

Article

Multiscale Feature Fusion for the Multistage Denoising of Airborne Single Photon LiDAR

Shuming Si¹, Han Hu^{1,*}, Yulin Ding¹, Xuekun Yuan¹, Ying Jiang¹, Yigao Jin¹, Xuming Ge¹, Yeting Zhang², Jie Chen³ and Xiaocui Guo⁴

¹ Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University, Chengdu 611756, China

² State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430072, China

³ China Academy of Railway Sciences Group Co., Ltd., Beijing 100081, China

⁴ Beijing Jingwei Information Technology Co., Ltd., Beijing 100081, China

* Correspondence: han.hu@swjtu.edu.cn

Abstract: Compared with the existing modes of LiDAR, single-photon LiDAR (SPL) can acquire terrain data more efficiently. However, influenced by the photon-sensitive detectors, the collected point cloud data contain a large number of noisy points. Most of the existing denoising techniques are based on the sparsity assumption of point cloud noise, which does not hold for SPL point clouds, so the existing denoising methods cannot effectively remove the noisy points from SPL point clouds. To solve the above problems, we proposed a novel multistage denoising strategy with fused multiscale features. The multiscale features were fused to enrich contextual information of the point cloud at different scales. In addition, we utilized multistage denoising to solve the problem that a single-round denoising could not effectively remove enough noise points in some areas. Interestingly, the multiscale features also prevent an increase in false-alarm ratio during multistage denoising. The experimental results indicate that the proposed denoising approach achieved 97.58%, 99.59%, 95.70%, and 77.92% F1-scores in the urban, suburban, mountain, and water areas, respectively, and it outperformed the existing denoising methods such as Statistical Outlier Removal. The proposed approach significantly improved the denoising precision of airborne point clouds from single-photon LiDAR, especially in water areas and dense urban areas.

Keywords: single-photon LiDAR; point cloud denoising; multiscale features; random forest



Citation: Si, S.; Hu, H.; Ding, Y.; Yuan, X.; Jiang, Y.; Jin, Y.; Ge, X.; Zhang, Y.; Chen, J.; Guo, X. Multiscale Feature Fusion for the Multistage Denoising of Airborne Single Photon LiDAR. *Remote Sens.* **2023**, *15*, 269. <https://doi.org/10.3390/rs15010269>

Academic Editor: Xuan Zhu

Received: 12 October 2022

Revised: 18 December 2022

Accepted: 28 December 2022

Published: 2 January 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, airborne LiDAR is widely used for large-scale topographic mapping [1,2], urban building extraction and modeling [3], and forest inventories [4,5]. Airborne LiDAR can be divided into three categories based on the sensor: linear mode LiDAR (LML), Geiger mode LiDAR, and single-photon LiDAR (SPL) [6]. With the development of single-photon detection technology, SPL is equipped with photon-sensitive detectors, and therefore, it needs only to detect a few returned photons to achieve ranging. This greatly enhances the efficiency of data acquisition [7]. Researchers have found that SPL has great potential in national mapping work [8,9]. Thus, SPL has been broadly employed in the 3D Elevation Program (3DEP) [10] and European operational mapping projects [11]. However, SPL captures a lot of noise, mainly caused by the dark counts from sensitive detectors, solar backscatter from the atmosphere within the inspected surface and pixel field of view (FOV), and laser backscatter from the atmosphere [12].

The noisy points from the LML are mostly because of the low outliers generated by multiple path errors and are well removed using statistical or priori knowledge-based techniques.

The basic idea of the statistical methods is to calculate the point numbers within each neighborhood or to calculate local point density to obtain a statistical histogram, and finally, distinguish between noisy and non-noisy points based on a threshold value. The basic idea of the priori knowledge-based methods is to apply morphological closing to remove the low outliers and to apply morphological opening to connect the isolated ground points. However, there will still be a large number of noise points in the SPL point clouds after denoising using traditional methods, and further denoising is inevitable. (Figure 1). Therefore, an effective SPL point cloud denoising method is urgently needed. There are also many other significant problems that have to be solved. Two of these problems are as follows:

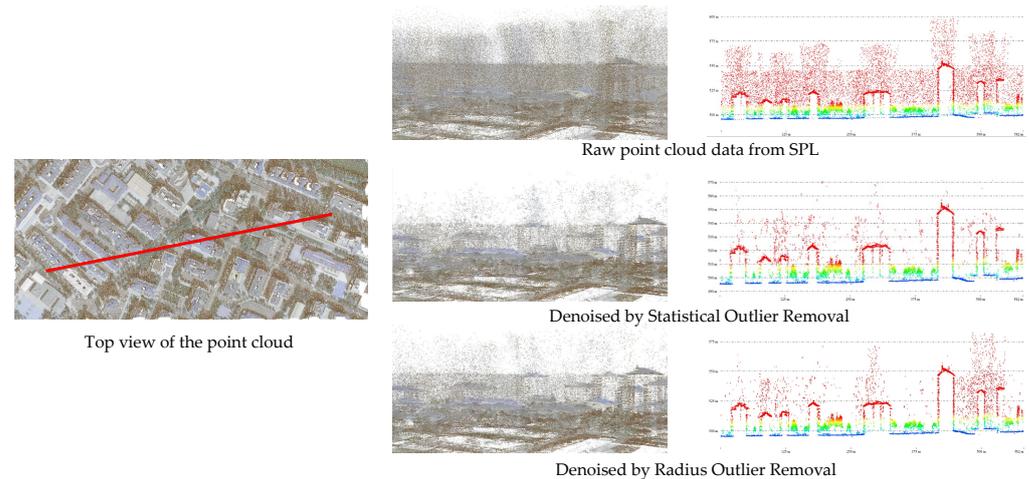


Figure 1. The raw point clouds data from SPL and the denoising results using existing algorithms. The three images in the right hand show the profile line in the area of the red line on the left.

1. **The sparsity assumption of point cloud noise does not hold:** The noise in the LML point clouds is generally sparse, and many existing denoising algorithms have been proposed based on the sparsity assumption. However, the high photon sensitivity of SPL results in numerous noisy points. The noisy point density in the SPL point clouds far exceeds that in the LML point clouds. Therefore, it is hard to remove the noisy points in SPL using existing denoising algorithms (see the second and third rows of Figure 1).
2. **The noisy points cannot be identified with a clear mechanism of generation:** The noise in the LML point clouds is primarily due to low outliers generated by the multiple path errors. The low outliers can be removed by morphological opening and closing. In contrast, noisy points in the SPL point clouds are generated for various uncertain reasons, thus, it is difficult to construct a denoising model using a priori knowledge-based method [13].

To address these problems, we proposed a machine learning approach for SPL point cloud denoising using multiscale features and multistage denoising techniques. We utilized the *multiscale features* of the point cloud to extract the local information of the point cloud under different receptive fields to obtain more comprehensive local features of the point cloud and to train a more effective denoising model. Then, we proposed to use *multistage denoising* to solve the problem of unsatisfactory single-round denoising effect in some regions. Specifically, the steps of this paper's approach are as follows. First, we computed the point cloud multiscale features based on the multiscale neighborhoods. Second, we introduced the random forests module to learn the multiscale features extracted from the training point cloud data. Then, we used the trained classifier to denoise the testing point cloud data. Finally, we carried out multiround denoising to improve the denoising accuracy.

In summary, our contributions are as follows: We proposed (1) a multiscale feature fusion module that combined structural information from a larger context in the SPL point clouds and (2) a multistage denoising approach that carried out multiround denoising without significantly increasing the false-alarm ratio. The rest of this paper is organized as follows. The summary of the related work on the SPL, point cloud denoising methods, and the point cloud features are introduced in Section 2. Section 3 introduces the details of the proposed approach for SPL point cloud denoising. The experiments and the denoising results are illustrated in Section 4. Section 5 summarizes the main work of this paper.

2. Related Work

2.1. Single-Photon LiDAR

Linear mode LiDAR uses the Avalanche Photodiode (APD) detectors to capture reflected signals. Hundreds of photons are required for LML to reliably detect an object, while the single-photon LiDAR system utilizes single-photon avalanche diode (SPAD) detectors [14,15]. As for SPAD detectors, the photon sensitivity enables the SPL to detect a single object with only a few photons and to perform terrain mapping at higher altitudes [16]. Leica's SPL100 is the first commercially available single-photon LiDAR system, and it emits a very short laser pulse with a pulse duration of 400 ps, which is divided into 10×10 sub-beams by diffractive optical elements [17]. For each sub-beam, reflected signals can be received in the form of photons by an individual SPAD detector. The SPAD detectors will excite photoelectron pulses after receiving photons, and then the photon counting module will obtain an effective Time of Flight (ToF) that can be employed to calculate the position of a single object [18–22].

To minimize the influence of atmospheric and solar background noise, a range gate technique is used in the single-photon detector. The returned photons are accepted when the time corresponds to the range gate, otherwise, they are treated as noise [10,23]. Although the range gate technique can suppress some of the noise, a large number of noisy points are still captured, thus, further denoising is still needed.

Like LML, SPL is a hybrid multisensor 3D information acquisition system. Take the SPL100 as an example, in addition to the laser sensor, it contains a GNSS module for determining the position of the sensor platform, an IMU module for determining the attitude [18], and an 80-megapixel camera to provide color information for the point cloud [17].

2.2. Point Cloud Denoising

In this section, we briefly review the previous approaches to denoise point clouds.

(1) Digital image processing based denoising approaches: Many early scholars projected 3D point cloud data to 2D digital images and then denoised them according to the denoising algorithms of digital images [24]. Using these approaches to denoise the SPL point clouds, a lot of information would be lost in the projection process from point cloud data to digital images.

(2) Local statistical denoising approaches: Zhao et al. [25], Duan et al. [26], Zhang et al. [27], Balta et al. [28] make use of the local information of the point clouds. First, the point number or the point density is counted in each neighborhood to obtain the statistical histogram of the point cloud, and then the appropriate threshold is set to complete the judgment of noisy points. These approaches have been widely employed and have been integrated into *PCL* (Point Cloud Library) and *open3d*. However, these approaches are all proposed based on the sparsity assumption of point cloud noise. The large number of the noise points in the original SPL point cloud does not satisfy this assumption, and the SPL point cloud cannot be denoised well using these approaches.

(3) Clustering-based denoising approaches [29]: Earlier, many scholars often used the DBSCAN clustering algorithm to classify noisy points [30]. Zhu et al. [31] found that the DBSCAN algorithm did not work well in complex terrain and proposed the modified ordering points to identify the clustering structure (OPTICS) approach. However, clustering

denoising for SPL point clouds can only eliminate some nonclustered points, and the spatial correlation between points cannot be well expressed if only using a single scale.

