blind spots are added to the input of the coordinate learner during training. Previously, blind spot methods have been used in deep image denoising [14,15]. We combine the multi-echo neighbor encoder with the blind-spot approach. For a p_i, the input is its KNN set without the query (Fig. 3). Only its neighbors are processed to learn the coordinate of p_i, and the coordinate of p_i is predicted. However, to learn the correlation of p_i, the whole KNN set, including the query, is processed. This ensures that the correlation learner can utilize the valuable information of the query point. The benefit of our method is that physically neighboring data points are more relevant when estimating the value of the hidden point compared to prior work where more unrelated 2D grid relations are used.

3.2. Characteristics similarity regularization

We propose the characteristics similarity regularization (CSR). CSR accelerates the convergence and increases the accuracy. Notably, it requires only *one* hyper-parameter, the size of the search k_{CSR}. The CSR is built on the following assumptions about the nature of the noise caused by snowfall. (1) An expected echo caused by an airborne snowflake has a typical intensity [19]. (2) Snowfall causes more sparse point clouds compared to other objects [6]. On the basis of these assumptions, a process that guides the predictions into a distribution similar to intensity and sparsity should increase the convergence. With this insight, we build a regularization process to guide the predictions to the above distribution[1].

[1] It is important to note here that we do not assume any specific distribution of the characteristics of the noise points but instead encourage the model to learn the connections between these characteristics and other features relative to the output.

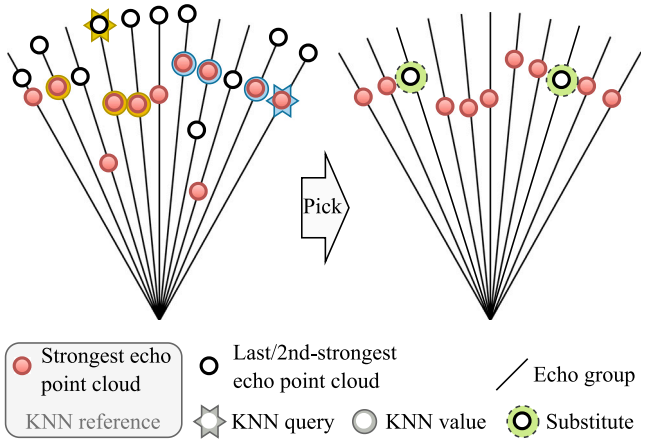

Fig. 3. In the proposed multi-echo neighbor encoder, KNN sets are computed using multi-echo query and KNN reference. We use the strongest echo point cloud as it is *de facto* in single-echo approaches. Two example queries illustrate how KNN sets look. "Exclude self – Include neighbors" simply discards KNN queries and keeps the values. From this, the coordinate learner predicts the queries. The point cloud on the right illustrates the final output of our method, where substitutes are points from alternative echoes.

The intensity is scaled with a squared range to accommodate possible range-based bias in the intensity values

$$\mathbf{I} = \mathbf{I}_{raw} \odot \mathbf{P}_r^2 \tag{3}$$

where $\mathbf{P}_r = \|\mathbf{P}_{xyz}\|_2$. The sparsity is defined as follows,

$$\mathbf{S} = E_d(\mathbf{P}_{meo}) \odot \frac{1}{\mathbf{P}_r} \tag{4}$$

where $E_d(\cdot)$ returns Euclidean distance to the nearest neighbor for each point. Thus, sparsity can be obtained as a *free lunch* from $\mathbf{\Pi}$. It is normalized with the range matrix $\mathbf{P}_r$, as sparsity $s \propto r$. Then, we get the characteristics map as a concatenation of the components $\mathbf{\Theta} = \mathbf{I} \oplus \mathbf{S}$. The similarity is computed using normal distributions of neighbors of $\mathbf{\Theta}$. That is, for each element in $\mathbf{\Theta}$, arguments of k_{CSR}-nearest-neighbors are computed using the Euclidean distance in $\mathbf{\Theta}$. A regression goal for a corresponding output is the Z-score of the distribution collected with the neighbor search. This regression goal forms the characteristics similarity loss term using the absolute error. The described regularization process forces those correlation learner outputs $\mathbf{O}_{cor}$ with similar characteristics to have similar values, which we expect will encourage the model to converge. Finally, we summarize CSR as the Algorithm 1.

