



Article

A Method Based on Improved iForest for Trunk Extraction and Denoising of Individual Street Trees

Zhiyuan Li ¹, Jian Wang ^{1,*}, Zhenyu Zhang ¹, Fengxiang Jin ¹, Juntao Yang ¹, Wenxiao Sun ² and Yi Cao ¹¹ College of Geodesy and Geomatics, Shandong University of Science and Technology, Qingdao 266500, China² College of Surveying and Geo-Informatics, Shandong Jianzhu University, Jinan 250102, China

* Correspondence: wangj@sdust.edu.cn

Abstract: Currently, the street tree resource survey using Mobile laser scanning (MLS) represents a hot spot around the world. Refined trunk extraction is an essential step for 3D reconstruction of street trees. However, due to scanning errors and the effects of occlusion by various types of features in the urban environment, street tree point cloud data processing has the problem of excessive noise. For the noise points that are difficult to remove using statistical methods in close proximity to the tree trunk, we propose an adaptive trunk extraction and denoising method for street trees based on an improved iForest (Isolation Forest) algorithm. Firstly, to extract the individual tree trunk points, the trunk and the crown are distinguished from the individual tree point cloud through point cloud slicing. Next, the iForest algorithm is improved by conducting automatic calculation of the contamination and further used to denoise the tree trunk point cloud. Finally, the method is validated with five datasets of different scenes. The results indicate that our method is robust and effective in extracting and denoising tree trunks. Compared with the traditional Statistical Outlier Removal (SOR) filter and Radius filter denoising methods, the denoising accuracy of the proposed method can be improved by approximately 30% for noise points close to tree trunks. Compared to iForest, the proposed method automatically calculates the contamination, improving the automation of the algorithm. Our method can provide more precise trunk point clouds for 3D reconstruction of street trees.

Keywords: street tree; mobile laser scanning; iForest algorithm; trunk extraction; denoising

Citation: Li, Z.; Wang, J.; Zhang, Z.; Jin, F.; Yang, J.; Sun, W.; Cao, Y. A Method Based on Improved iForest for Trunk Extraction and Denoising of Individual Street Trees. *Remote Sens.* **2023**, *15*, 115. <https://doi.org/10.3390/rs15010115>

Academic Editor: Mohammad Awrangjeb

Received: 28 November 2022

Revised: 15 December 2022

Accepted: 22 December 2022

Published: 25 December 2022



Copyright: © 2022 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

The 3D reconstruction of real urban scenes represents a key element of digital twin cities. Street trees, as essential parts of urban ecology and landscaping, play a critical role in improving the urban water cycle and maintaining urban health. For urban street trees, the trunk is an important component whose measurement and visualization are crucial to parameter calculation, trunk surface defect detection [1], and 3D reconstruction [2]. It is therefore necessary to segment tree point clouds to extract the trunk point cloud precisely and provide accurate base data for the 3D reconstruction of real urban scenes.

Mobile laser scanning (MLS) offers an effective means to acquire data from complex urban scenes, allowing flexible and effective acquisition of dense point cloud data across an extensive urban road environment [3]. To date, many studies have investigated the use of airborne [4,5], vehicle-borne [6], and handheld scanning systems based on simultaneous localization and mapping (SLAM) [7–9] to extract individual trees. These have proposed methods for high-accuracy extraction of individual trees from particular environments. Extensive research has been conducted on tree classification, individual tree extraction [10], spatial position identification and parameter calculation (e.g., tree height [11,12], diameter at breast height (DBH) [13], canopy [14], and crown volume [15] for urban street trees [16]. The resulting parameters are also widely used for monitoring tree growth and measuring individual tree biomass [17]. Compared to TLS data, however, SLAM-based MLS point clouds are generally noisier or fuzzier, and they have lower spatial accuracy [18]. Therefore, the MLS data are a little more difficult to process.

Due to the different geometric features of the trunk and crown, the separation of trunk and crown helps in 3D model construction [19]. Tree trunks are currently extracted using a method roughly identical with detecting pole-like objects. Luo et al. [20] used a slicing strategy to identify rod-shaped features of ancient buildings from point clouds. Lehtomäki et al. [21] proposed an automatic method that was developed for the extraction of pole-like objects from vehicle-based laser scanning (VLS) data. Cabo et al. [22] horizontally sliced voxelized point clouds and analyzed the geometry of all horizontal sections to extract pole-like objects from MLS data automatically. Yu et al. [23] first separated the road points and non-road points of MLS data and then extracted street light poles semi-automatically by Euclidean distance clustering algorithm. Huang et al. [24] conducted the initial positioning of the rod-shaped ground object by using the slicing strategy. Kang et al. [25] combined an adaptive cylinder model and spatial context information to extract robustly rod-shaped ground objects.

While all these methods are able to extract pole-like structures from point clouds, few can focus on exacting tree trunks. Monnier et al. [26] extracted individual tree trunks by cylinder fitting and separated the trunk from branches and leaves to provide trunk point clouds for 3D tree reconstruction. Lamprecht et al. [27] developed an ALS-based (Airborne Laser Scanning) trunk detection algorithm, which uses a CBH (Crown Base Height) estimation and 3D-clustering approach to isolate points associated with single trunks. Li et al. [28] proposed a dual growing method for the automatic extraction of individual trees from MLS data. Xu et al. [29] proposed a method for the individual stem detection in residential environments using mobile LiDAR (Light Detection and Ranging) point clouds. Hyyppä et al. [30] tested an approach based on a pulse-based backpack laser scanner for estimating stem volumes in Finnish boreal forest conditions. Ai et al. [31] developed an automatic tree skeleton extraction approach based on multiview slicing for extracting the trunk from individual tree point cloud data. With the improvement of individual tree segmentation (ITS) accuracy, trunk–crown separation at the individual tree level shows better results. Xu et al. [32] proposed a clustering method to group points from the same stem or branch, and solve the cylinder fitting based on the random sample consensus (RANSAC) method directly. Neuville et al. [33] improved the machine learning Hierarchical Density-Based Spatial Clustering of Application of Noise (HDBSCAN) clustering algorithm to segment tree stems. Alvites et al. [34] performed stem detection and reconstruction in TLS data, which can be used to describe the tree morphology of industrial targets.

High quality of point cloud data can improve the accuracy and reproducibility of modeling. However, MLS data will inevitably cause a lot of noise in the acquisition process, including the obstruction of tree trunks by pedestrians, parked vehicles and other features in the urban environment, which will generate a lot of noise that is difficult to eliminate [35]. Therefore, denoising is an important step for trunk reconstruction in 3D, and the statistical-based method is the common method for denoising nowadays. To improve and speed up the process of outlier removal for large-scale outdoor environments, Balta et al. [36] improved the SOR algorithm and proposed Fast Cluster Statistical Outlier Removal (FCSOR). Pirotti et al. [37] use a completely open-source workflow to assess two outlier detection methods, statistical outlier removal (SOR) filter and local outlier factor (LOF) filter. Hua et al. [38] used Radius filtering to remove outliers with large differences in height from the vegetation. For tree trunk point cloud data, de Conto et al. [39] assessed three different methods of trunk denoising and three different methods of trunk modeling and found that the combination of the Hough transformations trunk denoising method and the iteratively reweighted total least-squares modeling method had the best overall performance.

After the trunk extraction and denoising, the 3D model can be constructed; the existing methods mainly focus on cylindrical fitting of the trunk. Raunonen et al. [40] proposed a new method which can quickly and automatically construct precise tree models from point clouds of the trunk and branches obtained by Terrestrial LiDAR Scanning (TLS).

The surface of the visible parts of the tree is robustly reconstructed by creating a flexible cylinder model of the tree. Méndez et al. [41] used an improved cylinder fitting approach for the 3D reconstruction of deciduous tree trunks. Liang et al. [42] assessed typical trees that are modeled by a skeleton, circle, cylinder or other geometric primitives.

Although scholars have made many breakthroughs in the processing of street tree point cloud data, there are still many issues that require improvement in the extracted trunk point cloud, including incomplete extraction and noise in the trunk. Since the row trees of MLS data are usually only a part of the scanned, mostly incomplete columns [43], it is difficult to identify the whole trunk by the method of cylinder fitting. When there is shading, such as sagging branches and leaves, the stratification process using horizontal slices is terminated prematurely and the wrong trunk canopy separation height occurs. At present, the traditional filtering denoising method is not effective in removing the noise points close to the tree trunk and is prone to excessive denoising. All these problems affect the quality of the point cloud of the trunk, which has an impact on the parameter calculation and subsequent 3D modeling of street tree. To overcome the shortcoming and obtain precise trunk point clouds, a street tree trunk extraction and denoising method has been investigated.

The paper is organized as follows: (a) an introduction to multiple techniques and their application in point cloud data processing of trees is presented in Section 1. (b) The datasets, the evaluation metrics and the extraction and denoising method for trunks are proposed in Section 2. (c) The experimental results are illustrated in Section 3. (d) The discussion is presented in Section 4, and (e) The conclusions are summarized in Section 5.

2. Materials and Methods

2.1. Study Area and Datasets

Datasets from two regions were selected for the study, one of which is a standard dataset from Lille, Paris, France which has a temperate marine climate. The Paris–Lille-3D Dataset was scanned by the mobile laser system L3D2 equipped with a 32-wire Velodyne HDL32 LiDAR [44]. A part of the Paris-Lille-3D Dataset is designated as dataset 1.

The second study area is Qingdao, Shandong, China which is characterized by a monsoon climate of medium latitudes. The Trimble MX9 scanning system was used to obtain dataset 2, dataset 3 and dataset 4 for a street in the city and a street on campus. The Trimble MX9 vehicle-borne scanning system has two angle-adjustable laser scanners and lateral cameras and a 360° panoramic camera, the parameters of which are shown in Table 1. Dataset 5 was collected by a handheld scanner GEOSLAM ZEB-Horizon on a campus in Qingdao, and some of the data were selected for processing and validation.

Table 1. Some of the parameters of the two scanners.

Trimble MX9			GEOSLAM ZEB-Horizon	
Datalogger Carrier	Vehicle-Borne		Datalogger Carrier	Backpack or Shoulder Strap
Scan speed	500 scans/sec		Scanner points per second	300,000
Laser class	1, eye-safe		Laser class	1, eye-safe
Maximum range target Reflectivity > 80% ²	475 m	370 m	235 m	Range 100 m
Maximum range target Reflectivity > 10% ²	170 m	130 m	85 m	Vertical angular resolution 2°
Minimum range	1 m @ PRR ≥ 1 MHz, 1.2 m @ PRR < 1 MHz		Horizontal angular resolution	0.2°
Accuracy ³ /precision ⁴	5 mm/3 mm		Raw data file size	25–50 MB/min
Field of view	360° “full circle”		Relative accuracy	Up to 6 mm
			Field of View	360° × 270°

The datasets were collected in Paris–Lille in France, an urban street in Qingdao and a campus street in Qingdao. The three scenes have different types of trees and growth states due to different geographical locations, climates, and seasons. Scenes vary in complexity,

data are acquired by different Vehicle-borne LiDAR, and thus data quality varies. The five datasets provide a wide variety of data that make it possible to validate the feasibility and applicability of our algorithm. The MLS data include mainly ground, road, bushes, pedestrians, streetlights, buildings, street trees, etc. Figure 1 shows an overall view of the datasets. Datasets 1, 2 and 3 are broad-leaved species, while datasets 4 and dataset 5 are coniferous species. The five datasets have 14, 48, 29, 19 and 14 trees respectively. In order to better represent the characteristics and differences between the different datasets, the mean tree height, DBH and canopy of the five data sets are presented in Table 2 through software processing and manual measurements. These three values can characterize the growth dynamics of the trees. The DBH calculated by extracting a point cloud slice at a tree height of 1.2 m and the fitted radius was obtained using least-squares circle fitting [40]. The canopy is obtained by calculating the average of the north-south and east-west widths of an individual tree after first projecting it horizontally [15].