(4) Smoothing-based denoising approaches: The basic idea of smoothing-based denoising approaches is to adjust each point to the fitted surface to achieve a smoothing effect. The main smoothing-based denoising techniques include the Mean-Shift algorithm [32], moving least squares (MLS) [33], and bilateral filtering [34]. Han et al. [35] claimed that MLS could be used to handle denoising problems by adjusting the points onto the fitted surface iteratively and Digne and de Franchis [34] employed the bilateral filtering algorithm to adjust the points along their normal. However, the amount of noise in the SPL point cloud is much larger than that in the linear mode LiDAR point cloud, and only a small number of outliers can be removed using these approaches.

(5) Hybrid denoising approaches: Hybrid denoising techniques usually fuse multiple denoising algorithms based on the priori information of the noise to enhance denoising precision. Zaman et al. [32] proposed an algorithm for fusing kernel density estimation with particle swarm optimization, mean-shift, and bilateral filtering. To develop the process of fusion denoising, we first need to figure out the noise generation mechanism of SPL point cloud. However, the noise generation mechanism of SPL is completely different from that of LML, and it is difficult for us to build a noise model, so it is not easy to choose a suitable fusion denoising approach.

2.3. Point Cloud Features

2.3.1. Neighborhood Definition

SPL often collects amounts of point information without explicit connection among points, but the connection among points is crucial for local feature description. For being able to build the connection among points, it is inevitable to define a local neighborhood containing the considered LiDAR points [36]. Generally, different strategies can be employed to establish the neighborhood of each LiDAR point. Among them, the following methods of neighborhood definition are commonly used:

(1) Spherical neighborhood: a fixed radius of sphere $r \in \mathbb{R}$ should be defined, and all points within r constitute the neighborhood of the LiDAR point; (2) k-neighborhood: a fixed number $k \in \mathbb{N}$ has to be defined, and the nearest k points to the LiDAR point constitute the neighborhood of the spatial LiDAR point; and (3) cylindrical neighborhood: a fixed radius $r \in \mathbb{R}$ should be defined. When projected onto the ground, all the points within r constitute the neighborhood of the LiDAR point [37,38].

2.3.2. Feature Selection

Feature extraction of the LiDAR point cloud has been extensively studied. Geometric feature extraction approaches, proposed by Weinmann et al. [37], Blomley and Weinmann [39], introduced local point density, verticality, and height difference to extract neighborhood features. Many studies implied that eigenvalue-based LiDAR features were potentially valuable for the description of local geometric features and Dittrich et al. [40] validated eigenvalue-based LiDAR features' robustness. Weinmann et al. [37], Singh and Sreevalsan-Nair [41], Che et al. [42], Tomková et al. [43], Gallwey et al. [44], Ni et al. [45], and Vetrivel et al. [46], Thomas et al. [47] calculated the eigenvalues and eigenvectors of the covariance matrix in the neighborhood of each LiDAR point by Principal Component Analysis (PCA) and performed the local geometric feature extraction of the point cloud by eigenvalues and eigenvectors. Wang et al. [48], Yastikli and Cetin [49] utilized height-based features, echo-based features, intensity features, texture features, and waveform features to acquire the LiDAR point cloud classification. Lucas et al. [50] used echo-based features, geometric features, and eigenvalue-based features to classify the vegetation. The results suggest that classifiers trained with these features were able to successfully separate vegetation from the background.

2.3.3. Multiscale Construction

Brodu and Lague [51] first proposed a strategy to construct multiscale features by computing geometric features at N scales and concatenating these features so that each point has multiscale local neighborhood features. Experiments showed that multiscale features improved the spatial resolution of the classification compared with a single scale.

The optimal neighborhood scale is not the same for different features, and Dong et al. [52] recommended that we should choose the optimal neighborhood for different local neighborhood features.

Neighborhood selection is a crucial part of feature extraction, therefore, Demantke et al. [53] first estimated the optimal neighborhood of the point cloud and then calculated the local geometric features based on the optimal neighborhood. After weighing the computational cost against the optimal size of the neighborhood, Huang et al. [54] proposed to extract geometric features using multiscale neighborhoods instead of directly estimating the optimal size of the neighborhood. They extracted features at three neighborhood scales for classifier training based on the point density, and the final classification precision was better than that of the single-scale feature classification. Singh et al. [55] successfully achieved the separation of bolts from roofs using a binary classifier trained with multiple scale features.

3. Methodology

3.1. Architecture Overview

To resolve the problem that existing methods are not effective in removing SPL noise, we proposed a multistage denoising approach using a classifier trained based on multiscale features. The structure of the proposed network is shown in Figure 2. It largely contains three parts: (1) the definition of multiscale point cloud neighborhoods, (2) the extraction of multiscale point cloud features, and (3) the noisy point labeling by machine learning algorithms. In the first part, we constructed multiscale neighborhoods of the point cloud using the KD tree. In the second part, we computed the neighborhood-wise features of each LiDAR point on multiple-scale neighborhoods, while the point-wise features were calculated by the intensity and echo values of each point. Furthermore, all these features were concatenated to consist of the multiscale features of the LiDAR point. In the third part, we applied a machine learning algorithm to train a noisy point classifier for SPL point clouds using attribute information consisting of echo-based, intensity-based, and eigenvalue-based features. Finally, we tested the classifier on the test point cloud data. To enhance the denoising precision of the SPL point clouds, we also carried out multistage denoising.

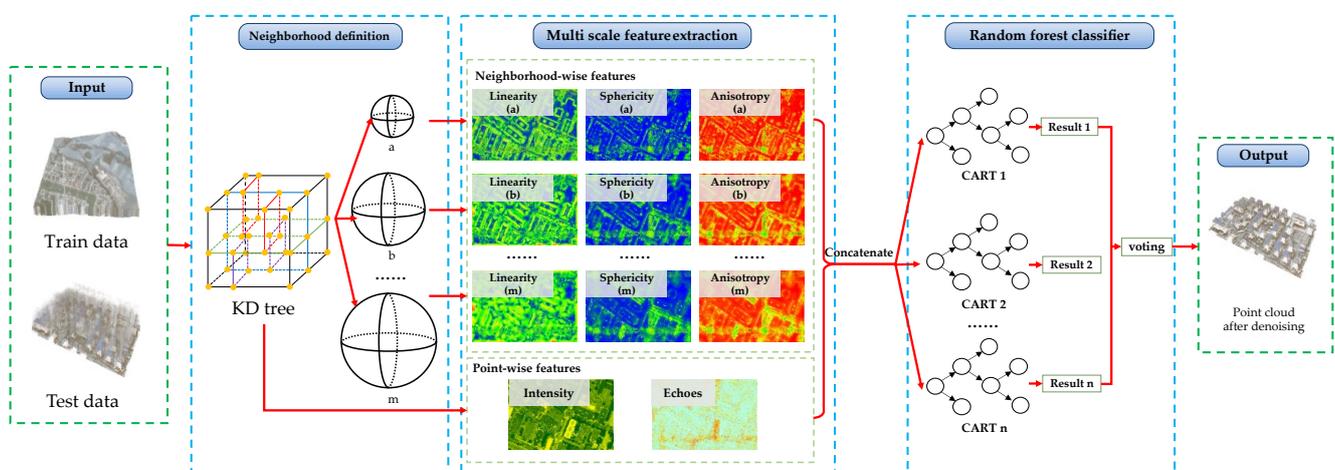


Figure 2. Structure of the proposed network. There are three main parts: neighborhood definition, a multiscale feature extraction, and a random forests module. In the neighborhood definition module, a, b, \dots, m represent different scales.

3.2. Multiscale Hybrid Features of Single-Photon LiDAR

After reviewing the relevant literature, we selected and calculated the point-wise features and the neighborhood-wise features in our framework, which could be categorized with respect to the following feature types:

3.2.1. Point-Wise Features

An SPL point cloud can be represented as a set of points $P = \{P_i | i = 1, \dots, n\}$, where each LiDAR point P_i not only has its three coordinates (x, y, z) but also has an intensity value (I), the number of returns (R_t), a return number (R), and RGB values, which can be used in feature extraction. We acquired point-wise features using intensity-based features and echo-based features obtained from the raw SPL point cloud data.

- **Intensity-based features:** The intensity values reflect the intensity of the reflected signals by the object being measured. The intensity values can distinguish some noisy points from building points, since the intensity of some background noisy points is much lower than that of building points (Figure 3d).
- **Echo-based features:** N and N_e can describe the echo-based features, where N represents the total number of echoes contained in the current pulse and N_e stands for the normalized number of echoes. Echo-based features can initially extract the vegetation points, since there may be multiple echoes from the same pulse in the vegetation areas (Figure 3e).

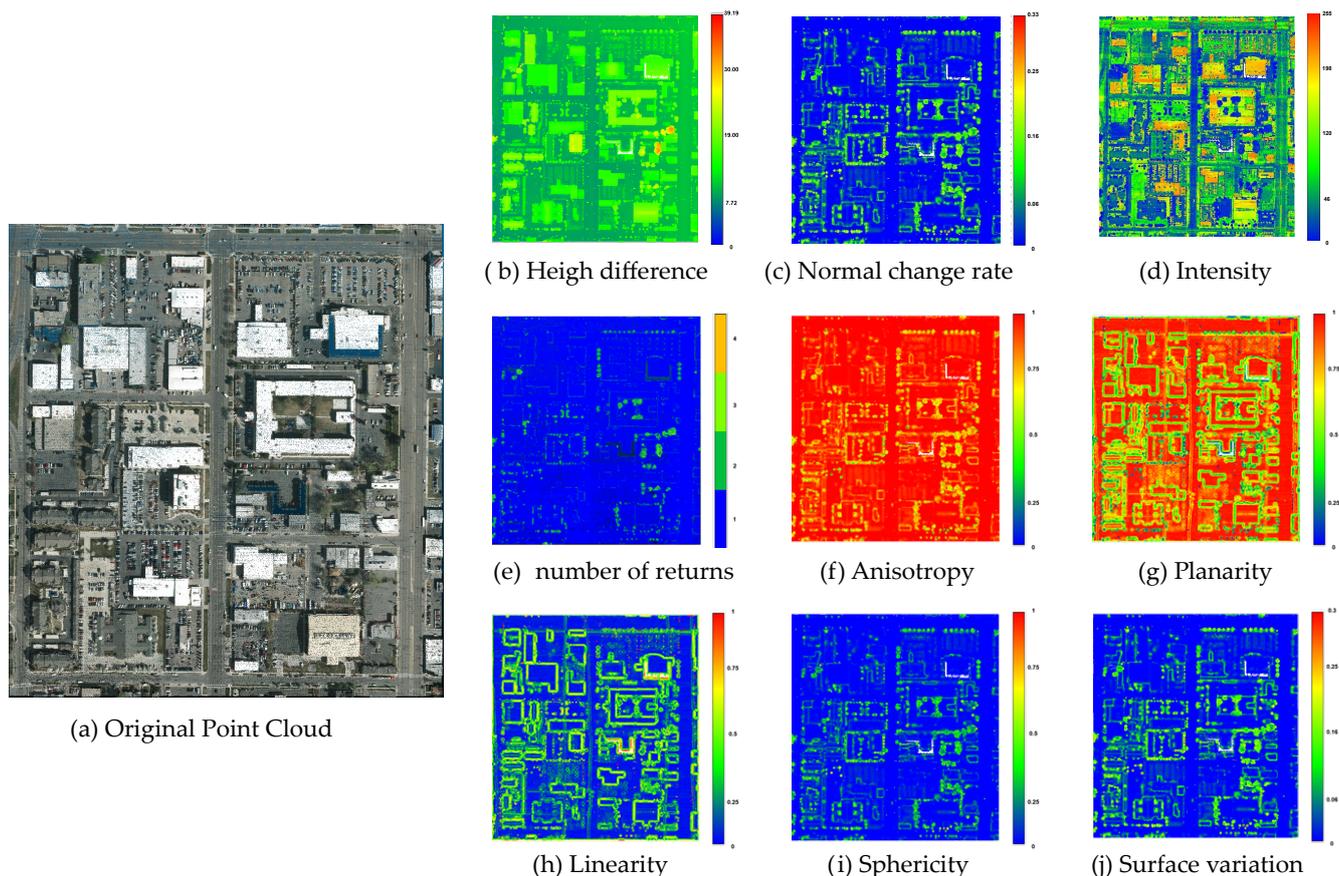


Figure 3. Original point cloud and some representative LiDAR point cloud features.