Algorithm 1: Characteristics similarity regularization.

for a batch ∈ batches **do**
- $\mathbf{I} \leftarrow$ Equation (3) ▷ Intensity
- $\mathbf{S} \leftarrow$ Equation (4) ▷ Sparsity
- $\mathbf{\Theta} \leftarrow \text{CONCAT}(\mathbf{I}, \mathbf{S})$ ▷ Characteristics map
- $\mathbf{\Phi} \leftarrow \mathbf{O}_{cor}[\text{ARGNEIGBORS}(\mathbf{\Theta}, k_{CSR})]$
- $\mathbf{\Xi} \leftarrow |\text{ZSCORE}(\mathbf{O}_{cor}, \mathbf{\Phi})|$ ▷ Characteristics similarity loss

3.3. Architecture and loss function

Architecture. Our proposed architecture is presented in Fig. 2. The architecture consists of two encoder–decoder neural networks. The networks are identical except for the input since the coordinate learner processes point clouds with blind spots. The encoder–decoder closely resembles 4DenoiseNet [9] using three residual blocks without the temporal branch.

Loss function. The two networks are trained jointly. As stated above, the Exclude self – Include neighbors is performed only for a randomly selected subset of points, denoted by $\mathbf{P}_s$. Therefore, we train only those points. The loss function in its final form is defined as follows:

$$\mathcal{L} = \frac{1}{|\mathbf{P}_s|} \sum_{p_s \in \mathbf{P}_s} \left(\frac{\lambda \cdot |\mathbf{O}^{coo}_{p_s} - \mathbf{P}^r_{p_s}|}{\lceil \mathbf{P}^r \rceil_{p_s} \odot \text{EXP}(\mathbf{O}^{cor}_{p_s})} + \mathbf{O}^{cor}_{p_s} + \mathbf{\Xi}_{p_s} \right) \tag{5}$$

where $\mathbf{O}^{coo}$ and $\mathbf{O}^{cor}$ are the outputs of the coordinate and correlation learner, respectively. The loss function minimizes the absolute coordinate prediction error in the numerator. The denominator forces the output $\mathbf{O}^{cor}$ to a high value when the coordinate is challenging to predict, i.e., noisy snowfall points. Exponential of $\mathbf{O}^{cor}$ was discovered experimentally to improve performance. Thus, during inference, $\mathbf{O}^{cor}$ is used as a score to define the noisy points. A fixed hyperparameter λ scales this term relative to the other terms. A regulating term $+\mathbf{O}^{cor}$ prevents predictions from exploding. The division by $\lceil \mathbf{P}^r \rceil$ reduces the range-induced bias of the learning process, as the correlation with neighbors is inversely proportional to the range. $\lceil \mathbf{P}^r \rceil$ is rounded to total meters to stabilize the learning process. $\mathbf{\Xi}_{p_s}$ is the CSR loss, which is summed up with the rest of the loss function. The model converges when the coordinate learner has learned meaningful high-level features of valid points and the correlation learner learns to output a high score, i.e., low correlation value, for invalid points.

3.4. Inference mode

During the inference mode, only the correlation learner is used. The output $\mathbf{O}_{cor}$ is processed into class labels. For this purpose, we formulate the multi-echo denoising classes in the following manner:

- valid strongest echo, $(S \wedge T) \rightarrow VS$,
- potential substitute, $(\neg S \wedge T \wedge B \wedge D) \rightarrow PS \in E_g$,
- discarded, $(\neg VS \vee \neg PS) \rightarrow DI$,

where S, T, B, D are boolean formulas for "the strongest echo", "satisfies threshold T_n", "the best score", and "a different coordinate to strongest", respectively. E_g denotes the echo group, i.e., the values along the N_e-dimension of $\mathbf{O}_{cor}$. Here, we formulate B in general form, but in our case, it is equal to the predicted correlation value $\mathbf{O}_{cor}$. During inference time, the DI labeled points are removed (T_m in Fig. 2), and the remaining point cloud is the final output of our method.