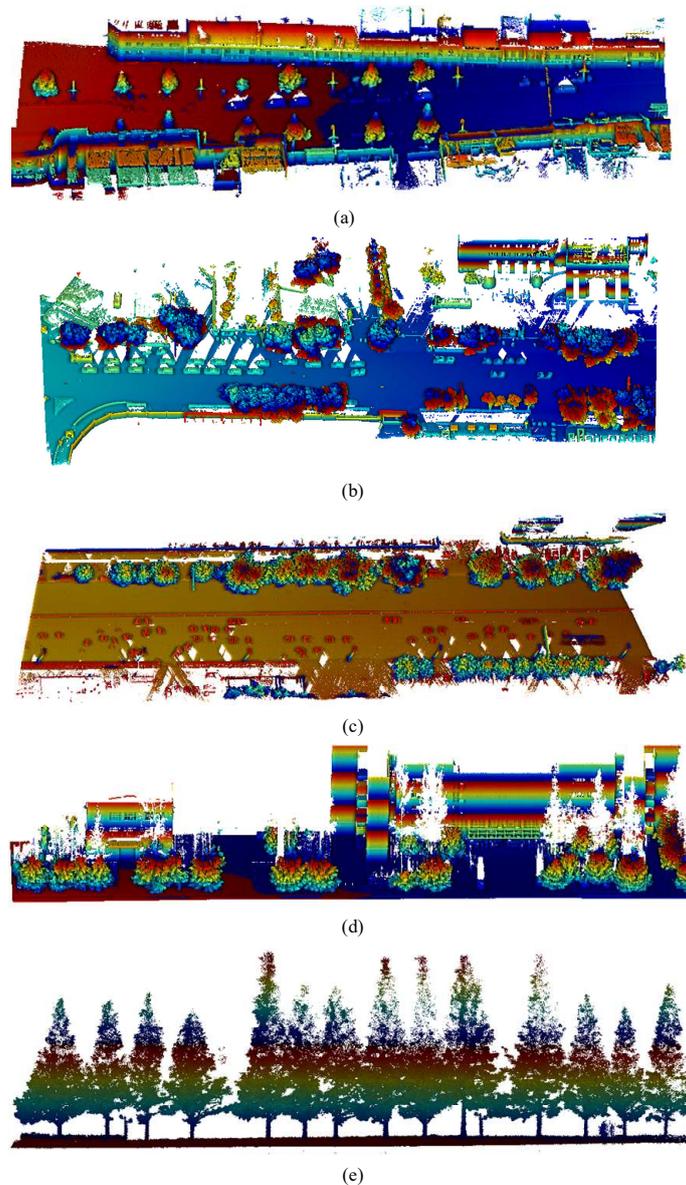


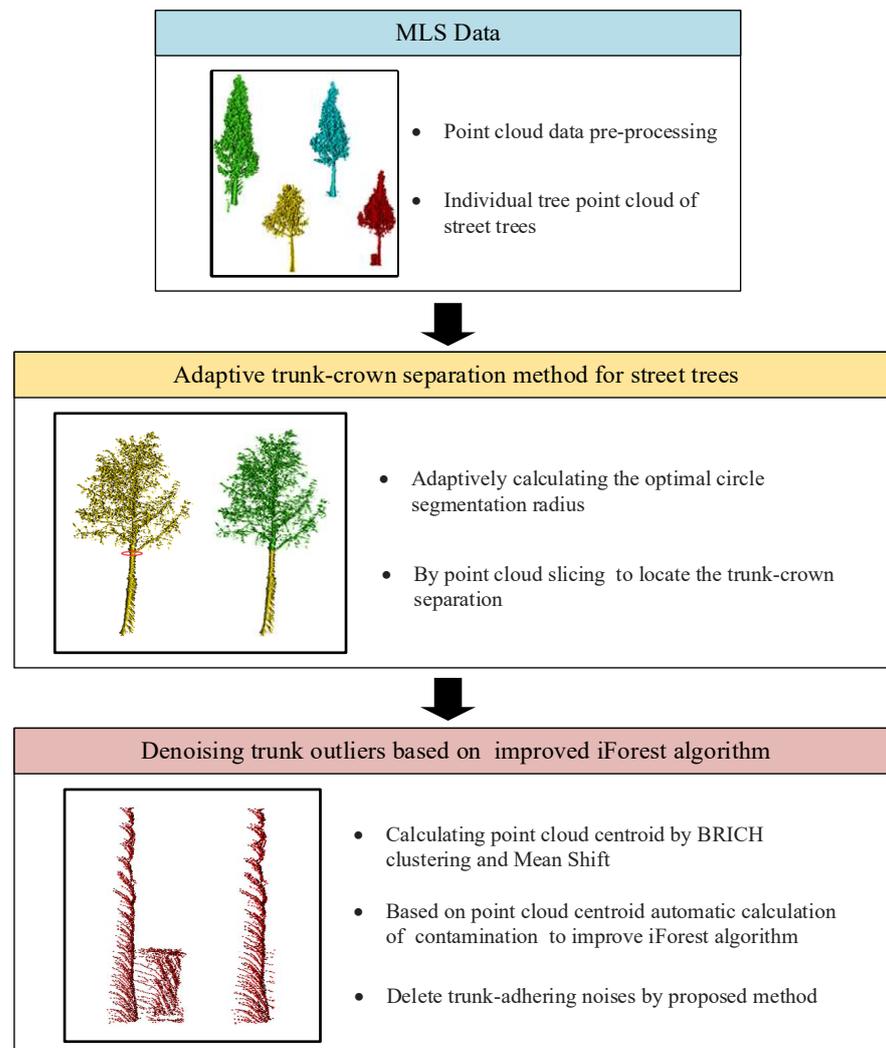
Figure 1. Original point cloud data: (a) dataset 1; (b) dataset 2; (c) dataset 3; (d) dataset 4; (e) dataset 5.

Table 2. Street tree information in 5 datasets.

Dataset	Category	Number	Mean Tree Height	Mean DBH	Mean Canopy
1	broad-leaved forest	14	11.31 m	0.36 m	3.91 m
2	broad-leaved forest	48	9.46 m	0.16 m	7.42 m
3	broad-leaved forest	29	11.06 m	0.15 m	8.35 m
4	coniferous forest	19	10.84 m	0.94 m	6.73 m
5	coniferous forest	14	16.80 m	0.17 m	6.40 m

2.2. Overview of Methods

This method focuses on the extraction and denoising methods of street tree trunks in order to obtain complete and more refined trunk point clouds. As shown in Figure 2, in order to improve the integrity of the trunk extraction, the individual tree was first sliced horizontally during trunk extraction [28]. Then, the reference slicing is selected by calculating the Root-Mean-Square Error (RMSE) of the horizontal slicing. The reference slicing is able to provide the center and radius of the optimal circle segmentation, which can help extract the part of the trunk that is obscured by branches and leaves. Finally, the improved iForest algorithm is used to remove the noise close to the extracted tree trunk.

**Figure 2.** The workflow of the proposed method.

2.3. Evaluation Metrics

For trunk–crown separation and trunk denoising, the experimental result is evaluated by *Precision* and *Recall*, *F-score* [45] as shown in Equations (1)–(3):

$$Precision = \frac{TP}{FP + TP} \quad (1)$$

$$Recall = \frac{TP}{FN + TP} \quad (2)$$

$$F\text{-score} = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (3)$$

When evaluating the trunk extraction result, *TP* is the number of correctly extracted trunks, *FP* is the number of incorrectly extracted trunks, and *FN* is the number of trunks not recognized in the result. When evaluating the outlier deletion, *TP* is the number of correctly segmented noises, *FP* is the number of incorrectly segmented trunk point clouds (i.e., the number of over segmented point clouds), and *FN* is the number of unrecognized noises. *F-score* can balance the effects of *Precision* and *Recall*. It evaluates the segmentation and denoising effects more comprehensively, and a larger value indicates better results.

2.4. The Method of Trunk-Crown Separation

Trunk-crown separation for street trees is normally performed based on geometric features. However, when slicing individual trees to locate the geometric mutation, the existence of drooping branches or shrubs at the bottom of a street tree can affect any slicing method used alone. To solve these problems, we propose an adaptive method to determine the optimal circle segmentation radius to improve the point cloud slicing which can obtain horizontal sections to discriminate the separation position of the tree trunk. The effect of nesting is achieved by adaptive optimal circle partitioning, which can basically solve the above situation. The workflow of trunk–crown separation is shown in Figure 3.

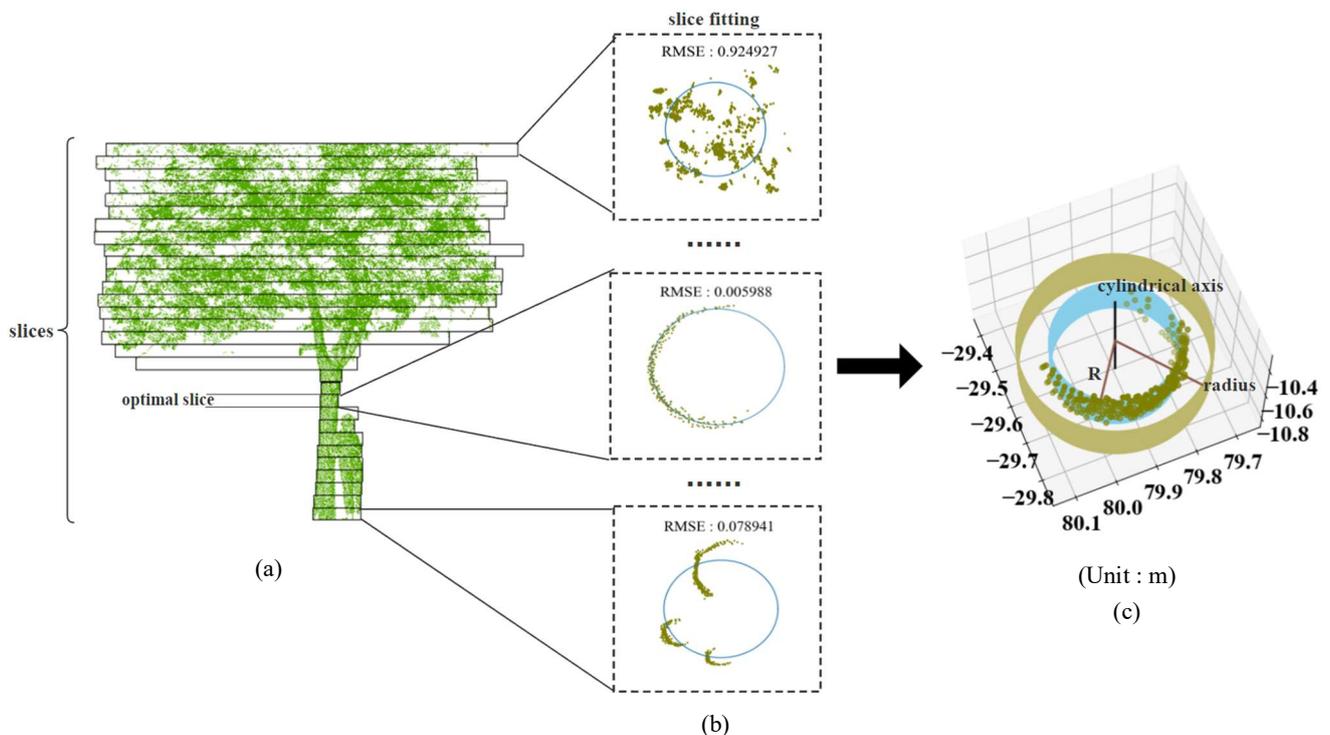


Figure 3. The workflow of trunk–crown separation: (a) individual tree layered slices; (b) the optimal slice is selected under the condition of minimum RMSE; (c) optimal circle segmentation.