3.2.2. Neighborhood-Wise Features

The neighborhood-wise features are computed by relying on the local neighborhoods. The local neighborhood around the point p_c is a point set N_i including all the points within a sphere with a fixed radius r centered at p_c . The neighborhood-wise features can be separated into height-based features and eigenvalue-based features.

- **Height-based features** predominantly include height difference (Δ_z), height standard deviation (σ_z^2), and normal change rate (C). The values of height-based features will be larger in vegetated areas, high noise areas, and building boundaries, where there are more undulations (Figure 3c). We can therefore use height-based features to highlight smooth areas such as ground and building roofs.
- **Eigenvalue-based features** are widely used for local feature extraction from point clouds, which can effectively portray the distribution of the point cloud in the neighborhood. The eigenvalue-based features are based on the covariance matrix $D \in \mathbb{R}^{3 \times 3}$ computed within the neighborhood N_p following Equation (1):

$$D = \begin{bmatrix} p_1 - \bar{p} \\ \vdots \\ p_n - \bar{p} \end{bmatrix}^T \begin{bmatrix} p_1 - \bar{p} \\ \vdots \\ p_n - \bar{p} \end{bmatrix} \quad (1)$$

Here, $p_i = (x_i, y_i, z_i)$ is a point contained in the neighborhood N_p . The geometric center \bar{p} can be defined by Equation (2):

$$\bar{p} = \frac{1}{n} \sum_{i=1}^n p_i \quad (2)$$

Since the covariance matrix is a symmetric positive-definite matrix, its three eigenvalues λ_1, λ_2 , and λ_3 ($\lambda_3 \leq \lambda_2 \leq \lambda_1$) exist. Therefore, the eigenvalues can be used to characterize the local neighborhood shapes by calculating the eigenvalue-based features represented by Anisotropy (A_λ), Planarity (P_λ), Sphericity (S_λ), and Linearity (L_λ), according to Equations (3)–(6):

$$A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1} \quad (3)$$

$$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (4)$$

$$S_\lambda = \frac{\lambda_3}{\lambda_1} \quad (5)$$

$$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}, \quad (6)$$

as shown in Figure 3f–i.

3.2.3. Multiscale Neighborhood Features

The selection of the neighborhood size is a critical factor for extracting neighborhood-wise features. However, for specific point cloud data, it is not trivial to find the optimal size of the local neighborhood. For the same LiDAR point in the same point cloud, the local features that appear linear in the neighborhood are shown in Figure 4a, those that appear planar in the neighborhood are shown in Figure 4b, and those that appear scattered in the neighborhood are shown in Figure 4c. Thus, for the same LiDAR point in the same point cloud, the extracted local features are not the same in different sizes of local neighborhoods. When combining the information from multiple-scale neighborhoods, we can extract finer and more comprehensive local features. Based on the above analysis, we propose applying the multiscale neighborhood features in the feature extraction process. The main idea of the multiscale neighborhood is to combine neighborhood-wise features extracted from different scale neighborhoods.

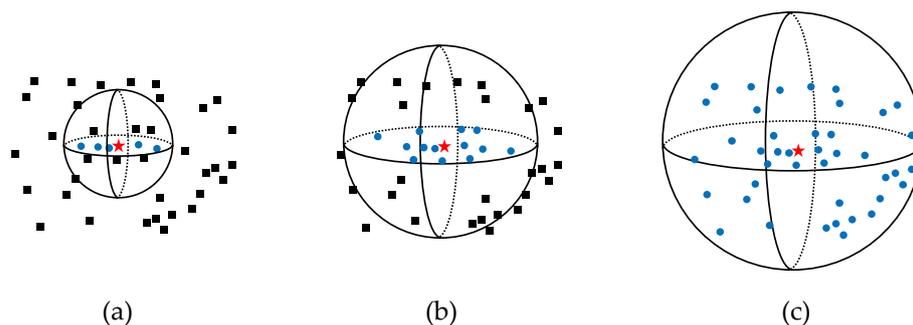


Figure 4. Neighborhoods at different scales. In this representation, the red star is the centroid point of the neighborhood, the blue points are the neighbors within the local neighborhood, and the black points are the outside points of the local neighborhood. In the figure, (a–c) represent different scales.

Therefore, to fully capture local structures, the neighborhood-wise features were calculated using three different neighborhood sizes (r); here, $r = r_1, r_2$, and r_3 , and in our experiments, $r_1 = 5$ m, $r_2 = 10$ m, and $r_3 = 15$ m. Then, the neighborhood-wise features were concatenated to consist of the multiscale features for training the noisy point classifier. The multiscale feature vector (f) for each point is given by Equation (7):

$$f = [A_{\lambda(r_1)}, A_{\lambda(r_2)}, A_{\lambda(r_3)}, P_{\lambda(r_1)}, P_{\lambda(r_2)}, P_{\lambda(r_3)}, S_{\lambda(r_1)}, S_{\lambda(r_2)}, S_{\lambda(r_3)}, L_{\lambda(r_1)}, L_{\lambda(r_2)}, L_{\lambda(r_3)}, N, N_e, I]^T \quad (7)$$

The extracted multiscale features were used for training and testing the noisy point classifier, which is covered in detail in Section 3.3.

3.3. Noise Removal Using Random Forests

Random forests are one of the most popular machine learning algorithms and are based on the idea of ensemble learning, which combines several weak learners to create a single strong learner. Random forests proposed by Breiman [56] consist of a set of unpruned decision trees. They can perform well on large datasets without overfitting, and the feature selection process can be automatically accomplished by the Gini coefficients during the training phase. Random forests were employed to select the most significant features and to distinguish between noisy and non-noisy points. Therefore, we did not need to select the features manually.

In the training phase, we first computed the multiscale features of all points in the point cloud of the training dataset and saved the results in the set U . Using a sampling with replacement method, n points were drawn from the set U to form the set U_t , which was used to train the t th decision tree of the random forests. N decision trees were constructed in the same way, and experiments indicated that the random forests classifier was able to create a good balance between classification precision and algorithm efficiency when $N = 500$.

Decision tree generation is a crucial part of random forests classifier training. During the training, we selected the classification and regression tree (CART). The classification and regression tree can complete the construction of the decision tree by continuously splitting the nodes through the Gini coefficients, which can be calculated using Equations (8) and (9). Compared with other decision tree generation algorithms, CART can build trees more

efficiently, since CART is a binary tree and the Gini coefficients are used in node splitting, which can reduce a large number of logarithmic operations.

$$Gini(U_{ti}) = 1 - \sum_{k=0}^{|y|} p_k^2 \quad (8)$$

$$Gini_coefficient(U_{ti}, a) = \sum_{v=1}^V \frac{|U_{ti}^{V_j}|}{|U_{ti}|} Gini(U_{ti}^{V_j}) \quad (9)$$

Here, U_{ti} is the feature set in the i th node in the t th decision tree, $p_k (k = 0, 1)$ is the proportion of points of the class k in the set U_{ti} , and $U_{ti}^{V_j}$ is all points in the set U_{ti} that take a value of a^{V_j} on feature a .

In the testing phase, we first calculated the multiscale features for a single LiDAR point in the SPL point cloud of the testing dataset, and then put the features into the trained random forests classifier. Each CART in the random forests could give a unit prediction for the point, and finally, the random forests classifier voted for the most popular label.

4. Experiment

We developed a prototype framework for the proposed approach of using multiscale features to denoise airborne SPL point cloud data by using the C++ programming language. We constructed the point cloud neighborhood using the KD tree and completed the feature learning using random forests, in which the open source *nanoflann* [57] was used for the implementation of the KD tree, and the *ranger* (random forests generator) [58] library was utilized for the implementation of random forests.

The experiments were conducted on a computer running Windows 10(×64) with one 8-core AMD Ryzen 7 5800X CPU, 32 GB Random Access Memory (RAM), and a 4 TB hard disk drive.

4.1. Datasets

We evaluated the performance of the proposed approach on the Navarra dataset provided by the Cartographic Department of the Navarra government. In 2017, this dataset was obtained using SPL100 from an altitude of about 4200 m above ground level (AGL) with a 30-degree field of view, which produced a swath width of about 2260 m. Flying at a speed of 90 m/s with an effective scan rate of 6 MHz led to an average point density of 14.5 points/m².

Moreover, all LiDAR points can obtain the X , Y , and Z coordinates, intensity values, and time stamps from the SPL100 system [59]. The detailed parameters of the dataset are shown in Table 1. The point cloud can be colorized by the orthoimages obtained from the RCD30 80 MPix RGBI camera [17]. According to the description, point clouds are automatically classified by traditional classification methods, and then the labels were further optimized by manual adjustment. The point cloud dataset with semantic information can be downloaded for free via Amazon Web Services (AWS) [60]. In addition, we also checked the labels manually once we had acquired the data.

In this experiment, we first selected a portion of the point cloud data from the Navarra dataset as the training data, which contained a total of 1,809,827 points. Then, we used four areas that were different from the training dataset as test data: urban, suburban, mountain, and water areas (Figure 5). For the urban test data, we used data from some parts of the largest city in the Navarra province, Pamplona, which contained a total of 342,622,218 points. The terrain was nearly flat, but there was a great variety of categories. The architectures included buildings, sports fields, and churches. In addition to architectures, this area also covered a large number of rivers and vegetation. For the mountain test data, we utilized data from some parts of Sorlada, which contained a total of 357,761,913 points and covered mainly trees, grassland, rocks, and a few other categories. However, the topography was very relieved with steep slopes. For the water test data,

we employed data from Laguna de Pitillas and its vicinity, which contained a total of 264,319,000 points and was dominated by lakes and low vegetation. For the suburban test data, we chiefly used the data from the area around Bevinzana, which contained a total of 240,141,925 points. The categories in this area were mostly trees and grasses, with few tall categories and some topographic relief.