4. Experimental results

4.1. Implementation details

Datasets. The experiments are conducted with the STF dataset [20], which includes multi-echo LiDAR point clouds from snowfall. STF provides the strongest and last echo point clouds. If the last echo is the same as the strongest one, it is replaced by the second strongest one. We have labeled a 500 multi-echo point cloud snowfall subset of STF [20] dataset since point-wise labels for multi-echo data were not previously available. Moreover, a semi-synthetic single-echo SnowyKITTI dataset [9] is used for comparison against single-echo methods. Multi-echo results were achieved with models trained with STF [20] dataset, and single-echo results were achieved with models trained with SnowyKITTI [9] dataset.

Training and testing details. The hyper-parameter settings were selected with a grid search as follows. The learning rate is 0.01, the linear learning rate decay is 0.99, the fixed hyper-parameter in Eq. (5) $\lambda = 5$, and the $k_{CSR} = 9$. We use the stochastic gradient descent optimizer with a momentum of 0.9 and train for 30 epochs. The train/validate/test-set split ratio is 55/10/35, where all sets have a wide range of different snowfall conditions and are from different sequences. A threshold $T_n = 0$ is empirically found to yield the best results during inference. The models are run on an RTX 3090 GPU using Python 3.8.10 and PyTorch 1.12.1 [21]. The runtime result of LIOR [7] is achieved with a Python implementation. Since implementations for LIOR and SLiDE are not easily accessible, we implemented them and shared the code in our repository[2]. We use the official C++ implementation of DROR [6][3]. For the learned approaches, the runtime of the neural network is reported. Our model is trained and tested with the coordinates of the points, including intensity. We noticed that removing the intensity did not have a significant effect on the results.

4.2. Multi-echo denoising results

We analyze the performance qualitatively on the STF [20] dataset. The results are presented in Fig. 4, where each row indicates an individual sample. The corrupted strongest echo is on the left-hand side, where a detail window is denoted with fuchsia. The strongest echo is visualized to highlight the idea of recovering substitute points from alternative echoes. With that, we want to emphasize that the input to the methods is the multi-echo point cloud. The baseline method is next to the corrupted strongest echo point cloud, and our SMEDen is on the right-hand side.

We compare our method to the baseline Multi-Echo DROR (MEDROR). It is a simple modification to the classical DROR [6]. The neighbors are computed similarly as in SMEDen. The searched neighbors are then thresholded as in DROR [6], but all echoes are computed instead of processing only the strongest echo. Then, based on the inlier–outlier classification, we classify the points according to Section 3.4.

The substitute points are visualized in red. These are recovered points from alternative echoes (in our case, conditional last and second strongest). As seen in Fig. 4, SMEDen finds viable substitute points. The results prove the concept of multi-echo denoising in snowfall. They also demonstrate that a simple classical method such as MEDROR is insufficient as there are only a few recovered substitute points and many false negatives.

Classical MEDROR performs much better with low noise than with a high noise level. Note here that the threshold of MEDROR can be adjusted, but we noticed that it results in a great number of false positives. On the contrary, our SMEDen performs well in all noise

[2] https://github.com/alvariseppanen/SMEDen
[3] https://github.com/nickcharron/lidar_snow_removal