2.4.1. Adaptively Optimal Circle Segmentation

To remove interference by nesting cylinders, the cylindrical axis and radius of the nested cylinders need to be determined. Since the point cloud distribution of tree crown is scattered and disorderly, while the point cloud distribution of tree trunk is approximately circular, the geometric information can be obtained by fitting the point cloud of the tree trunk slice. Firstly, the individual tree point cloud is sliced into layers. The purpose of this slicing process is to obtain the reference slice which can provide the axis and radius of the circle segmentation. In order to reduce the interference of all kinds of miscellaneous points, the thickness of the initial slice processing should not be too great, and the thickness of the slice can be taken as 0.1 m. The slices are respectively extracted from the lowest point of the individual tree along the Z-axis direction upwards. Considering that the trunk is concentrated in the lower half of the tree, only half of the individual tree point cloud needs to be sliced, as shown in Figure 3a. Then, in order to obtain the optimal trunk slice as the reference slice of the cylindrical nesting, each slice is projected onto the horizontal plane in turn to obtain the set of projected points A_i .

$$A_i = \{(x_i, y_i) \mid (x_i, y_i), i = 1, 2, \dots, n\} \tag{4}$$

The least-squares (LS) method is used to fit a circle to the point set. The center of the circle (C_x, C_y) and the radius R of each slice can be calculated; this is also how trunk DBH is calculated for most cases [46]. Then, the Root-Mean-Square Error (RMSE) can be calculated by Equation (5).

$$RMSE = \sqrt{\frac{\sum_{i=1}^n ((x_i - C_x)^2 + (y_i - C_y)^2 - R^2)}{n}} \tag{5}$$

The fitting results and RMSE are shown in Figure 4. It can be seen that the RMSE responds to the accuracy of the circle fit. The RMSE of slices disturbed by appendages or sagging branches will be large. Therefore, the slice with the smallest RMSE can be used as the reference slice, as shown in Figure 3b.

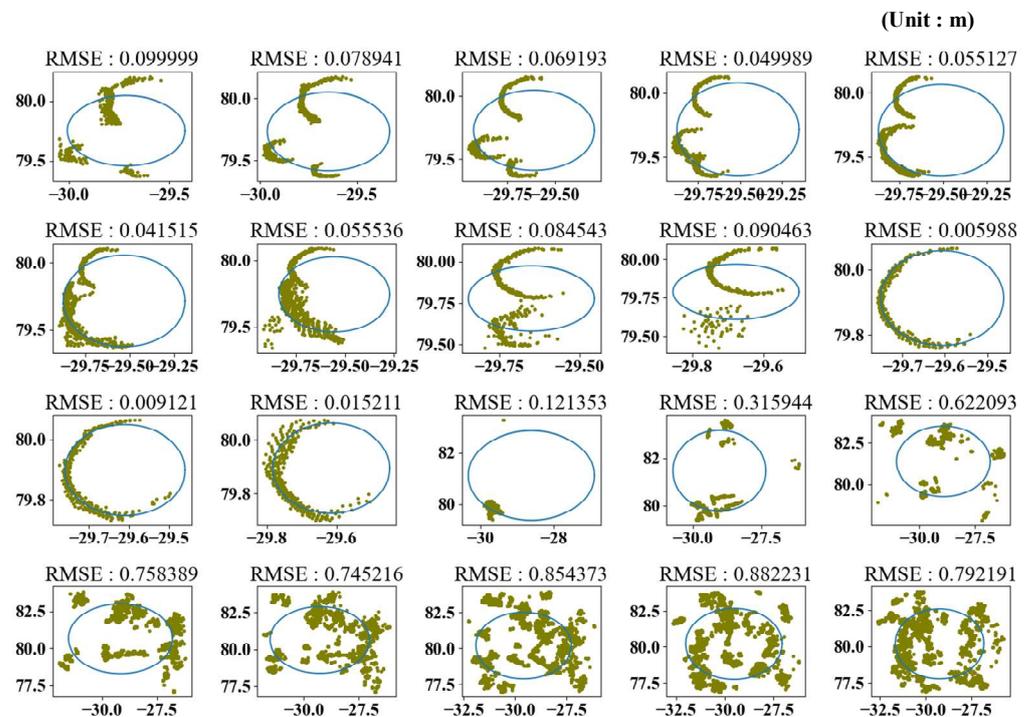


Figure 4. The fitting results and RMSE of slices.

The reference slice will provide the cylindrical axis and radius for optimal circle segmentation. It is important to consider that the tree trunk is not a completely vertical pole

and the degree of thickness in the trunk is not fixed. In order to maximize the preservation of the complete trunk, the radius of optimal circle segmentation needs to be slightly enlarged on the basis of R . This circle segmentation deletes most of the non-trunk point clouds and retains a complete trunk point cloud, as shown in Figure 3c.

The reference slice is adaptively determined by obtaining the RMSE of each slice to provide the optimal axis and radius for the circle segmentation. This is not affected by differences in tree species and does not require artificially specifying the radius of the circle segmentation. The optimal circle segmentation method can eliminate most of the disturbances while preserving the trunk, making the difference between the geometric features of the trunk and the crown more obvious. Therefore, a good basis for the extraction of more complete trunks is provided by the slicing method.

2.4.2. Trunk–Crown Separation Based on Slice

The remaining point cloud after the circle segmentation is sliced horizontally again and retrieved from the bottom up to find the location of the separation between the trunk and the crown. The thickness of the slices should not be too great; 0.05 m is more appropriate after experimental verification. Finally, the fitting radius of each point cloud slice is calculated, and the slice with a sudden increase in fitting radius is sought to determine the height of trunk–crown separation.

As shown in Figure 5a, b, the red line represents the position of geometric mutation. Due to the influence of drooping branches and leaves, direct use of the slice method can lead to incomplete trunk–crown separation. Figure 5c is the result of extracting the trunk directly using the slicing method for Figure 5a, while Figure 5d is the result of extracting the trunk using the slicing method for Figure 5b which after circle segmentation. By comparing Figure 5c with Figure 5d, however, the proposed method can effectively reduce the influence of this kind of situation and obtain a more complete trunk point cloud.

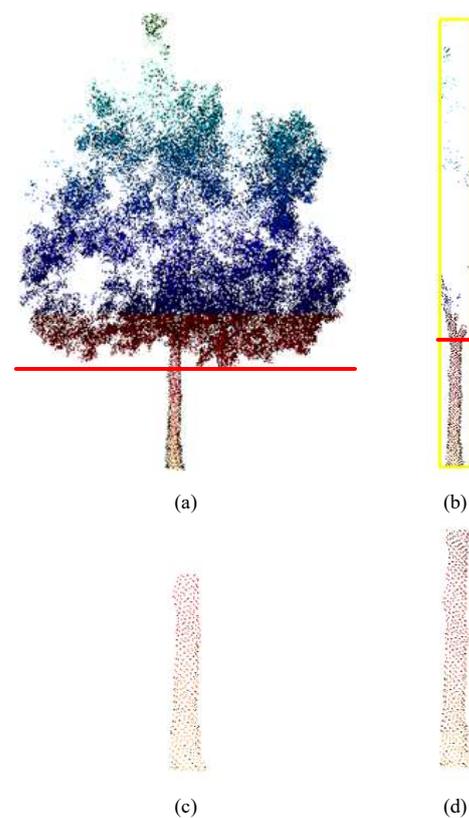


Figure 5. Trunk–crown separation: (a) a street tree; (b) result of circle segmentation; (c) trunk obtained by direct slicing; (d) trunk obtained by proposed method.

2.5. Tree Trunk Point Cloud Denoising

Street trees often have ground objects close to the trunk, which are difficult to eliminate during individual tree segmentation (ITS) and can affect the accuracy of the trunk diameter at breast height (DBH) and the modeling performance. Outliers in point clouds are usually characterized by a sparse and scattered distribution, far from the target point cloud with high density. In the individual tree point cloud of street tree, there are often trunk-adhering noises that are difficult for a traditional algorithm to remove. To address this problem, this paper proposes an outlier segmentation method based on the Balanced Iterative Reducing and Clustering Using Hierarchies (BIRCH) algorithm and the improved iForest algorithm based on the spatial distribution characteristics of point cloud outliers and their density characteristics. This method separates the outliers from the target point cloud by determining whether they are outliers or not based on the distribution characteristics of the point cloud.

2.5.1. BIRCH Clustering

BIRCH is a type of rapid hierarchical clustering. BIRCH only needs a single-pass scan of the dataset to cluster and build a clustering feature tree (CF Tree) according to the clustering features (CF) [47,48]. CF and CF Tree are defined as follows:

CF is the core of the BIRCH algorithm and is a triplet, expressed in (N, LS, SS) , representing all the information of a single clustered cluster. N represents the number of sample points in the CF. LS represents the sum vector of each characteristic dimension of the sample point. SS represents the sum of squares of each characteristic dimension of the sample point. This information allows calculation of the centroid and radius values for each cluster. The centroid and radius of clusters containing n points X_i ($i = 1, 2, 3, \dots, n$) are calculated as Equations (6) and (7):

$$Centroid = \frac{(X_1 + X_2 + \dots + X_n)}{n} \quad (6)$$

$$Radius = \frac{|X_1 - Centroid|^2 + |X_2 - Centroid|^2 + \dots + |X_n - Centroid|^2}{n} \quad (7)$$

CF Tree is a compressed representation of a dataset, which is essentially a B^+ tree, with several CF per node, including leaf nodes. It has several important parameters: internal node equilibrium factor B (maximum number of CF per internal node), leaf node equilibrium factor L (maximum number of CF per leaf node) and Cluster radius threshold T (maximum sample radius threshold per CF in each leaf node).

The BIRCH algorithm can set its own spherical cluster diameter threshold. Within a sufficiently small radius threshold, the processed point cloud data are divided into numerous homogeneous groups. Even if the outlier point and the target point cloud are closer, the target point cloud and the outlier point can be separated into different clusters, and the purpose of separating the outlier point from the target point cloud can be realized.

To validate the superiority of this clustering method, an experiment is carried out on four trunks with much noise. Figure 6 shows the original point cloud data and the centroid points of the clusters after the original point cloud is clustered. Compared with the original point cloud, the centroid data not only basically retain the spatial distribution of the original point cloud, but also largely reduce the number of points, thus improving subsequent data processing efficiency.

2.5.2. Improved iForest Algorithm

iForest is an unsupervised anomaly detection method for discrete data. It uses multiple dichotomies to partition all sample points until each sample point or very few sample points are divided into the same region [49,50]. iForest does not require labeled samples to train, but features need to be continuous. In general, clusters with higher densities need to be cut very many times to be isolated, while points with low densities are more likely

to be isolated. As can be seen from Figure 7, normal points need to be cut multiple times before they are separated, while points that appear abnormal can be separated with only a few cuts.