Table 1. Dataset information.

Dataset Name	Navarra Dataset
LiDAR system	SPL100
Point density	14.5 points/m ²
Flight height	4200 m (AGL)
Field of view (FoV)	30°
Flight speed	90 m/s
Swath width	2260 m
Effective scan rate	6 MHz
Data coverage	The Navarra province of Spain
Coordinate system	ETRS89 / UTM zone 30 N (EPSG25830)

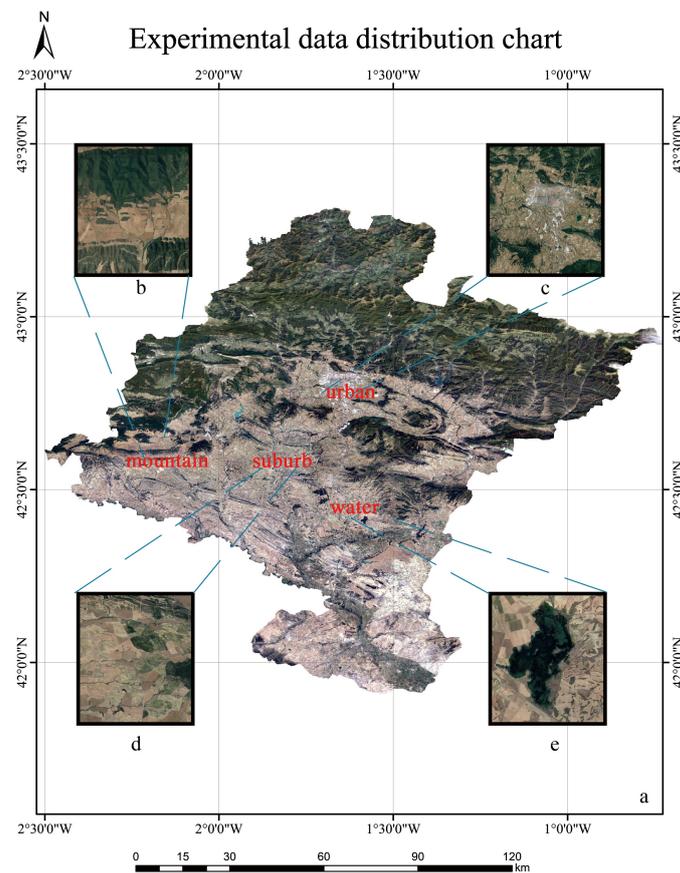


Figure 5. Experimental data distribution chart; (a) overview map of Navarra, background: Google Earth; (b) the mountain area of Sorlada; (c) part of the urban area of Pamplona; (d) part of the suburban area of Berbinzana; (e) Laguna de Pitillas Lake and its vicinity.

4.2. Evaluation Metrics

The purpose of denoising the SPL point clouds using machine learning is to determine whether each point is a noisy or non-noisy point. To compare with existing denoising methods, we applied the confusion matrix and employed four commonly used evalua-

tion metrics: recall (R), precision (P), overall accuracy (OA), and F1-score (F_1). They are computed based on the confusion matrix as follows:

$$R = \frac{TP}{TP + FN} \quad (10)$$

$$P = \frac{TP}{TP + FP} \quad (11)$$

$$OA = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \quad (13)$$

Here, TP refers to the number of predicted points, where the noisy point classifier correctly predicts the noisy point as the noisy point, FN represents the number of predicted points where the noisy point classifier incorrectly predicts the noisy point as the non-noisy point, FP stands for the number of predicted points where the noisy point classifier incorrectly predicts the non-noisy points as the noisy point, and TN signifies the number of predicted points where the noisy point classifier correctly predicts the non-noisy point as the non-noisy point.

4.3. Comparison of Experimental Results

We compared the performance of the proposed method with some classical point cloud denoising approaches, such as Statistical Outlier Removal and Radius Outlier Removal, as well as the denoising approach in the *lastools* toolkit. Statistical Outlier Removal and Radius Outlier Removal achieved a great improvement by using statistical information within local neighborhoods, which is broadly employed in data manipulation of the point cloud. We implemented these approaches using *PCL*. Our approach used multiscale features and identified noisy points using a random forests classifier. Our approach outperformed the existing point cloud denoising and attained a better result in the four different areas.

4.3.1. Experiments in the Urban Area

The urban area was characterized by a wide variety of categories, including buildings, rivers, and vegetation, meaning that point clouds had complex spatial geometry features. We conducted comparative experiments on SPL point cloud data in the urban area to compare the denoising effect of our approach with existing approaches.

Table 2 reveals that our approach outperformed the existing denoising approaches in both accuracy and F1-score. We also used the original RandLA-Net [61] to train a noise point classifier and conducted comparison experiments. To compare the denoising results of our approach with other existing approaches, we visualized the example of denoising results for urban areas in Figure 6. The figure contains data from nine sample point cloud files, covering a total of 9 km² of the urban area. Four figures in the top right-hand corner exhibit local details by profile line figures. The middle four figures display the difference in details of the profile lines after applying different methods of denoising, and the bottom four figures depict the overall denoising results by Digital Surface Model (DSM) for convenient comparison. Our method enabled a more comprehensive and more accurate labeling of noisy points in the point cloud. As shown in the area marked by red dashed lines in the middle row, existing denoising algorithms still reserved a large number of noisy points, and many ground points were filtered out. This means that, as for SPL point clouds, the use of machine learning to fully learn the local features of the point cloud can more accurately identify noisy points, especially in areas with dense buildings and complex category types.

Table 2. Performance evaluation of the compared approaches and ours in the urban area.

Methods	Recall (%)	Precision (%)	Accuracy (%)	F1-Score (%)
Radius Outlier Removal	82.63	98.51	93.71	89.87
Statistical Outlier Removal	65.34	99.55	88.19	78.89
Lastools' Denoising Workflow	88.84	99.25	95.83	93.76
RandLA-Net	93.98	94.76	94.07	94.37
Ours	96.82	98.35	98.38	97.58

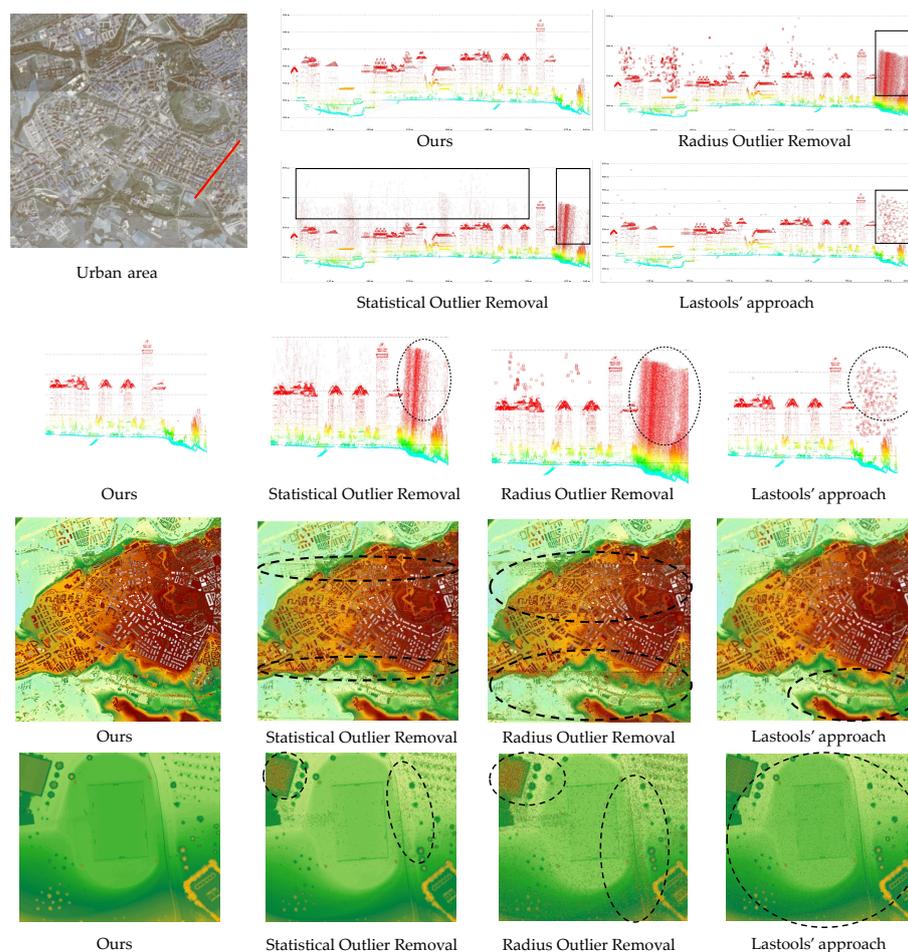


Figure 6. Examples of the denoising results in the urban area. The four images in the upper right-hand corner show the profile line in the area of the red line on the left. The third row represents the profile line partial details for convenient comparison. The fourth row indicates the DSM generated by the different denoising methods. The bottom row indicates some details of the DSM to facilitate comparison.

4.3.2. Experiments in the Suburban Area

In the suburban area, there were few tall categories, but there was some topographic relief. A comparative experiment was carried out on the point cloud data of the suburban area to compare the denoising effect of our approach with the existing approaches for SPL point clouds data in the suburban area.

Table 3 shows that all four approaches obtained splendid denoising results due to the presence of a relatively homogeneous category. However, our approach still achieved a 2.69% improvement in F1-score compared with Radius Outlier Removal and a 3.20% improvement compared with Statistical Outlier Removal. To the best of our knowledge, both approaches are broadly employed for point cloud denoising, and Statistical Outlier Removal has been integrated into the open source software named *CloudCompare*.

Table 3. Performance evaluation of the compared approaches and ours in the suburban area.

Methods	Recall (%)	Precision (%)	Accuracy (%)	F1-Score (%)
Radius Outlier Removal	94.01	99.97	98.70	96.90
Statistical Outlier Removal	93.05	99.99	98.49	96.39
Lastools' Denoising Workflow	88.52	99.98	97.22	93.91
RandLA-Net	98.81	99.20	98.81	99.01
Ours	99.86	99.31	99.82	99.59

We also visualized the denoising results for point cloud data from the suburban area in Figure 7 to compare our denoising results with those of other approaches. As can be seen from the results, our approach was able to remove the vast majority of noisy points and to retain the original ground information well, as we introduced a multiscale feature extraction module, and the classifier was also able to fully learn the local structures. Our proposed approach distinguished the noisy points from non-noisy points more accurately than the existing approaches, especially for areas where the point cloud density was not uniformly distributed. The denoising results of our method and existing methods are shown in the profile line figures and DSM figures (Figure 7).