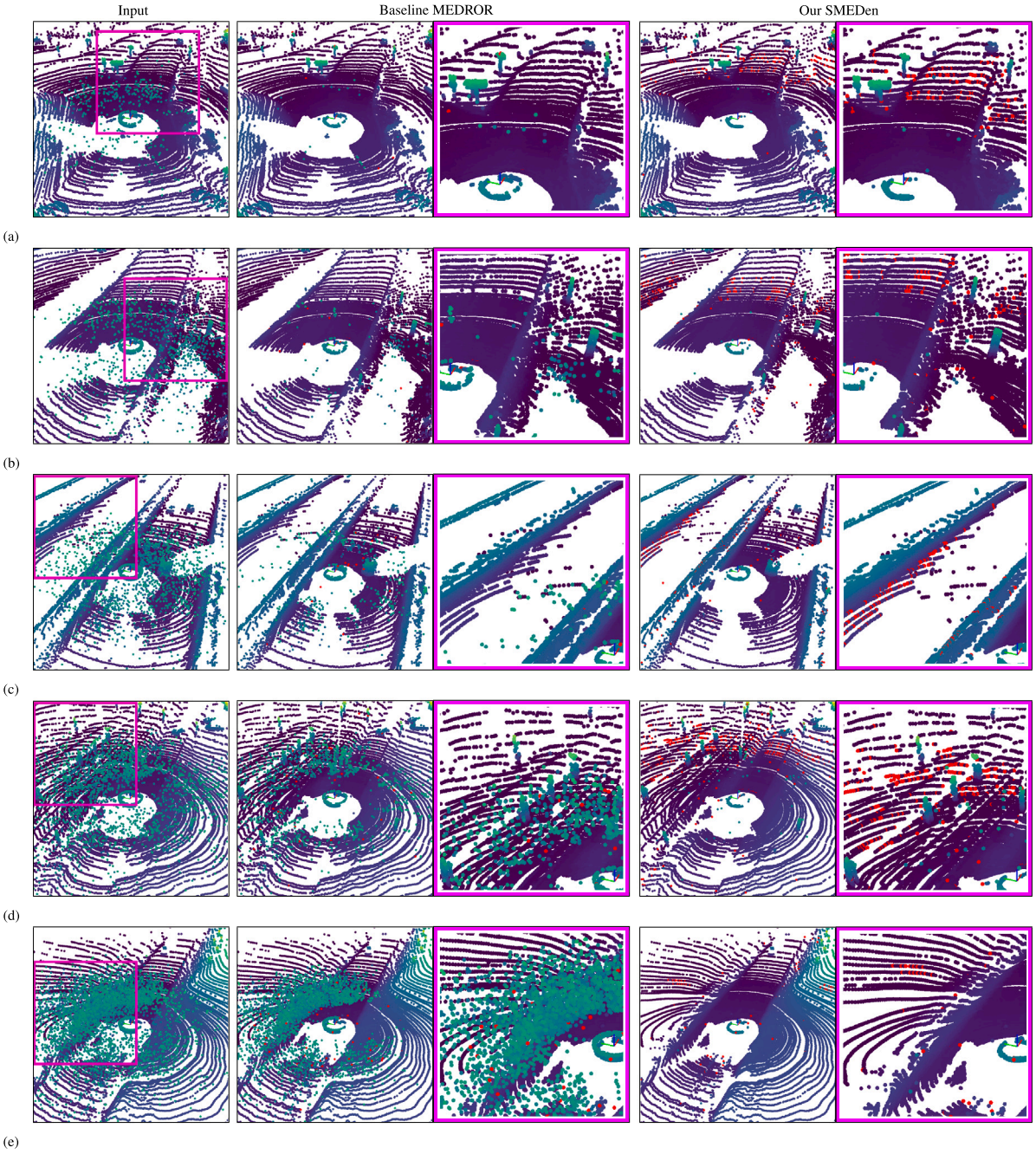

Fig. 4. Multi-echo denoising performance on real data. Each row is an individual sample. The corrupted strongest echo is on the leftmost column, the output of the baseline method MEDROR is in the middle, and the output of our SMEDen is on the right column. **Red indicates potential substitute points**. MEDROR mostly fails, whereas our SMEDen picks successfully viable substitute points.

levels, even in extreme conditions in Fig. 4(e). This shows that our method achieves superior performance.

The quantitative multi-echo results are presented in Table 1. SMEDen performs significantly better compared to MEDROR in both noise removal and finding substitutes. The quantitative results confirm that a learned approach is superior to the classical approach. As a pioneer in multi-echo denoising, these results serve as a baseline for future studies in multi-echo denoising in snowfall.

Table 1
Multi-echo results on the labeled snowfall subset of the STF [20] dataset.

Method		MEDROR	SMEDen (Ours)
Type		Classical	Self-supervised
Runtime ms		125	2.3
Param. $\cdot 10^6$		10^{-5}	1.13
IoU	Noise	0.269	0.792
	Substitute	0.953	0.981

4.3. Single-echo denoising results

Single-echo performance on the SnowyKITTI-dataset [9] can be seen in Table 2. Our algorithm is compared to the state-of-the-art self-supervised SLiDE [10] and with well-proven classical methods DROR [6] and LIOR [7], and supervised methods WeatherNet [22] and 4DenoiseNet [9]. The methods are evaluated with the widely

Table 2
Single-echo results on the SnowyKITTI [9] dataset. SSEDen is SMEDen in single-echo mode.