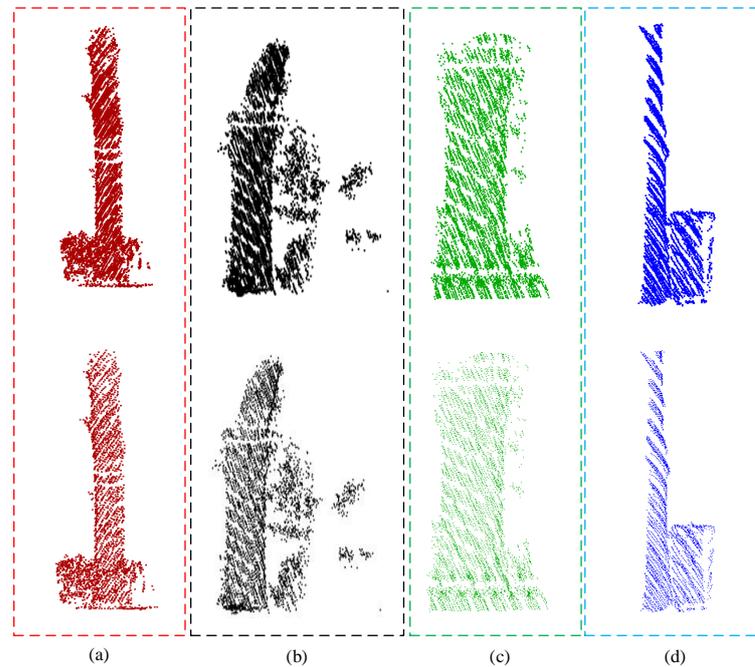


Figure 6. BIRCH calculation result of four samples: Four trunks with much noise: (a) Sample 1; (b) Sample 2; (c) Sample 3; (d) Sample 4. The top half of the graph shows the original data and the bottom half is the results of BIRCH clustering.

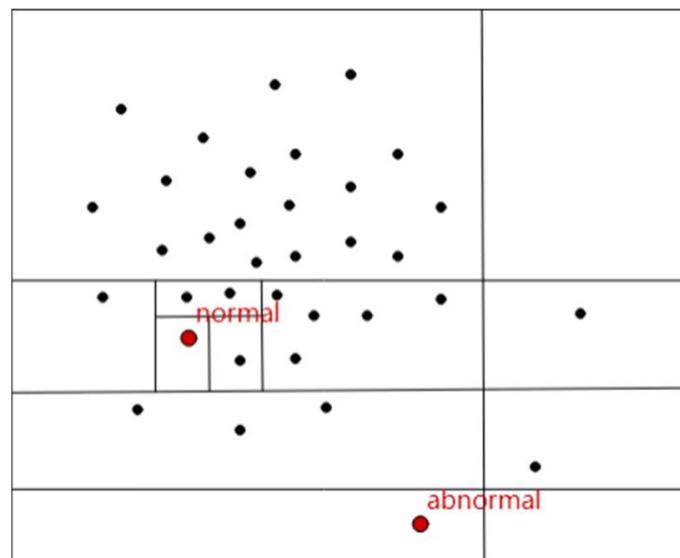


Figure 7. iForest to determine the outlier: characteristics of abnormal points: few and different.

In the iForest algorithm, the contamination value is an important hyperparameter that represents the approximate proportion of outliers in the dataset. However, most of the current contamination values need to be specified by continuous commissioning and are not highly automated. For this problem, this paper designs a method to automatically calculate the pollution value by using the centroid data obtained after processing by BIRCH clustering algorithm.

First of all, we calculate the point with the highest neighbor density among all centroid points on XOY plane by means of a mean shift algorithm. The formula of the mean shift algorithm is as follows[M1] :

$$m_h(X) = \frac{\sum_{i=1}^n X_i g\left(\left\|\frac{X-X_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{X-X_i}{h}\right\|^2\right)} - X_c \quad (8)$$

$$X_{c+1} = X_c + m_h(X) \quad (9)$$

where X_c is the center point; X_i is the point within the bandwidth range; n is the number of points in the bandwidth range; and $g(X)$ is the negative derivative of the kernel function. By iteratively calculating the offset $m_h(X)$ between the center point and the centroid point in the bandwidth range, the iteration is stopped until X_c and X_{c+1} are very close, so as to find the point with the maximum neighbor density.

Figure 8 shows the distribution of individual centroids and their maximum neighbor density point on a two-dimensional plane for data in Figure 6 as an example. As the centroid density of the outlier cluster is much smaller than that of the target point cloud cluster, its maximum neighbor density point is closer to the centroid of the target point cloud. Next, we calculate the Euclidean distance from each centroid to the maximum neighbor density point. The Euclidean distance with coordinates (X_i, Y_i) and (X_j, Y_j) is calculated by the formula below:

$$D(i, j) = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2} \quad (10)$$

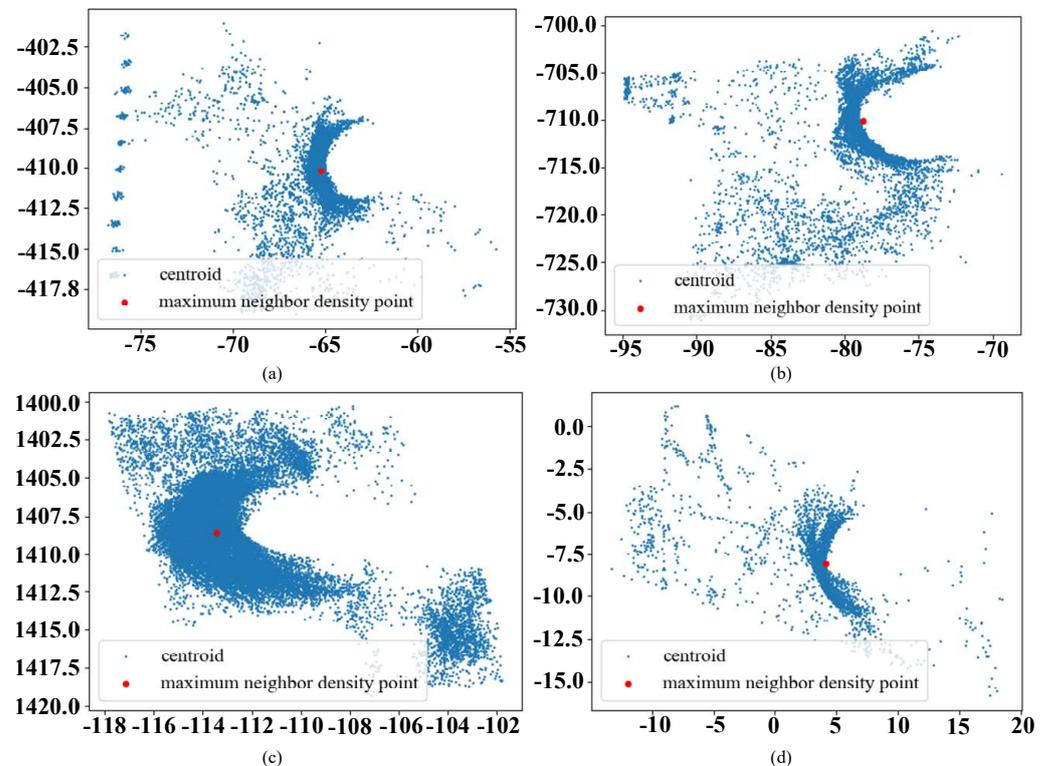


Figure 8. Maximum neighbor density point in four samples. (a) Sample 1 of Figure 6a; (b) Sample 2 of Figure 6b; (c) Sample 3 of Figure 6c; (d) Sample 4 of Figure 6d.

All Euclidean distance calculations are performed to obtain a histogram of the distribution as shown in Figure 9 and a line chart of the change of “points-distance”. Since the centroid after clustering of the target point cloud and the centroid after clustering of

outliers have obvious differences in point density, it can be seen from the “point-distance” line change chart of Figure 9 that, as the distance increases, the number of corresponding points has a more pronounced change.

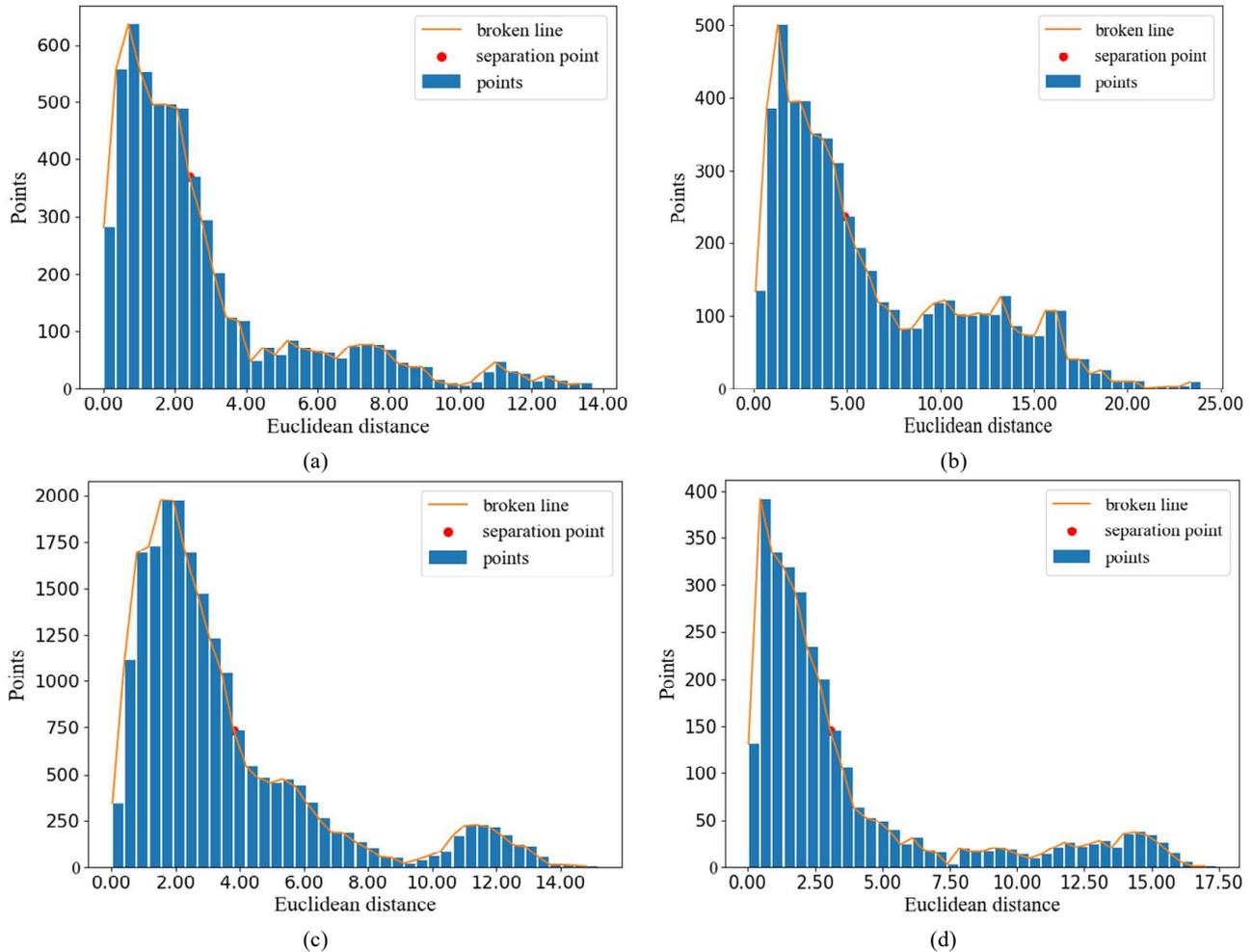


Figure 9. Points–distance line charts of four samples. (a) Sample 1 of Figure 6a; (b) Sample 2 of Figure 6b; (c) Sample 3 of Figure 6c; (d) Sample 4 of Figure 6d.