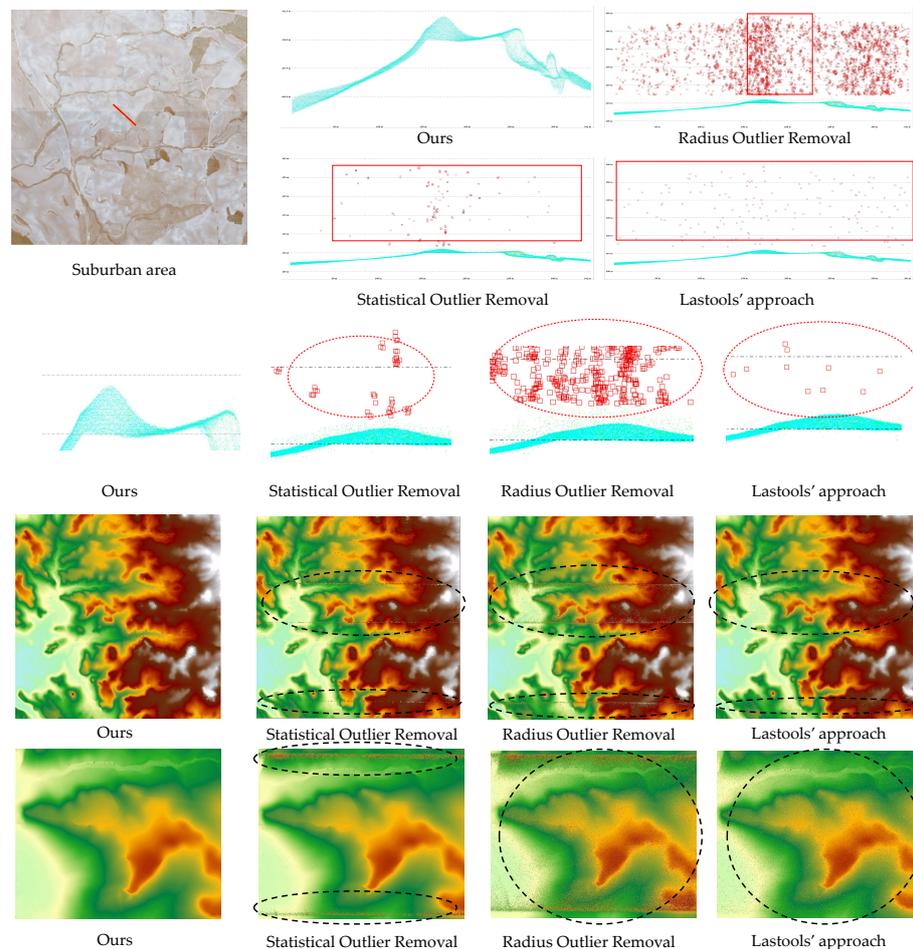


Figure 7. Examples of the denoising results in the suburban area. The four images in the upper right-hand corner show the profile line in the area of the red line on the left. The third row represents the profile line partial details for convenient comparison. The fourth row indicates the DSM generated by the different denoising methods. The bottom row indicates some details of the DSM to facilitate comparison.

4.3.3. Experiments in the Mountain Area

To validate the generalization of the proposed denoising approach in different areas, we conducted experiments on SPL point cloud data in the mountain area to compare the performance of our approach with that of existing denoising approaches. *Lastools* integrates a large number of tools for processing a large number of LiDAR point clouds, where *lasnoise* can accurately determine whether each point is noisy or not based on the search size and outliers of each point.

The results demonstrate that our approach outperformed the existing approaches (see Table 4). The mountain area had a single feature type; therefore, both the proposed approach and the existing approaches exhibited good denoising results. However, it can be observed that our approach had better accuracy and F1-score than existing denoising approaches.

Table 4. Performance evaluation of the compared approaches and ours in the mountain area.

Methods	Recall (%)	Precision (%)	Accuracy (%)	F1-Score (%)
Radius Outlier Removal	88.86	99.04	97.46	93.68
Statistical Outlier Removal	87.90	99.44	97.33	93.32
Lastools' Denoising Workflow	87.36	99.74	96.99	93.14
RandLA-Net	96.48	94.26	96.48	95.35
Ours	99.47	92.22	98.11	95.70

The example results in the mountain area are illustrated in Figure 8 for intuitive comparison. As can be seen from the results, because the point density of many noisy points in the SPL point cloud data was close to the point density of the features, a large number of noisy points were retained as feature points after denoising using the existing approaches. The proposed approach could remove the noisy points that were not removed by the existing approaches while retaining the non-noising points, as shown in the profile line images.

4.3.4. Experiments in the Water Area

The SPL point cloud data on the water surface are very sparse, caused by the poor reflection of the laser by the water surface. Moreover, there is a large amount of water vapor over the lake. Therefore, when the laser pulse detects the water vapor, it is backscattered and picked up by the detector, making the number of noisy points on the lake high and causing the denoising task to become difficult. Based on the above issues, comparative experiments were conducted in the water area to compare the denoising results of our approach with those of existing approaches.

Table 5 displays that our approach outperformed the classical denoising approaches. Especially, our approach achieved an 11.48% accuracy improvement compared with the Radius Outlier Removal and 12.01% compared with the Statistical Outlier Removal.

Examples of results in the water area are visualized in Figure 9 to compare our denoising results with those of existing methods. As can be observed from the experimental results, the existing denoising approaches could not identify the noisy points on the water surface well due to their large number and density. Although the proposed denoising approach was also affected by the sparse water surface point cloud data, the overall denoising effect was still better than the existing denoising approaches, since we introduced multiple neighborhood features to jointly train the noisy point classifier. Compared with existing denoising approaches, the proposed approach could extract noisy points more accurately and comprehensively, even on water surfaces with sparse point cloud data, as shown in the profile line figures and the DSM figures (Figure 9).

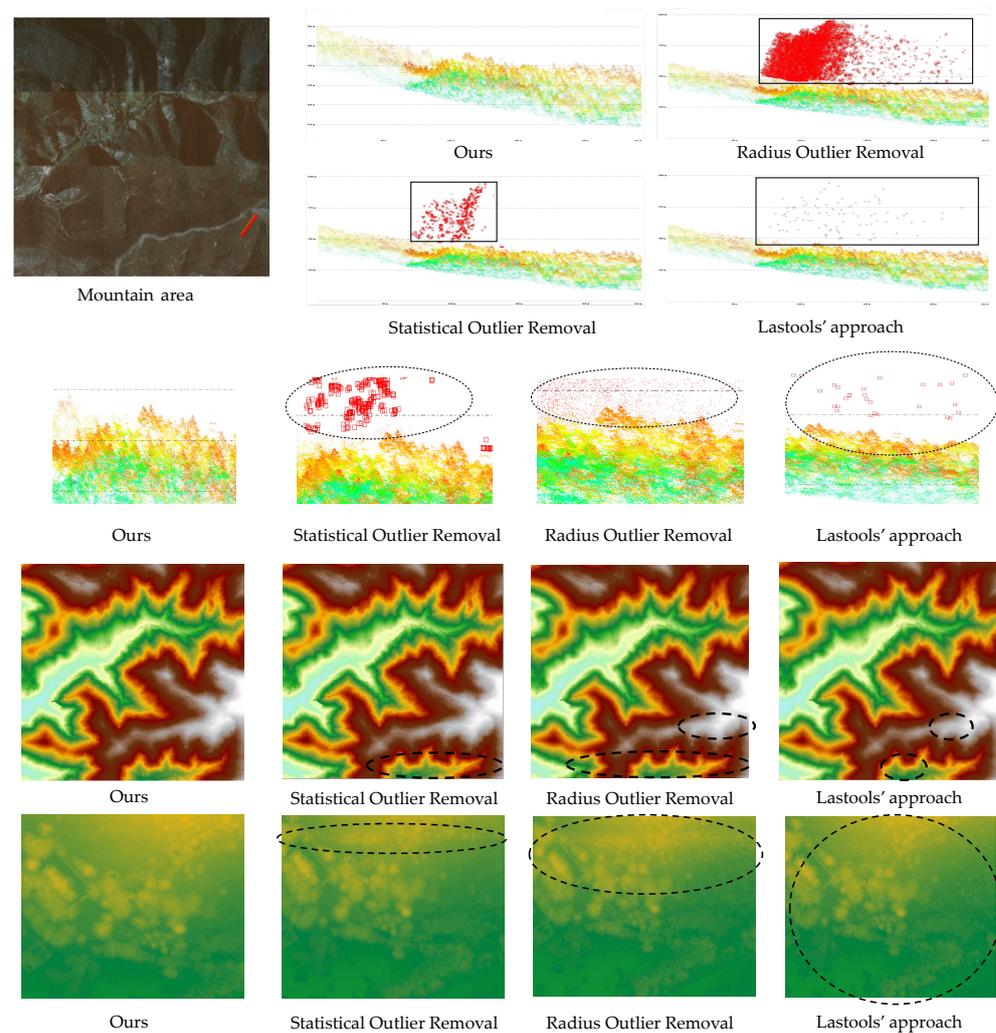


Figure 8. Examples of the denoising results in the mountain area. The four images in the upper right-hand corner show the profile line in the area of the red line on the left. The third row represents the profile line partial details for convenient comparison. The fourth row indicates the DSM generated by the different denoising methods. The bottom row indicates some details of the DSM to facilitate comparison.

Table 5. Performance evaluation of the compared approaches and ours in the water area.

Methods	Recall (%)	Precision (%)	Accuracy (%)	F1-Score (%)
Radius Outlier Removal	37.58	87.11	71.57	52.51
Statistical Outlier Removal	31.19	98.63	71.04	47.40
Lastools' Denoising Workflow	45.74	98.41	74.19	62.46
Ours	71.49	85.61	83.05	77.92

4.4. Result Analysis

4.4.1. Analysis of Multiscale Features

To verify whether multiscale features could improve the denoising precision of SPL point cloud data compared with single-scale features, we also trained noisy point classifiers for denoising using single-scale and multiscale features in the urban, suburban, mountain, and water area data, respectively. Unlike single-scale features, multiscale features can fuse neighborhood features from multiple scales. The results are presented in Table 6, indicating the improved performance of accuracy and F1-score under multiscale features compared

with single-scale features. The results suggest that our proposed approach achieved greater accuracy and F1-score using multiscale features than single-scale features.