Method	Type	IoU			Runtime	Param.
		Light	Medium	Heavy	ms	$\cdot 10^6$
DROR [6]	Classical	0.453	0.451	0.440	97	10^{-5}
LIOR [7]	Classical	0.448	0.447	0.434	120	10^{-5}
WeatherNet [8]	Supervised	0.884	0.889	0.865	1.9	1.5
4DenoiseNet [9]	Supervised	0.975	0.976	0.977	1.3	0.6
SLiDE [10]	Self-supervised	0.778	0.745	0.743	2.0	1.73
SSEDen (Ours)	Self-supervised	0.933	0.854	0.843	2.2	1.13
Self-supervised improvement		0.155	0.109	0.100		

Table 3
Ablations of different modules. MH indicates multi-hypothesis prediction [10], CSR is the characteristics similarity regularization, NE is the single-echo variant of the Multi-echo neighbor encoder, and C_r is the cutoff radius. NN denotes a neural network, where 4DN is 4DenoiseNet variant [9], and WN is WeatherNet variant [8]. SSEDen is SMEDen in single-echo mode.

Method	MH	CSR	C_r	NE	NN	IoU		
						Light	Medium	Heavy
SLiDE [10]	✓	✗	✗	✗	WN	0.778	0.745	0.743
SSEDen variants	✗	✓	✗	✗	WN	0.872	0.778	0.778
	✗	✗	✓	✗	WN	0.776	0.757	0.816
	✗	✗	✗	✓	WN	0.879	0.823	0.799
	✗	✗	✗	✗	4DN	0.798	0.758	0.780
	✗	✗	✓	✓	4DN	0.912	0.825	0.802
	✗	✓	✗	✓	4DN	0.871	0.819	0.808
	✗	✓	✓	✗	4DN	0.854	0.795	0.806
	✗	✓	✓	✓	WN	0.885	0.819	0.792
	✓	✓	✓	✓	4DN	0.922	0.855	0.851
SSEDen (Ours)	✗	✓	✓	✓	4DN	0.933	0.854	0.843

adopted Intersection over Union (IoU) metric. The results are presented separately for the light, medium, and heavy noise levels for a more detailed indication of performance. Based on the results, the performance varies depending on the noise level. Our method achieves superior performance to these methods in all levels of noise.

We also report the runtime and parameter count. As the results indicate, our method has approximately the same runtime and uses 35% fewer parameters compared to SLiDE [10]. This is significant as fewer parameters help with over-fitting, reduce memory footprint, and enable faster convergence. A slight difference in the runtime between our method and SLiDE [10] results from the input layer, as the neighbor encoder is slower than a standard 2D convolution. Despite having similar architecture, our method has a slower runtime than 4DenoiseNet [9]. This is caused by a larger KNN search radius that it has. Overall, our method achieves superior performance and thus sets the new state-of-the-art in self-supervised snowfall denoising.

Ablation Study. An ablation study was conducted to assess the contributions of the characteristics similarity regularization, cutoff radius C_r, the neighbor encoder, and the neural network. Table 3 presents the IoU measurements when ablating the aforementioned modules. The ablation of CSR indicates that performance on the test set increases with the inclusion of the module. Excluding C_r has a significant effect on the performance as well. We suspect this is because limiting the input neighbors with a certain threshold excludes points irrelevant to the coordinate prediction. The neighbor encoder is replaced with a standard 2D convolution. The performance decreases because 2D convolution captures fails to capture spatial information. The neural network was changed to the WeatherNet [8] variant that SLiDE [10] uses. The network also affects self-supervised performance based on our experiment. Thanks to the proposed modules, the multi-hypothesis prediction [10] does not affect the performance. Therefore, it is not used in the final model.

5. Conclusion

We proposed multi-echo denoising in snowfall. The idea is to recover points that carry useful information from alternative echoes and remove points that are caused by airborne particles, refractions, and reflections. This can be thought of as seeing through the snowfall-induced noise. To achieve this goal, we also proposed the Self-supervised Multi-Echo Denoising (SMEDen) approach, which includes novel characteristics similarity regularization. Both quantitative and qualitative results on a real-world multi-echo snowfall dataset show the superiority of our method compared to the classical baseline MEDROR. Moreover, our method achieved new state-of-the-art self-supervised performance on a single-echo semi-synthetic dataset. The main limitation of this study is that the multi-echo data included only the strongest and conditional last echoes. Therefore, studies with a point cloud dataset that has more echoes would be valuable. However, such a dataset is not publically available during the time of this study. Based on the results, our work enables safer and more reliable LiDAR point cloud data in snowfall. Therefore, it should increase the safety of autonomous driving and driving assistance systems, for instance.