After obtaining all statistical data, we calculate the slope of each straight line in the broken line and identify the broken line with the largest slope, whose separation point is represented by the red point. The point number range is the largest in the interval where this broken line is located. The farther backward, the greater the distance. Hence, it can be concluded that the sum of all point numbers before this straight line is approximately the total number of all centroids after target point clustering, set as Sum_t , and the sum of all point numbers after that is approximately the total number of centroids after outlier clustering, set as Sum_f . From this, we can approximately derive contamination c :

$$c = \frac{Sum_f}{Sum_t + Sum_f} \quad (11)$$

2.5.3. Trunk Denoising Method

To sum up, an individual tree trunk is processed by BIRCH algorithm to paste the tag value to the original point cloud and obtain the centroid data for all clusters. Contamination c is calculated by the method described in Section 2.5.2 above. Then, the outliers of all centroids are identified by iForest algorithm and their tag values are calculated. The noises

in the trunk point cloud are deleted by deleting the corresponding points in the original point cloud according to the tag values.

Denoising experiments were carried out for the four tree trunks in Figure 5. The proposed method, iForest, Radius Filter and Statistical Outlier Removal (SOR) Filter were used to denoise each trunk, respectively, and the denoising accuracy was evaluated based on the manual denoising results. The denoising effect of Figure 6 is shown in Figure 10. It can be compared from the effect graph of denoising first. Since the noise points close to the target are closely connected to the target, are not easily distinguishable in spatial location, and the point clouds of the two are similar in density, both the statistically based Radius Filter and SOR Filter perform poorly. Some of the outlier points can be removed, but it is basically impossible to eliminate the noise points of this type. While both the proposed method and iForest can achieve better results because they can reject outliers, the traditional iForest requires constant manual debugging of parameters to achieve better results. Using the result of manual denoising as the true value, the evaluation results are shown in Table 3 and Figure 11. Comparing the average Precision, Recall and F-score of the four data sets, the proposed method gives better results. Its Precision is about 30% higher than SOR Filter and about 60% higher than Radius Filter. Compared with the iForest, not only is the contamination automatically calculated, but the denoising accuracy is also improved.

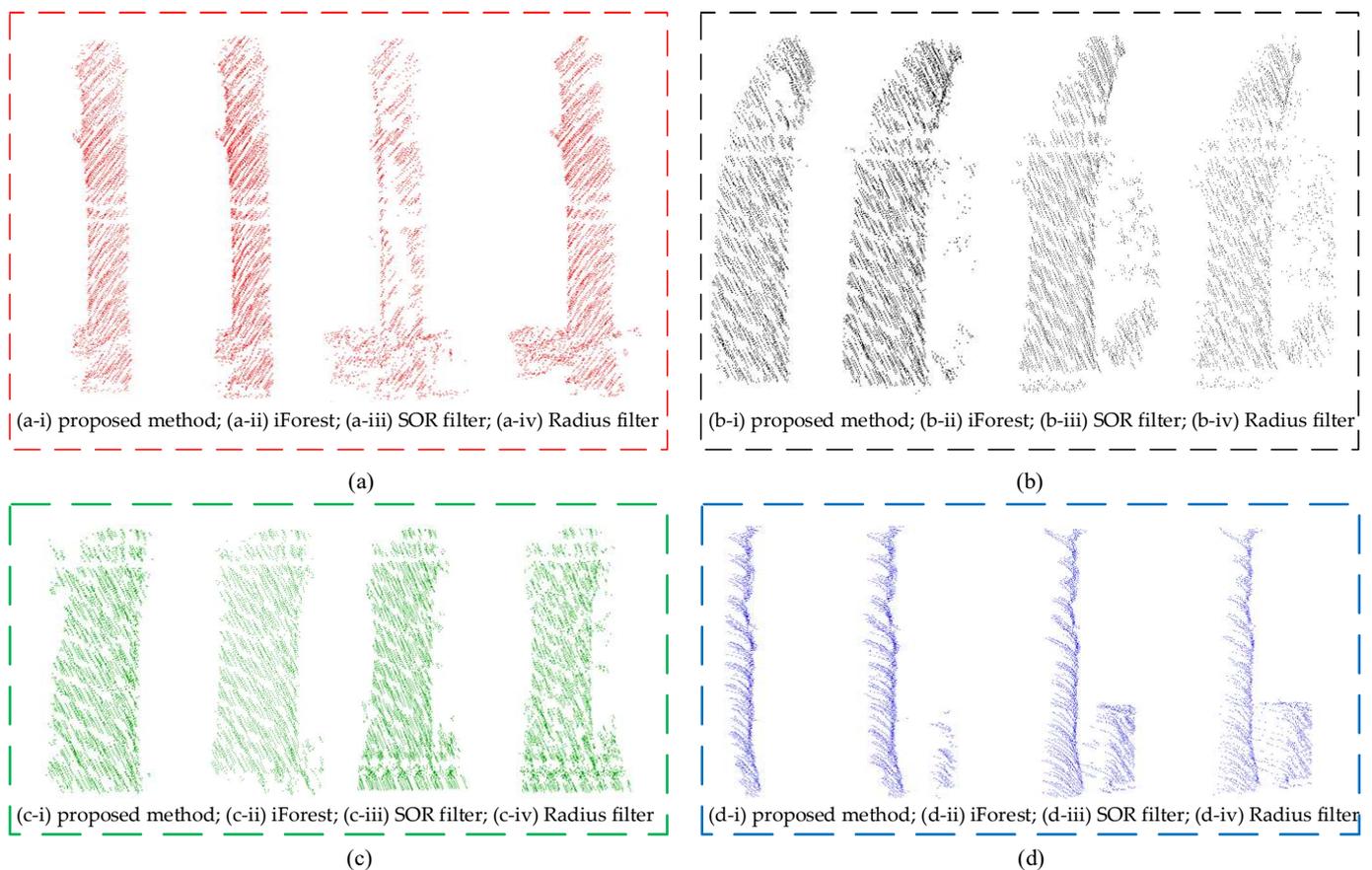
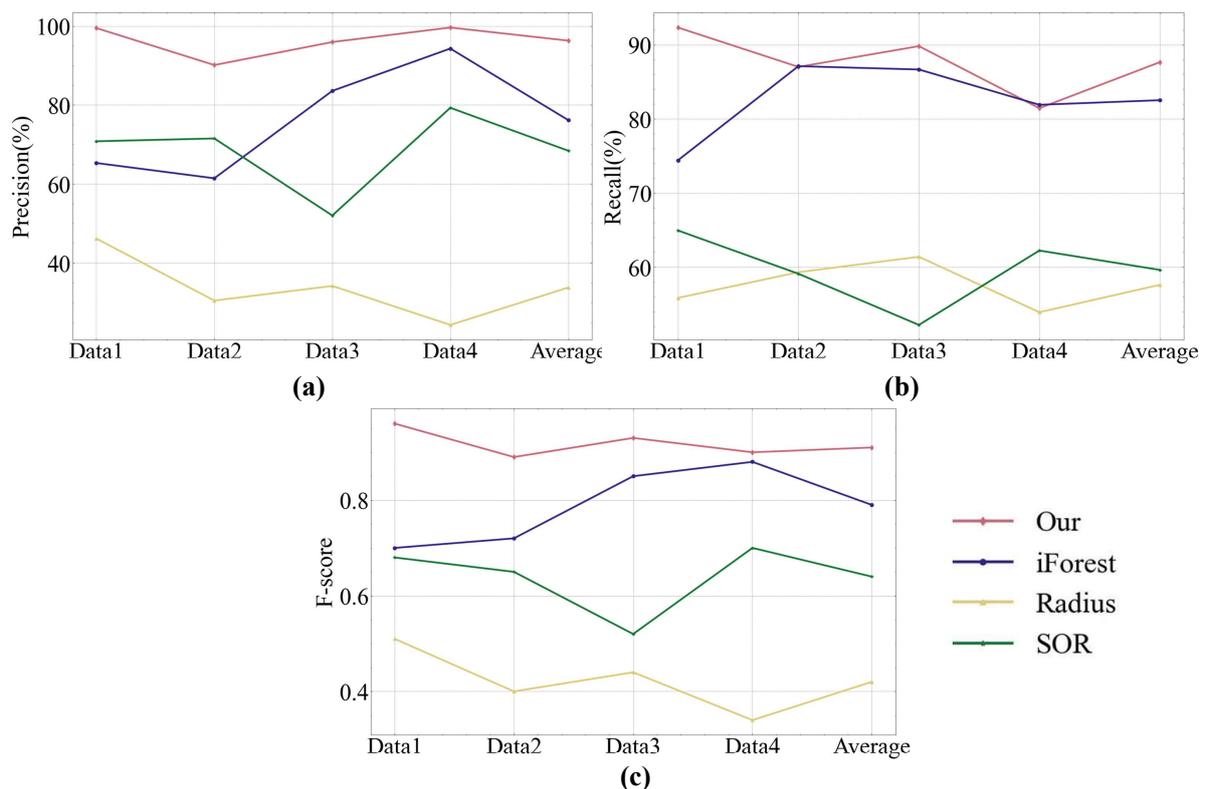


Figure 10. Denoising effect comparison: (a–d) The four denoising effects in Figure 6a–d.

Table 3. Accuracy evaluation of trunk denoising experiment.

Data	Methods	TP	FP	FN	Precision (%)	Recall (%)	F-Score
1	Proposed method	1540	8	128	99.48	92.32	0.96
	iForest	1320	701	454	65.31	74.41	0.70
	Radius Filter	606	706	479	46.19	55.85	0.51
	SOR Filter	1223	504	660	70.81	64.95	0.68
2	Proposed method	1641	180	245	90.11	87.01	0.89
	iForest	1724	1082	255	61.44	87.11	0.72
	Radius Filter	487	1111	334	30.48	59.32	0.40
	SOR Filter	1165	464	806	71.52	59.11	0.65
3	Proposed method	2036	86	231	95.94	89.81	0.93
	iForest	1696	333	261	83.58	86.66	0.85
	Radius Filter	999	1923	628	34.19	61.40	0.44
	SOR Filter	1560	1439	1428	52.02	52.21	0.52
4	Proposed method	4125	16	940	99.61	81.44	0.90
	iForest	3800	230	840	94.29	81.90	0.88
	Radius Filter	1060	3299	906	24.32	53.92	0.34
	SOR Filter	3063	800	1858	79.29	62.24	0.70
Average	Proposed method				96.29	87.65	0.91
	iForest				76.16	82.52	0.79
	Radius Filter				33.79	57.62	0.42
	SOR Filter				68.41	59.63	0.64

**Figure 11.** Comparison of the accuracy of the four denoising methods. (a) Precision comparison; (b) Recall comparison; (c) F-score comparison.