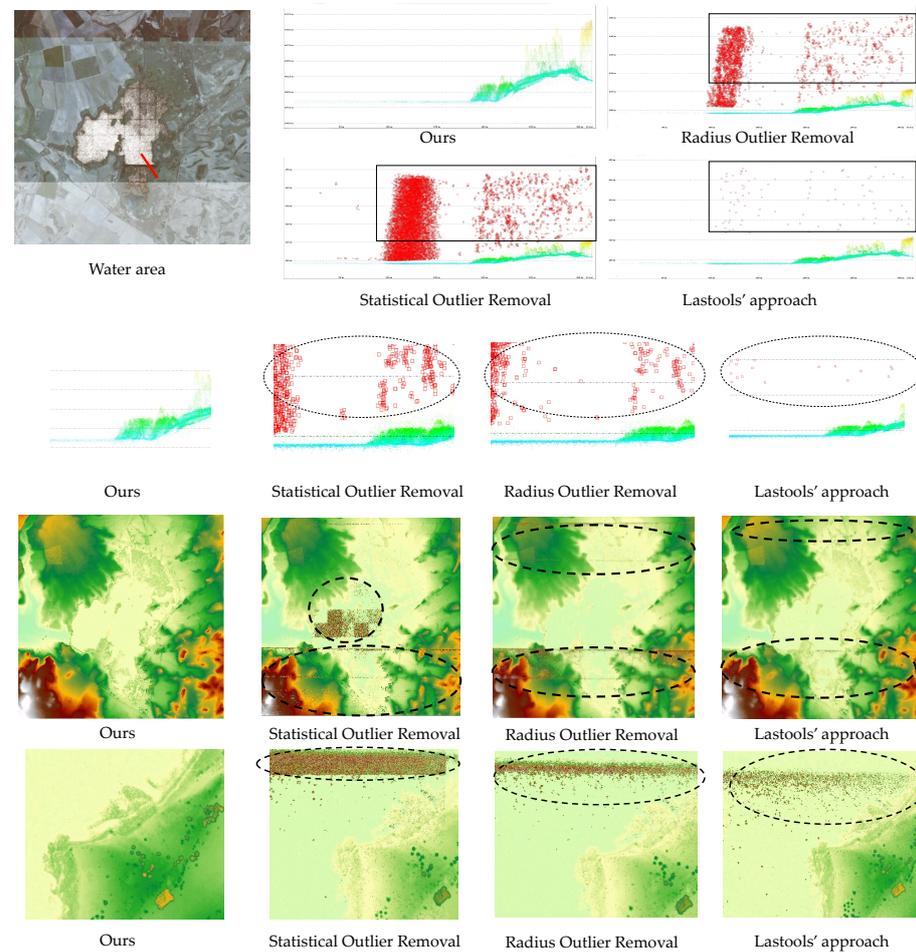


Figure 9. Examples of the denoising results in the water area. The four images in the upper right-hand corner show the profile line in the area of the red line on the left. The third row represents the profile line partial details for convenient comparison. The fourth row indicates the DSM generated by the different denoising methods. The bottom row indicates some details of the DSM to facilitate comparison.

Table 6. Performance evaluation of denoising by the proposed approach, using different scales in the urban, suburban, mountain, and water areas.

Areas	Metrics	Neighborhood (m)				
		5	10	15	5, 10	5, 10, 15
Urban	Accuracy	96.60	97.15	97.74	98.04	98.38
	F1-score	94.97	95.72	96.61	97.08	97.58
Mountain	Accuracy	94.82	95.33	95.37	96.91	98.11
	F1-score	88.86	89.81	89.98	93.16	95.70
Suburban	Accuracy	98.54	98.83	99.27	99.59	99.82
	F1-score	96.70	97.28	98.31	99.06	99.59
Water	Accuracy	77.35	79.66	79.44	82.65	83.05
	F1-score	71.46	74.35	72.71	77.49	77.92

We also visualized the denoising results for different scale features to compare the impact of multiscale features with single-scale features on the denoising precision. From the three-dimensional views shown in Figure 10, we could observe that the noise classifier trained with multiscale features was able to remove more noisy points while retaining the feature points well.

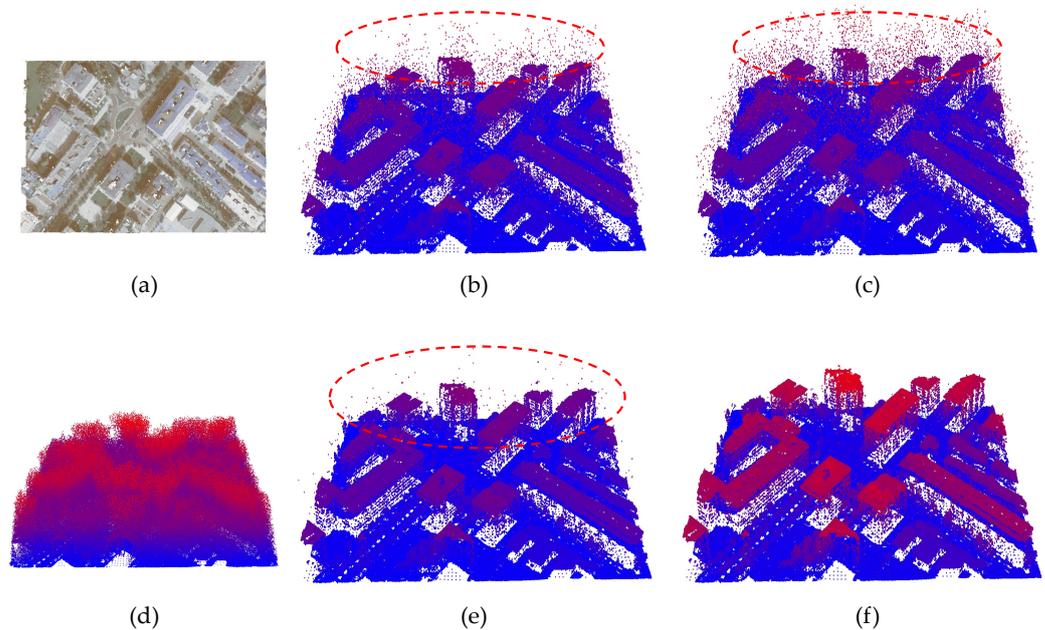


Figure 10. Analysis of single-scale and multiscale denoising results; (a) overview of the urban area; (b) three-dimensional view after denoising in the 5 m neighborhood; (c) three-dimensional view after denoising in the 10 m neighborhood; (d) raw SPL point cloud with noise; (e) three-dimensional view after denoising in the 5 m and 10 m neighborhoods; (f) three-dimensional view after denoising in the 5 m, 10 m, and 15 m neighborhoods.

Figure 11 suggests that if only single-scale features are used, the optimal neighborhood size is not the same for the different test data. In the urban and water areas, the best denoising was achieved when the neighborhood size was taken as 10 m, while for mountain and suburban areas, the best denoising was achieved when the neighborhood size was taken as 15 m. However, when using multiscale neighborhoods, we can obtain features from different neighborhoods. When using the random forests algorithm, the algorithm will preferentially select the feature with the highest discrimination by the Gini coefficients as the basis for classification. Thus, when we use a multiscale for feature learning, random forests can consider more features and can automatically perform feature selection to train a noisy point classifier that outperforms a single scale.

4.4.2. Analysis of Multistage Denoising

In point cloud processing, multistage denoising is often employed to improve the denoising effect. To verify whether the proposed approach could substantially enhance the denoising precision of SPL point cloud data after multistage denoising, we also performed a study on the accuracy of multiround denoising for the water area data where the single-round denoising results were generally poor. Table 7 shows the evaluation metrics after two denoisings, as well as the evaluation metrics improvement on the water area data using different denoising approaches. The results imply that our proposed denoising approach achieved great improvement in the recall, precision, accuracy, and F1-score after the second denoising (the accuracy was enhanced by 7.11%, and the F1-score was improved by 10.68%).

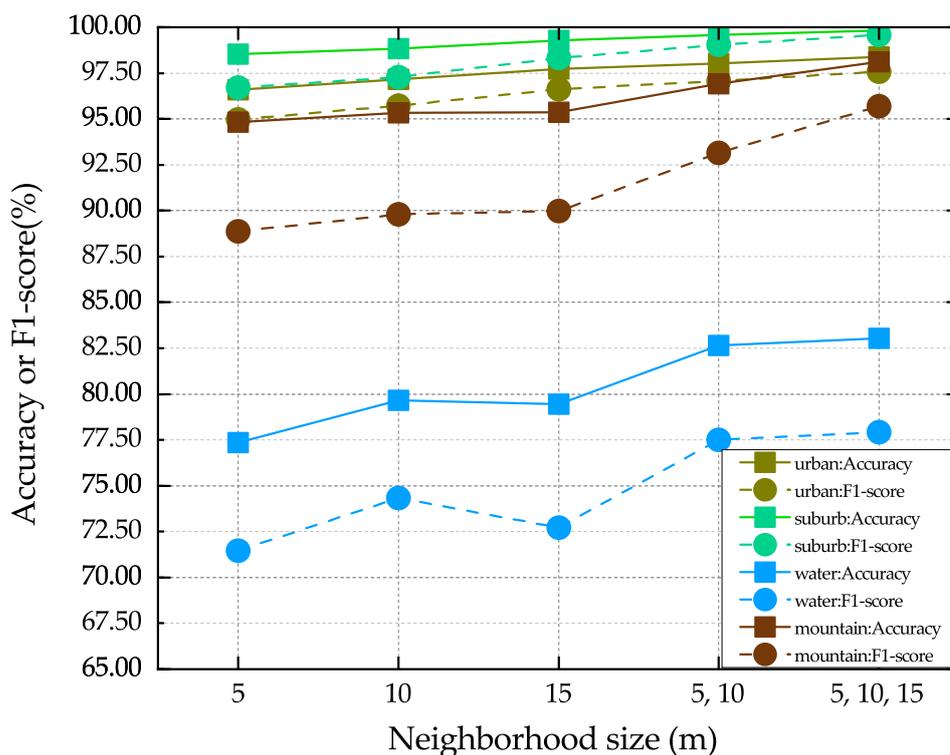


Figure 11. Visualization of the effect of single-scale versus multiscale neighborhood features on denoising results on the four datasets.

Table 7. Performance evaluation of denoising by multiple denoising, using different denoising approaches in the water area.

Methods	Metric	First Denoising	Second Denoising	Improvement
Radius Outlier Removal	Recall (%)	37.58	38.85	1.27
	Precision (%)	87.11	84.12	-2.99
	Accuracy (%)	71.57	71.35	-0.22
	F1-score (%)	52.51	53.15	0.64
Statistical Outlier Removal	Recall (%)	31.19	38.05	6.86
	Precision (%)	98.63	94.70	-3.93
	Accuracy (%)	71.04	73.20	2.16
	F1-score (%)	47.40	54.29	6.89
Lastools' Denoising Workflow	Recall (%)	45.74	47.14	1.40
	Precision (%)	98.41	97.64	-0.77
	Accuracy (%)	74.19	74.66	0.47
	F1-score (%)	62.46	63.58	1.12
Ours	Recall (%)	71.49	91.44	19.95
	Precision (%)	85.61	85.94	0.33
	Accuracy (%)	83.05	90.16	7.11
	F1-score (%)	77.92	88.60	10.68

The example results of the first denoising and the second denoising of the water area data are illustrated in Figure 12 for an intuitive comparison of the effect of multiple denoising on the denoising evaluation metrics. Figure 12d,e indicates that the proposed approach removed more noisy points after multiple denoising, while retaining the maximum amount of feature information.