CRediT authorship contribution statement

Alvari Seppänen: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Risto Ojala:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis. **Kari Tammi:** Supervision, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] M. Kutila, P. Pyykönen, M. Jokela, T. Gruber, M. Bijelic, W. Ritter, Benchmarking automotive LiDAR performance in arctic conditions, in: 2020 IEEE 23rd International Conference on Intelligent Transportation Systems, ITSC, IEEE, 2020, pp. 1–8.

[2] M. Hahner, C. Sakaridis, M. Bijelic, F. Heide, F. Yu, D. Dai, 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.

[3] European Commission, Road safety in the European union, 2018, http://dx.doi.org/10.2832/169706, Accessed 2 September 2022.

[4] US Department of Transportation: Federal Highway Administration, How do weather events impact roads? 2019, https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm. Accessed 2 September 2023.

[5] T.-H. Sang, C.-M. Tsai, Histogram-based defogging techniques for lidar, in: 2022 IEEE 16th International Conference on Solid-State & Integrated Circuit Technology, ICSICT, IEEE, 2022, pp. 1–3.

[6] N. Charron, S. Phillips, 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.

[7] J.-I. Park, J. Park, K.-S. Kim, Fast and accurate desnowing algorithm for LiDAR point clouds, IEEE Access 8 (2020) 160202–160212.

[8] R. Heinzler, F. Piewak, P. Schindler, W. Stork, Cnn-based lidar point cloud de-noising in adverse weather, IEEE Robot. Autom. Lett. 5 (2) (2020) 2514–2521.

[9] A. Seppanen, R. Ojala, K. Tammi, 4DenoiseNet: Adverse weather denoising from adjacent point clouds, IEEE Robot. Autom. Lett. (2022).

[10] G. Bae, B. Kim, S. Ahn, J. Min, I. Shim, SLiDE: Self-supervised LiDAR de-snowing through reconstruction difficulty, in: European Conference on Computer Vision, Springer, 2022, pp. 283–300.

[11] M.-Y. Yu, R. Vasudevan, 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.

[12] C. Liu, X. Hu, Deep neural network with deformable convolution and side window convolution for image denoising, Pattern Recognit. Lett. 171 (2023) 92–98, http://dx.doi.org/10.1016/j.patrec.2023.05.015, URL https://www.sciencedirect.com/science/article/pii/S016786552300140X.

[13] A. Hadri, A. Laghrib, H. Oummi, An optimal variable exponent model for Magnetic Resonance Images denoising, Pattern Recognit. Lett. 151 (2021) 302–309, http://dx.doi.org/10.1016/j.patrec.2021.08.031, URL https://www.sciencedirect.com/science/article/pii/S0167865521003275.

[14] T. Huang, S. Li, X. Jia, H. Lu, J. Liu, Neighbor2neighbor: Self-supervised denoising from single noisy images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14781–14790.

[15] Z. Wang, J. Liu, G. Li, H. Han, Blind2unblind: Self-supervised image denoising with visible blind spots, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2027–2036.

[16] P. Hermosilla, T. Ritschel, T. Ropinski, Total denoising: Unsupervised learning of 3D point cloud cleaning, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 52–60.

[17] S. Luo, W. Hu, Differentiable manifold reconstruction for point cloud denoising, in: Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 1330–1338.

[18] L. Stanislas, J. Nubert, D. Dugas, J. Nitsch, N. Sünderhauf, R. Siegwart, C. Cadena, T. Peynot, Airborne particle classification in lidar point clouds using deep learning, in: Field and Service Robotics: Results of the 12th International Conference, Springer, 2021, pp. 395–410.

[19] H.W. O'Brien, Visibility and light attenuation in falling snow, J. Appl. Meteorol. Climatol. 9 (4) (1970) 671–683.

[20] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, F. Heide, Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather, in: The IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2020.

[21] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Adv. Neural Inf. Process. Syst. 32 (2019).

[22] R. Heinzler, P. Schindler, J. Seekircher, W. Ritter, W. Stork, Weather influence and classification with automotive lidar sensors, in: 2019 IEEE Intelligent Vehicles Symposium, IV, IEEE, 2019, pp. 1527–1534.