3. Experiment and Results

3.1. Data Pre-Processing

Paris-Lille-3D Dataset has a large amount of data, so we chose Lille-6 as dataset 1 for the experiment. Lille-6 is located in the city and the street's trees are more evenly distributed. The point cloud quality is also better due to less shading. High vegetation from point cloud

data is extracted by Trimble Realworks Software, then individual trees are extracted by the CHM (Canopy Height Model)-based ITS algorithm in the LiDAR360 software. Since the purpose of the experiment is to verify the trunk extraction and denoising process, the necessary manual processing is performed for the misclassification and omission generated by the ITS algorithm. Subsequently, the experimental processing of individual tree point clouds in these scenes is carried out.

3.2. Experimental Results

3.2.1. Street Tree Trunk-Crown Separation

The five datasets are processed using the method described in Section 2 above. First, trunk point clouds are extracted by trunk-crown separation of individual street trees, as shown in Figures 12–16. The results of trunk-crown separation are shown in Figures 12–16 and Table 4. Table 4 shows the evaluation metrics of trunk extraction for the five datasets. For dataset 1 as shown in Figure 12, the trees are widely spaced from each other and have less noise. This is all beneficial to the extraction of the trunk point cloud. As a result, there were no incorrect segmentations among the 14 broad-leaved street trees, and the *Precision* is 100%. However, there is one tree that is undivided (Figure 12c), and the *Recall* is 92.85%. The specific reasons for not being segmented are explained in the Discussion section. The *F-score* of the dataset 1 is 0.96.

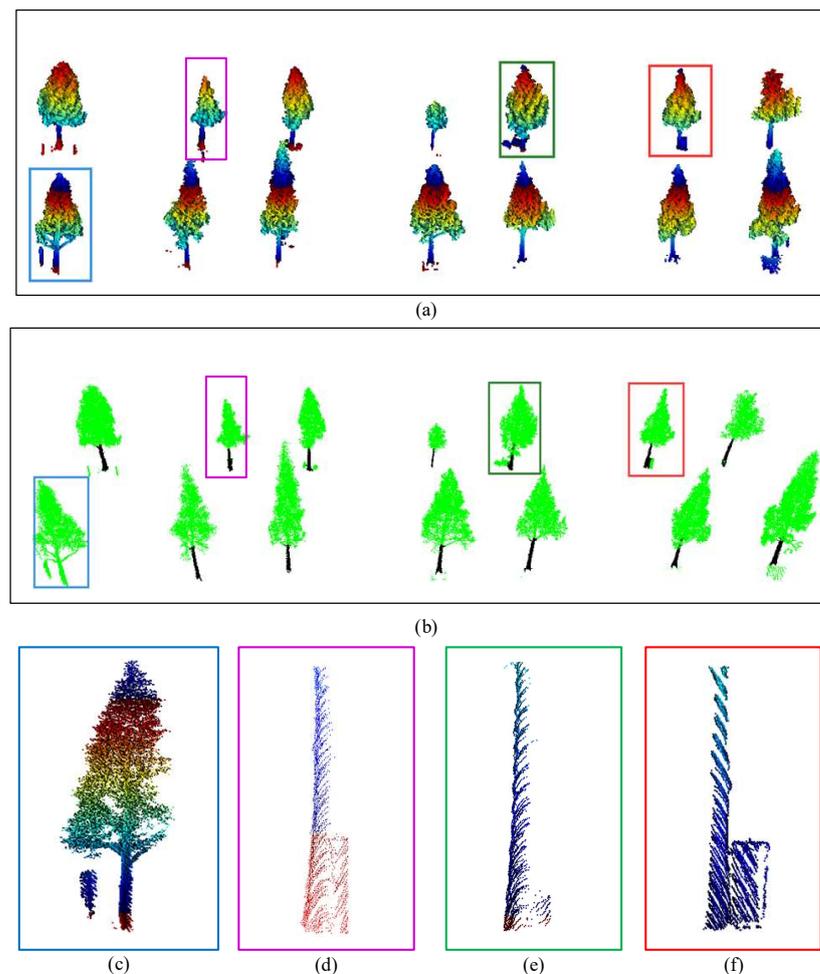


Figure 12. Processing result for dataset 1: (a) individual tree segmentation; (b) trunk extraction; (c) trunk extraction failed (trunk with much noise); (d–f) trunk-adhering noise.

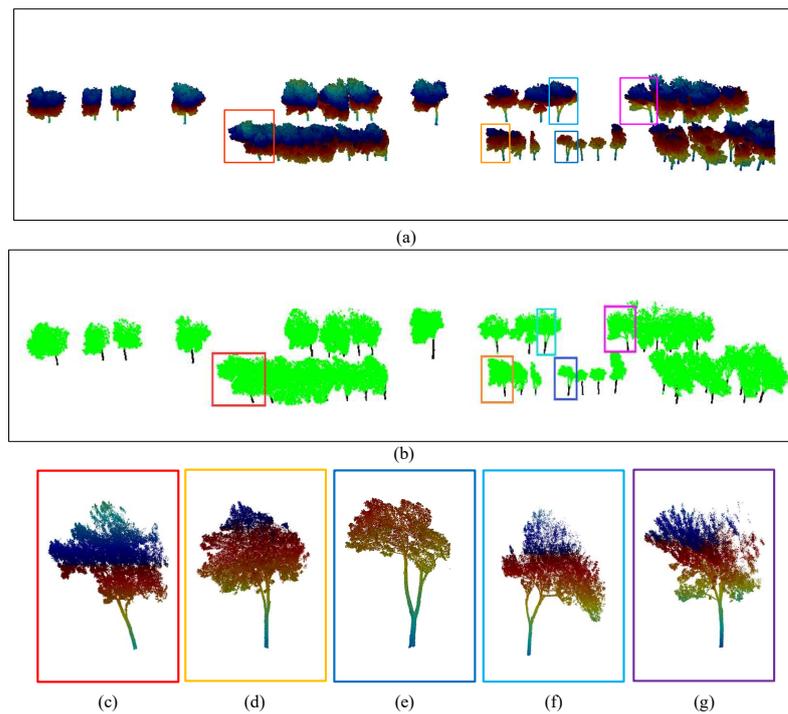


Figure 13. Processing result for dataset 2: (a) individual tree segmentation; (b) trunk extraction; (c) trunk extraction failed (a leaning tree trunk); (d–g) incomplete extraction of trunk.

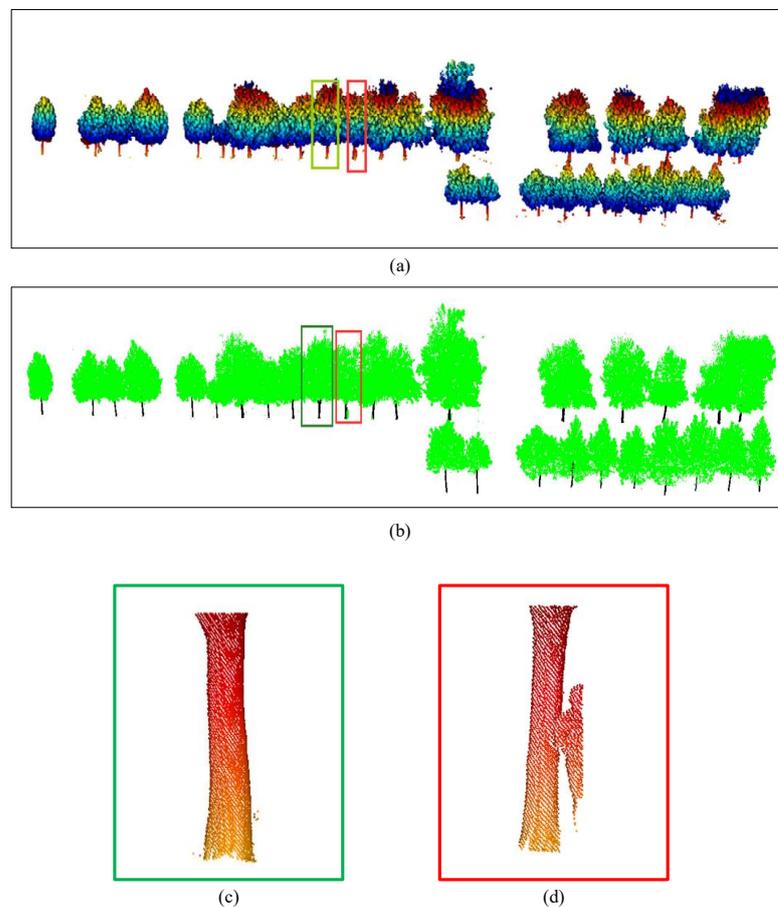


Figure 14. Processing result for dataset 3: (a) individual tree segmentation; (b) trunk extraction; (c) an incomplete trunk; (d) trunk-adhering noise.

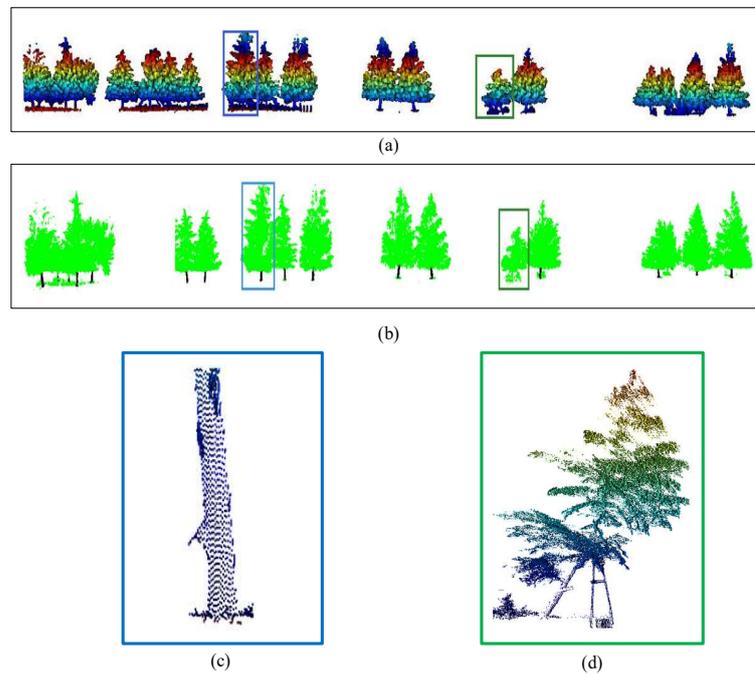


Figure 15. Processing result for dataset 4: (a) individual tree segmentation; (b) trunk extraction; (c) trunk with much noise; (d) trunk extraction failed.