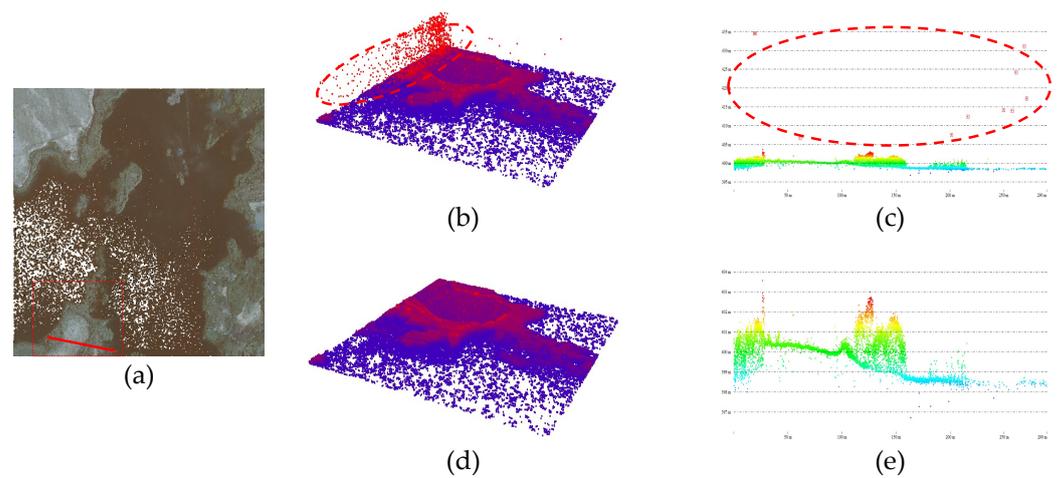


Figure 12. Visualization of the effect of the multiple denoising in the water area using our proposed approach. (a) overview of the water area; (b) three-dimensional view after the first denoising; (c) profile after the first denoising; (d) three-dimensional view after the second denoising; (e) profile after the second denoising.

Figure 13 depicts the improvement of the evaluation metrics after the second denoising using different methods. We could see that the improvement in the denoising effect was not obvious after using the existing methods to denoise the SPL point cloud data several times. In contrast, after multiple denoising, the proposed approach improved the denoising effect considerably better than the existing denoising methods.

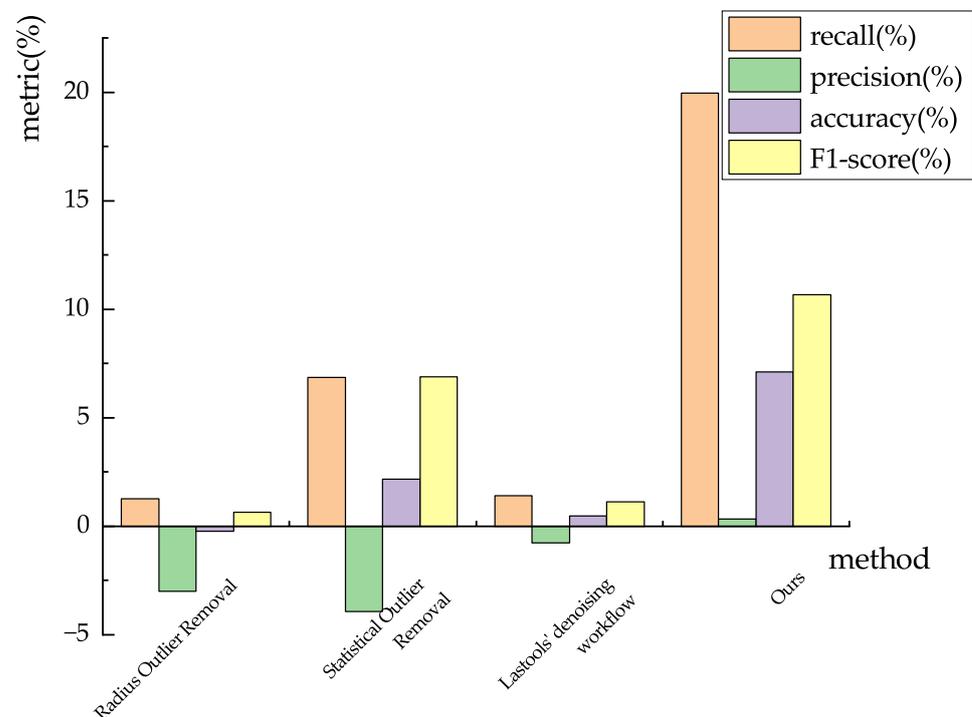


Figure 13. Visualization of the evaluation metrics improvement after multiple denoising.

5. Conclusions

This paper utilized the random forest approach and introduced multiscale features and multistage denoising methods to improve the denoising accuracy of SPL point clouds. To relieve the problem that the existing denoising methods cannot effectively remove noise points from SPL point clouds, we first extracted multiscale features of SPL point

clouds to obtain information under different receptive fields of the point clouds, and then used random forest to train the noise point classifier. Furthermore, we introduced a multistage denoising method by carrying out multiround denoising without significantly increasing the false-alarm ratio, so as to solve the problem that single-round denoising cannot effectively remove noise points in some areas.

We demonstrated the effectiveness of the proposed approach through sufficient experiments. We also proved that the proposed approach outperformed the existing denoising algorithms in the urban, suburban, mountain, and water areas. Unlike traditional denoising methods that only calculate statistical information, the proposed approach needs to calculate multiscale neighborhood features, resulting in a larger computational effort. However, from the results, the denoising accuracy of this method is much higher than traditional methods, and there is basically no need for postprocessing after denoising by the proposed approach. Furthermore, we evaluated the performance of the proposed approach using metrics such as recall, precision, accuracy, and F1-score to verify its denoising capability for SPL point cloud data. This research, which provided sufficient experiments on the denoising of SPL point cloud data, suggested that, compared with single-scale features, multiscale features were superior in improving the noise point classification process, and multiple denoising was introduced to further enhance the denoising precision. Additionally, we found that the denoising accuracy of SPL point clouds using the original deep learning network (RandLA-Net) was far better than traditional methods and may be better if improved.

Author Contributions: Conceptualization, H.H.; methodology, S.S.; software, Y.J. (Yigao Jin); validation, Y.Z. and J.C.; investigation, X.Y.; resources, Y.J. (Ying Jiang); data curation, Y.D. and X.G. (Xiaocui Guo); writing—original draft, S.S.; writing—review and editing, H.H.; visualization, X.G. (Xuming Ge). All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (Projects No: 42071355, 41871291, and 41871314) and the Project of Science and Technology Research and Development Plan of China National Railway Group Co., Ltd. with award number K2021G027.

Data Availability Statement: Data is contained within the article. The data used in this study are available in Section 4.1.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ye, Z.; Xu, Y.; Huang, R.; Tong, X.; Li, X.; Liu, X.; Luan, K.; Hoegner, L.; Stilla, U. Lasdu: A large-scale aerial lidar dataset for semantic labeling in dense urban areas. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 450. [[CrossRef](#)]
2. Mohtashami, S.; Eliasson, L.; Hansson, L.; Willén, E.; Thierfelder, T.; Nordfjell, T. Evaluating the effect of DEM resolution on performance of cartographic depth-to-water maps, for planning logging operations. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *108*, 102728. [[CrossRef](#)]
3. Jovanović, D.; Milovanov, S.; Ruskovski, I.; Govedarica, M.; Sladić, D.; Radulović, A.; Pajić, V. Building virtual 3D city model for Smart Cities applications: A case study on campus area of the University of Novi Sad. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 476. [[CrossRef](#)]
4. Kukkonen, M.; Maltamo, M.; Korhonen, L.; Packalen, P. Fusion of crown and trunk detections from airborne UAS based laser scanning for small area forest inventories. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *100*, 102327. [[CrossRef](#)]
5. Li, W.; Niu, Z.; Shang, R.; Qin, Y.; Wang, L.; Chen, H. High-resolution mapping of forest canopy height using machine learning by coupling ICESat-2 LiDAR with Sentinel-1, Sentinel-2 and Landsat-8 data. *Int. J. Appl. Earth Obs. Geoinf.* **2020**, *92*, 102163. [[CrossRef](#)]
6. Ahola, J.M.; Heikkilä, T.; Raitila, J.; Sipola, T.; Tenhunen, J. Estimation of breast height diameter and trunk curvature with linear and single-photon LiDARs. *Ann. For. Sci.* **2021**, *78*, 79. [[CrossRef](#)]
7. Li, Q.; Degnan, J.; Barrett, T.; Shan, J. First Evaluation on Single Photon-Sensitive Lidar Data. *Photogramm. Eng. Remote Sens.* **2016**, *82*, 455–463. [[CrossRef](#)]
8. Brown, R.; Hartzell, P.; Glennie, C. Evaluation of SPL100 Single Photon Lidar Data. *Remote Sens.* **2020**, *12*, 722. [[CrossRef](#)]
9. Wang, X.; Glennie, C.; Pan, Z. Adaptive noise filtering for single photon lidar observations. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 3361–3364.