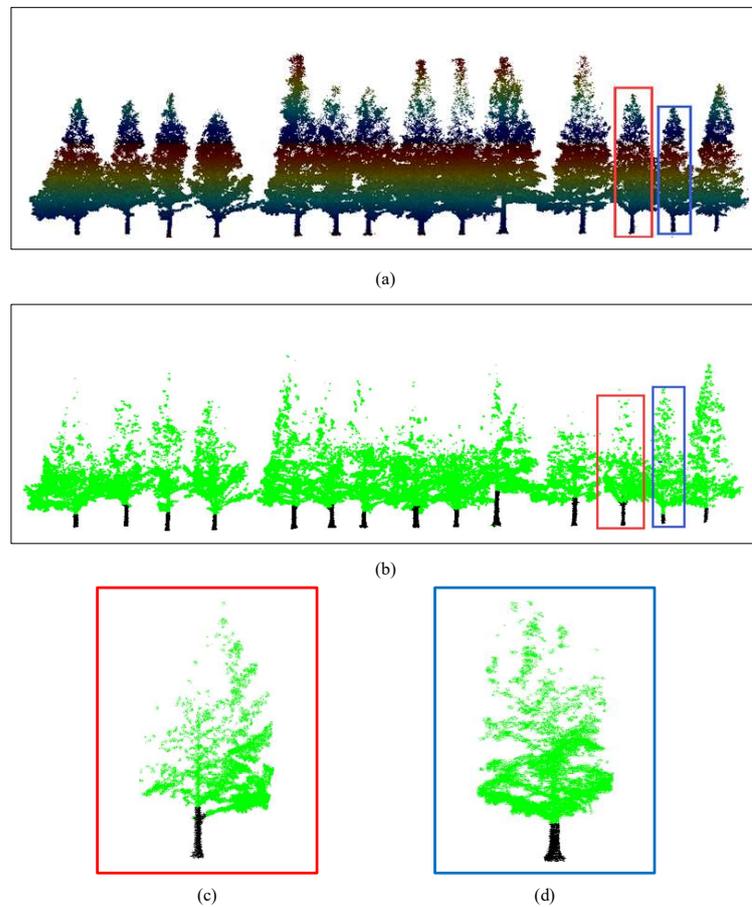


Figure 16. Processing result for dataset 5: (a) individual tree segmentation; (b) trunk extraction; (c) over-extraction of trunk; (d) incomplete extraction of trunk.

Table 4. Accuracy evaluation of trunk extraction experiment.

Dataset	TP	FP	FN	Precision (%)	Recall (%)	F-Score
1. Dataset 1	13	0	1	100	92.85	0.96
2. Dataset 2	43	1	4	97.73	91.49	0.94
3. Dataset 3	29	0	0	100	100	1
4. Dataset 4	18	1	0	94.73	100	0.97
5. Dataset 5	12	2	0	85.71	100	0.92
Total	115	4	5	96.64	95.83	0.96

Compared to dataset 1, the trees in dataset 2 and dataset 3 are dense and have overlapping canopies. This poses difficulties for the ITS of street trees. In dataset 2, an individual tree with a large degree of inclination appears (Figure 13c). Our optimal circle segmentation fails when the trunk is leaned to a large extent. In addition, the trunks of Figure 13d–g show significant branching, which can lead to the problem of inadequate trunk extraction. Thus, one unsegmented trunk and four trunks for which full extraction failed appeared in dataset 2. This also resulted in a lower *Recall* of 91.49% for this dataset; the *Precision* is 97.73%, and the *F-score* is 0.94. The separation between the crown and trunk of this species is more clearly defined in dataset 3. As shown in Figure 14b, the trunks of 29 trees were extracted normally and both the *Precision* and *Recall* are 100%; the *F-score* is 1.

Unlike the previous three datasets, dataset 4 and dataset 5 are coniferous forests. In dataset 4 as shown in Figure 15, trees are characterized by a shorter visible part of the trunk and more noise. Except for one leaning street tree (Figure 15d), the trunks of the remaining 18 trees could be extracted. The *Precision*, *Recall* and *F-score* of the dataset 4 are 94.73%, 100%, and 0.97, respectively. In dataset 5 as shown in Figure 16, incomplete and over-extraction of trunk extraction occurred once each (Figure 16c,d). This is very much related to the quality of the SLAM data. The noise affects the slice radius fit and incorrect trunk-crown separation occurs. The two incorrectly segmented trunks reduce the *Precision* and *F-score*. The *Precision*, *Recall* and *F-score* of the dataset 5 are 85.71%, 100%, and 0.92, respectively.

There were 124 street trees of different species in the five datasets, and 115 of the trees were correctly segmented, which basically realized the trunk-crown separation of all types of street tree. There were four trunks that failed to be extracted and five trunks that were extracted incorrectly. The *Precision* of trunk extraction was 96.64%, the *Recall* was 95.83%, and the *F-score* was 0.96. This way, the trunk and crown point clouds of individual trees can be effectively separated.

3.2.2. Refined Trunk Denoising

Four tree trunks with significant difficulty in denoising were selected to verify the proposed method. These four trunks are Figures 12d–f and 15c. Trunk outliers are denoised based on the improved iForest algorithm to delete most of the trunk noise, and the non-noise points of the trunk are rarely affected. It is possible to remove the tightly adhering noise while preserving the original form of the tree trunk to the maximum extent. Our trunk outlier denoising method based on improved iForest is more suitable for denoising pole-like structures, with better denoising effect and lower over-denoising potential, especially for outliers tightly adhering the target point cloud, which are difficult for the traditional algorithms to delete. The representative denoising effect is shown in Figure 17; the use of iForest adaptively solves contamination calculation and increases the automation degree of the algorithm.

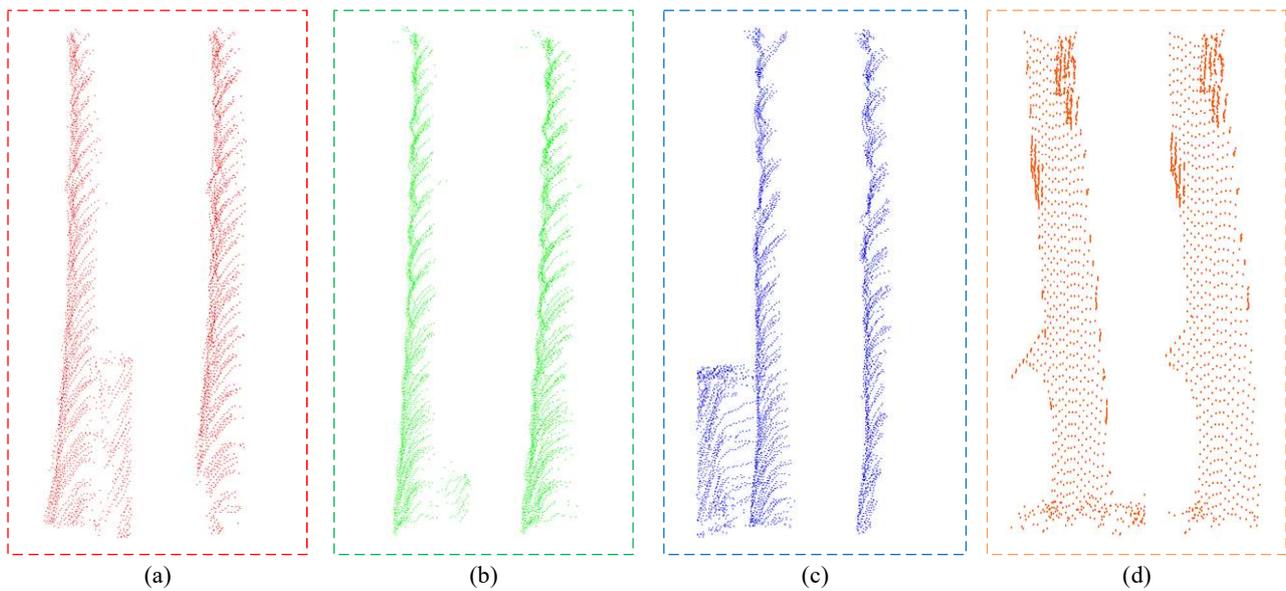


Figure 17. Denoising results: Comparison of four trunks denoising before and after. The left side of (a–d) is before denoising and the right side is after denoising.

4. Discussion

As the refinement of 3D modeling increases, higher demands are placed on data quality. Advances in ITS methods and optimizations have ensured the accuracy of extracting trunks at the individual tree level. However, in the process of MLS data acquisition, in addition to the noise caused by the scanning error and registration, there are often some features tightly attached to the trunk of the street trees, which become noise points [28,51] that are difficult to remove (as shown in Figure 17). They will have a large impact on trunk modeling. The objective of this study is to develop a complete workflow that properly separates the trunk canopy and removes the noise close to the trunk. For the noise close to the trunk that are difficult to resolve by traditional statistically based methods, the trunk and other noise are separated by an improved adaptive iForest to detect outliers.

4.1. Trunk-Crown Separation

The basic idea of separating the crown from the trunk is to find the trunk-crown separation point using point cloud slices. However, considering the influence of various types of miscellaneous points such as drooping branches and leaves as well as bushes, it will make the extracted trunk incomplete. Based on this, the main body of the trunk is extracted using a circle segmentation, which is a nesting-like idea. By comparing each slice with the initial slice to determine the reference slice, and then using the geometric information of the reference slice to perform a circular segmentation of the individual tree, the visible part of the trunk of the street tree can be extracted more completely by eliminating most of the interference of leaves and other shadows. This part is usually the part of the trunk below under branch height (UBH) and is also the key focus of the diameter at breast height calculation. The validation was carried out by using 124 street trees of different species and different growth states. Although the number of individual trees is small, they cover several common types of street trees and are obtained by different scanners, which also have some diversity. Thus, the feasibility and effectiveness of the algorithm can be verified. For MLS data, the proposed method also extracts a more complete trunk than the Cylindrical Fitting algorithm [32].

4.2. Trunk Denoising

For noise close to the target, it can be difficult to achieve the desired effect using denoising methods such as Radius Filter and SOR Filter. The proposed denoising method

based on the BIRCH clustering improved isolated forest algorithm is more suitable for rod denoising, with better denoising effect and less over denoising. Especially for the outliers that tightly close to the target point cloud, which are difficult to remove by traditional algorithms, the method in this paper can remove such noise. In the improved iForest algorithm, the pollution value calculation problem is solved adaptively, and the automation of the algorithm is improved. Experiments on the trunk with tightly adhering noise extracted in 2.5.3 show that the denoising effect and accuracy evaluation are better compared with iForest, Radius Filter and SOR Filter.

4.3. Analysis of Failure Cases

Failures are primarily associated with trunk extraction. In dataset 1, as shown in Figure 12c, as the noise is parallel to the trunk and almost equally long, its parallelism with the trunk during projection results in an error in reference slicing identification: the circle segmentation radius is too large and almost nests the entire tree. However, as shown in Figure 18a,b, after removing this large noise, we can also obtain the correct result. In dataset 2, we can see that the trunk of a tree (Figure 13c) is almost not extracted. This is due to the fact that the trunk of this tree is leaning to a large extent. The use of the optimal circle segmentation method leads to a wrong segmentation, as shown in Figure 18c. Our proposed method is more suitable for street trees also because street trees generally grow vertically and have a certain spacing distribution. However, there are also four trunks for which extraction failed completely, as shown in Figure 19. This is mainly due to the phenomenon that the main trunk of this tree species will branch, which causes our algorithm to be inadequate in trunk extraction, and there are trunks that fail to be completely segmented. In dataset 3, as shown in Figure 14c, as the trunk bifurcates here, a mutation takes place in the slicing radius earlier than normal. This results in incomplete trunk extraction, but rarely affects trunk DBH calculation and 3D reconstruction. In dataset 4, the inclination of the trunk greatly affects the initial slice, which is the reason why the segmentation failed in Figure 18d. This is a similar situation to the incorrect segmentation that occurred in dataset 2.