10. Stoker, J.M.; Abdullah, Q.A.; Nayegandhi, A.; Winehouse, J. Evaluation of Single Photon and Geiger Mode Lidar for the 3D Elevation Program. *Remote Sens.* **2016**, *8*, 767. [[CrossRef](#)]
11. White, J.; Woods, M.; Krahn, T.; Pappasodoro, C.; Bélanger, D.; Onafrychuk, C.; Sinclair, I. Evaluating the capacity of single photon lidar for terrain characterization under a range of forest conditions. *Remote Sens. Environ.* **2021**, *252*, 112169. [[CrossRef](#)]
12. Degnan, J.J. Scanning, Multibeam, Single Photon Lidars for Rapid, Large Scale, High Resolution, Topographic and Bathymetric Mapping. *Remote Sens.* **2016**, *8*, 958. [[CrossRef](#)]
13. Mongus, D.; Žalik, B. Parameter-free ground filtering of LiDAR data for automatic DTM generation. *ISPRS J. Photogramm. Remote Sens.* **2012**, *67*, 1–12. [[CrossRef](#)]
14. Itzler, M.A.; Entwistle, M.; Wilton, S.; Kudryashov, I.; Kotelnikov, J.; Piccione, B.; Owens, M.; Rangwala, S. Geiger-Mode LiDAR: From Airborne Platforms To Driverless Cars. In Proceedings of the Applied Industrial Optics: Spectroscopy, Imaging and Metrology, San Francisco, CA, USA, 26–29 June 2017; Optical Society of America: Washington, DC, USA, 2017; p. ATu3A.3.
15. Wu, D.; Zheng, T.; Wang, L.; Chen, X.; Yang, L.; Li, Z.; Wu, G. Multi-beam single-photon LiDAR with hybrid multiplexing in wavelength and time. *Opt. Laser Technol.* **2022**, *145*, 107477. [[CrossRef](#)]
16. Abdullah, Q.A. A star is born: The state of new lidar technologies. *Photogramm. Eng. Remote Sens.* **2016**, *82*, 307–312. [[CrossRef](#)]
17. Leica. Leica SPL100 Single Photon LiDAR Sensor Data Sheet. Available online: <https://leica-geosystems.com/en-us/products/airborne-systems/topographic-lidar-sensors/leica-spl100> (accessed on 17 February 2017).
18. Mandlbürger, G.; Lehner, H.; Pfeifer, N. A comparison of single photon and full waveform LiDAR. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *4*, 397–404. [[CrossRef](#)]
19. Mandlbürger, G.; Jutzi, B. On the Feasibility of Water Surface Mapping with Single Photon LiDAR. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 188. [[CrossRef](#)]
20. Jutzi, B. Less Photons for More LiDAR? A Review from Multi-Photon Detection to Single Photon Detection. In Proceedings of the 56th Photogrammetric Week (PhoWo 2017), Stuttgart, Germany, 11–15 September 2017; p. 6S.
21. Swatantran, A.; Tang, H.; Barrett, T.; DeCola, P.; Dubayah, R. Rapid, High-Resolution Forest Structure and Terrain Mapping over Large Areas using Single Photon Lidar. *Sci. Rep.* **2016**, *6*, 28277. [[CrossRef](#)]
22. Wang, X.; Glennie, C.; Pan, Z. Weak Echo Detection from Single Photon Lidar Data Using a Rigorous Adaptive Ellipsoid Searching Algorithm. *Remote Sens.* **2018**, *10*, 1035. [[CrossRef](#)]
23. Vacek, M.; Peca, M.; Michalek, V.; Prochazka, I. Single photon laser altimeter data processing, analysis and experimental validation. *Adv. Space Res.* **2015**, *56*, 1307–1318. [[CrossRef](#)]
24. Chen, B.; Pang, Y. A denoising approach for detection of canopy and ground from ICESat-2's airborne simulator data in Maryland, USA. In *AOPC 2015: Advances in Laser Technology and Applications*; SPIE: Bellingham, WA, USA, 2015; Volume 9671, p. 96711S.
25. Zhao, Q.; Gao, X.; Li, J.; Luo, L. Optimization Algorithm for Point Cloud Quality Enhancement Based on Statistical Filtering. *J. Sens.* **2021**, *2021*, e7325600. [[CrossRef](#)]
26. Duan, Y.; Yang, C.; Chen, H.; Yan, W.; Li, H. Low-complexity point cloud denoising for LiDAR by PCA-based dimension reduction. *Opt. Commun.* **2021**, *482*, 126567. [[CrossRef](#)]
27. Zhang, H.; Zhu, L.; Cai, X.; Dong, L. Noise removal algorithm based on point cloud classification. In Proceedings of the 2022 International Seminar on Computer Science and Engineering Technology (SCSET), Indianapolis, IN, USA, 8–9 January 2022; pp. 93–96.
28. Balta, H.; Velagic, J.; Bosschaerts, W.; De Cubber, G.; Siciliano, B. Fast Statistical Outlier Removal Based Method for Large 3D Point Clouds of Outdoor Environments. *IFAC-PapersOnLine* **2018**, *51*, 348–353. [[CrossRef](#)]
29. Wang, X.; Pan, Z.; Glennie, C. A Novel Noise Filtering Model for Photon-Counting Laser Altimeter Data. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 947–951. [[CrossRef](#)]
30. Lin, R.; Hu, H.; Wen, Z.; Yin, L. Research on denoising and segmentation algorithm application of pigs' point cloud based on DBSCAN and PointNet. In Proceedings of the 2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor), Trento-Bolzano, Italy, 3–5 November 2021; pp. 42–47.
31. Zhu, X.; Nie, S.; Wang, C.; Xi, X.; Wang, J.; Li, D.; Zhou, H. A Noise Removal Algorithm Based on OPTICS for Photon-Counting LiDAR Data. *IEEE Geosci. Remote Sens. Lett.* **2021**, *18*, 1471–1475. [[CrossRef](#)]
32. Zaman, F.; Wong, Y.P.; Ng, B.Y. Density-Based Denoising of Point Cloud. In *Proceedings of the 9th International Conference on Robotic, Vision, Signal, Processing and Power Applications*; Springer: Singapore, 2017; pp. 287–295.
33. Liu, Z.; Xiao, X.; Zhong, S.; Wang, W.; Li, Y.; Zhang, L.; Xie, Z. A feature-preserving framework for point cloud denoising. *Comput.-Aided Des.* **2020**, *127*, 102857. [[CrossRef](#)]
34. Digne, J.; de Franchis, C. The Bilateral Filter for Point Clouds. *Image Process. On Line* **2017**, *7*, 278–287. [[CrossRef](#)]
35. Han, X.F.; Jin, J.S.; Wang, M.J.; Jiang, W.; Gao, L.; Xiao, L. A review of algorithms for filtering the 3D point cloud. *Signal Process. Image Commun.* **2017**, *57*, 103–112. [[CrossRef](#)]
36. Ji, C.; Li, Y.; Fan, J.; Lan, S. A Novel Simplification Method for 3D Geometric Point Cloud Based on the Importance of Point. *IEEE Access* **2019**, *7*, 129029–129042. [[CrossRef](#)]
37. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [[CrossRef](#)]
38. Wang, Y.; Chen, Q.; Liu, L.; Li, K. A Hierarchical unsupervised method for power line classification from airborne LiDAR data. *Int. J. Digit. Earth* **2019**, *12*, 1406–1422. [[CrossRef](#)]

39. Blomley, R.; Weinmann, M. Using multiscale features for the 3D semantic labeling of airborne laser scanning data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *4*, 43–50. [[CrossRef](#)]
40. Dittrich, A.; Weinmann, M.; Hinz, S. Analytical and numerical investigations on the accuracy and robustness of geometric features extracted from 3D point cloud data. *ISPRS J. Photogramm. Remote Sens.* **2017**, *126*, 195–208. [[CrossRef](#)]
41. Singh, S.; Sreevalsan-Nair, J. Adaptive Multiscale Feature Extraction in a Distributed System for Semantic Classification of Airborne LiDAR Point Clouds. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [[CrossRef](#)]
42. Che, E.; Jung, J.; Olsen, M.J. Object Recognition, Segmentation, and Classification of Mobile Laser Scanning Point Clouds: A State of the Art Review. *Sensors* **2019**, *19*, 810. [[CrossRef](#)]
43. Tomková, M.; Lysák, J.; Potůčková, M. Semantic classification of sandstone landscape point cloud based on neighbourhood features. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *43*, 333–338. [[CrossRef](#)]
44. Gallwey, J.; Eyre, M.; Coggan, J. A machine learning approach for the detection of supporting rock bolts from laser scan data in an underground mine. *Tunn. Undergr. Space Technol.* **2021**, *107*, 103656. [[CrossRef](#)]
45. Ni, H.; Lin, X.; Zhang, J. Classification of ALS Point Cloud with Improved Point Cloud Segmentation and Random Forests. *Remote Sens.* **2017**, *9*, 288. [[CrossRef](#)]
46. Vetrivel, A.; Gerke, M.; Kerle, N.; Nex, F.; Vosselman, G. Disaster damage detection through synergistic use of deep learning and 3D point cloud features derived from very high resolution oblique aerial images, and multiple-kernel-learning. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 45–59. [[CrossRef](#)]
47. Thomas, H.; Goulette, F.; Deschaud, J.E.; Marcotegui, B.; LeGall, Y. Semantic classification of 3D point clouds with multiscale spherical neighborhoods. In Proceedings of the 2018 International Conference on 3D vision (3DV), Verona, Italy, 5–8 September 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 390–398.
48. Wang, C.; Shu, Q.; Wang, X.; Guo, B.; Liu, P.; Li, Q. A random forest classifier based on pixel comparison features for urban LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2019**, *148*, 75–86. [[CrossRef](#)]
49. Yastikli, N.; Cetin, Z. Classification of LiDAR data with point based classification methods. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *XLI-B3*, 441–445. [[CrossRef](#)]
50. Lucas, C.; Bouten, W.; Koma, Z.; Kissling, W.D.; Seijmonsbergen, A.C. Identification of Linear Vegetation Elements in a Rural Landscape Using LiDAR Point Clouds. *Remote Sens.* **2019**, *11*, 292. [[CrossRef](#)]
51. Brodu, N.; Lague, D. 3D terrestrial lidar data classification of complex natural scenes using a multiscale dimensionality criterion: Applications in geomorphology. *ISPRS J. Photogramm. Remote Sens.* **2012**, *68*, 121–134. [[CrossRef](#)]
52. Dong, W.; Lan, J.; Liang, S.; Yao, W.; Zhan, Z. Selection of LiDAR geometric features with adaptive neighborhood size for urban land cover classification. *Int. J. Appl. Earth Obs. Geoinf.* **2017**, *60*, 99–110. [[CrossRef](#)]
53. Demantke, J.; Mallet, C.; David, N.; Vallet, B. Dimensionality based scale selection in 3D LiDAR point clouds. *ISPRS Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2011**, *38*, W12. [[CrossRef](#)]
54. Huang, R.; Hong, D.; Xu, Y.; Yao, W.; Stilla, U. Multi-Scale Local Context Embedding for LiDAR Point Cloud Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 721–725. [[CrossRef](#)]
55. Singh, S.K.; Raval, S.; Banerjee, B. A robust approach to identify roof bolts in 3D point cloud data captured from a mobile laser scanner. *Int. J. Min. Sci. Technol.* **2021**, *31*, 303–312. [[CrossRef](#)]
56. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
57. Blanco, J.L.; Rai, P.K. Nanoflann: A C++ header-only fork of FLANN, a library for Nearest Neighbor (NN) with KD-trees, 2014. Available online: <https://github.com/jlblancoc/nanoflann> (accessed on 2 May 2014).
58. Wright, M.N.; Ziegler, A. Ranger: A Fast Implementation of Random Forests for High Dimensional Data in C++ and R. *J. Stat. Softw.* **2017**, *77*, 1–17. [[CrossRef](#)]
59. Mandlbürger, G.; Jutzi, B. Feasibility investigation on single photon LiDAR based water surface mapping. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *IV-1*, 109–116. [[CrossRef](#)]
60. Government, N. Navarra Dataset. Available online: <https://files.cartografia.navarra.es> (accessed on 19 March 2019).
61. Hu, Q.; Yang, B.; Xie, L.; Rosa, S.; Guo, Y.; Wang, Z.; Trigi, N.; Markham, A. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 11108–11117.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.