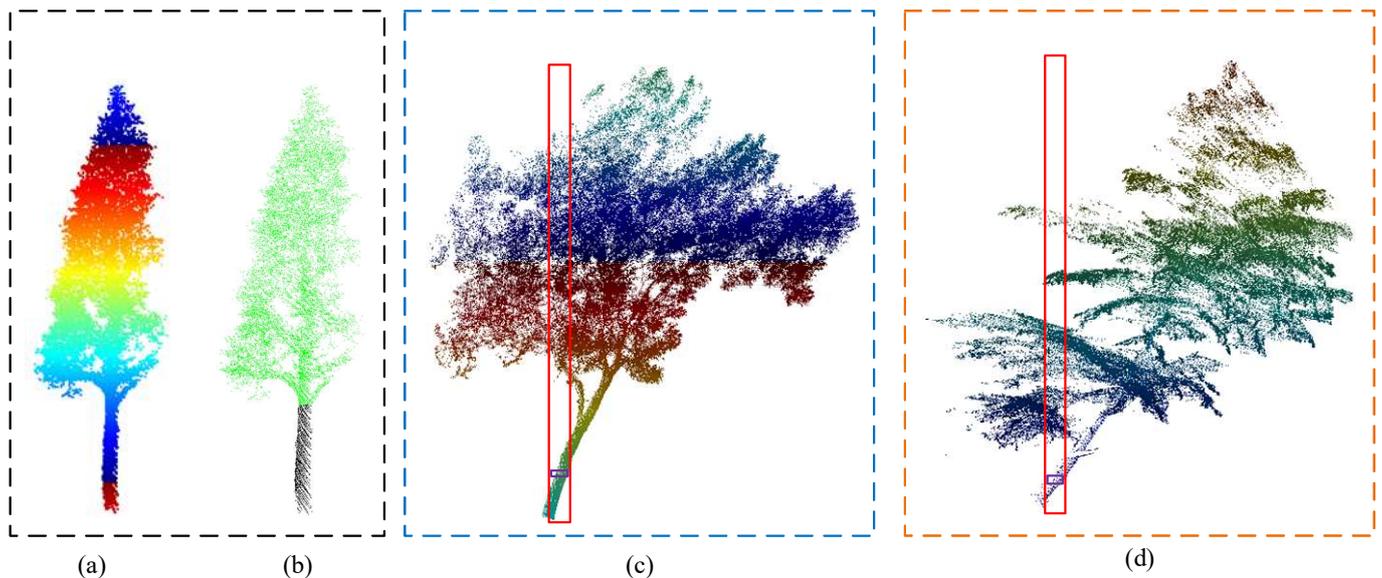


Figure 18. Mis-segmentation cases: (a) Manual denoising of Figure 12c; (b) The effect of trunk extraction; (c) Circle segmentation results of Figure 13c; (d) Circle segmentation results of Figure 15d. (The purple rectangles indicate the acquired reference slices and the red rectangles indicate the circle segmentation results in (c,d).

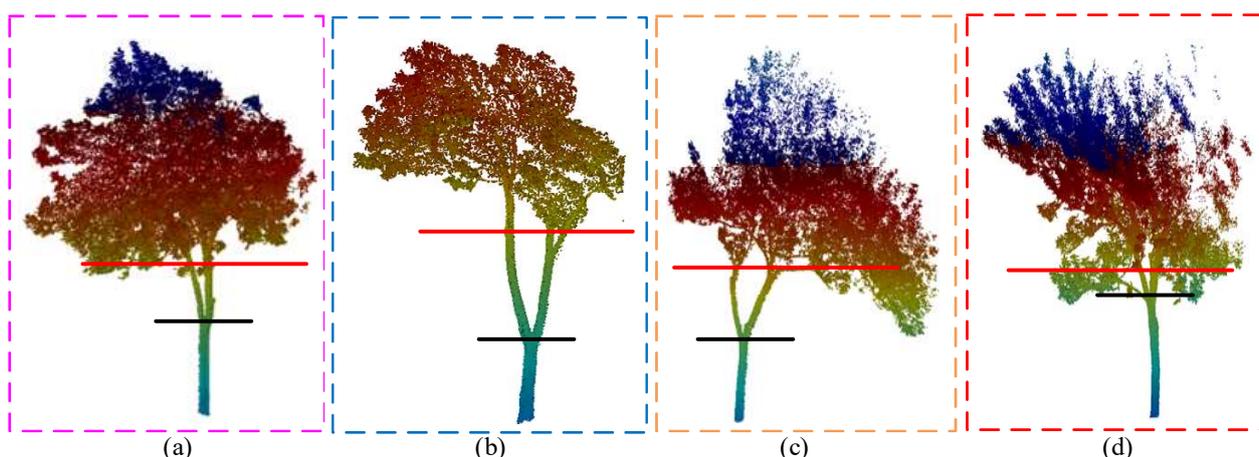


Figure 19. Incomplete segmentation cases: The red solid line indicates the correct segmentation position. The black solid line indicates the proposed method segmentation position. (a) Figure 13d; (b) Figure 13e; (c) Figure 13f; (d) Figure 13g.

In dataset 5, the separation of the trunk from the crown of this species is not obvious. the SLAM data contain a particularly large amount of noise as shown in Figure 20a, and although filtering and denoising has been performed before trunk extraction, there are still two trees that are affected by noise and appear to be incorrectly separated in Figure 16c,d. Furthermore, many small branches and leaves can appear on the trunk of the tree, affecting our algorithm as shown in Figure 20b. In particular, the problem of registration fault layers in SLAM data can cause noise when slicing. The noise affects the fitting radius of the slices, and it is problematic to use the change in fitting radius to judge the separation position of trunk and crown.

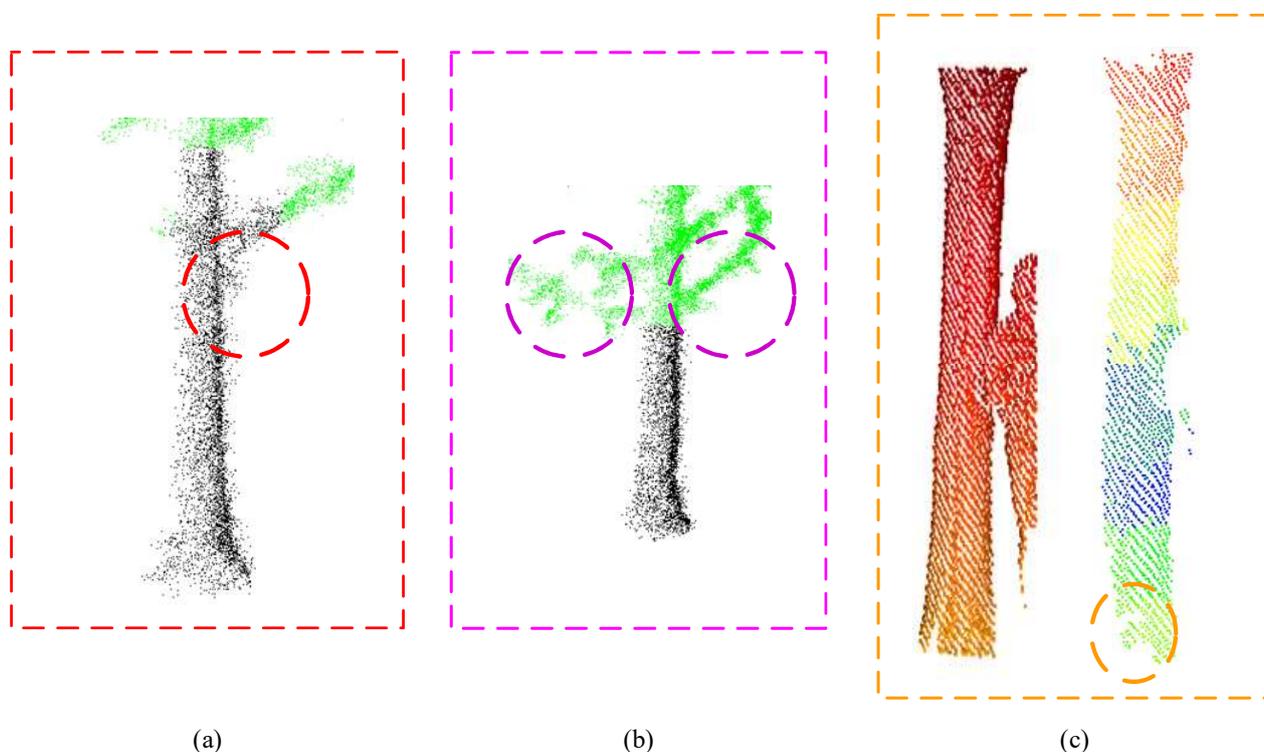


Figure 20. Other Failure Cases: (a) Figure 16c, the red circles indicate SLAM noise; (b) Figure 16d, the purple circles indicate the small branches on the trunk of the tree; (c) The denoising effect of Figure 14d; the orange circle indicates the excessive denoising part.

In general, there are three categories of causes of trunk and crown separation errors. i: Point cloud data quality problem. The main problem is the interference of trunk noise, including the support structure of the trunk, the scanning noise of the acquired point cloud, etc. ii: The problem of street tree species. Incomplete trunk extraction occurs for trunks that show obvious branches. iii: Tree posture problem. Although most of the street trees grow straight, there are problems of large inclination due to various reasons, which affects the nested extraction of the trunks. The first two types of causes can be effectively improved by manual denoising as well as by adjusting the radius parameter, but the third case is still a problem waiting to be solved.

The tree trunk in Figure 14d is excessively denoised due to the noise density close to the tree trunk as shown in Figure 20c. Some of the point clouds of the tree trunk were removed, probably because the pedestrian was leaning on this trunk and had about the same density as the trunk point cloud, resulting in a false identification, but most of the noise was also removed. However, such cases occur less frequently, and the small amount of excessive denoising basically does not affect the 3D reconstruction work of the trunk.

4.4. Limitations and Perspectives

In this paper, only trunk extraction from the highly visible part of street trees with a denser canopy was studied. Therefore, additional experiments are needed to improve the credibility of the method and to extend it to the extraction of trunks in cases such as canopy leaf drop, as well as to a variety of other street tree species.

From the methodological perspective, the accuracy of trunk–crown separation in a patchy area is limited by the accuracy of ITS. The more accurate individual tree extraction in the same area, the better trunk–crown separation can be achieved. Therefore, the accuracy of ITS was also artificially improved by pre-processing means during the experiment.

From a parametric point of view, multiple slices were used in the trunk–crown separation. The thickness of these slices was verified through several experiments. On the whole, the slice thickness is more accurate when it is small, but also due to the point cloud quality, the slice thickness is too small, which makes the slice incomplete and impossible to fit. When judging the trunk–crown separation position, the selection of the variation multiplier of slice radius is also achieved by adjusting the parameters several times.

Therefore, its adaptive parameter adjustment capability can be improved through deep learning or other means, and the pervasiveness of the algorithm can also be improved through automatic parameter adjustment in the future. In terms of principle and methodology, the proposed method is also applicable to tree trunk extraction in other scenes. In addition, the proposed denoising method is also applicable to the denoising of other pole-like objects. However, we did not find a suitable dataset to validate this, so we can add experiments to validate this in the future.

5. Conclusions

The method is proposed and validated to extract and denoise the trunk point cloud of street trees in MLS data. First, an individual street tree is sliced using horizontal slices, and the least disturbed slice among them is adaptively selected as the reference slice according to the RMSE value of each slice. The reference slices provide axis and radius information for optimal circle segmentation, which reveals the main body of trunk from the point cloud data. Then the individual tree after circular segmentation is sliced again. The radius of each slice is calculated, and the trunk crown separation position is judged by the radius change to extract the trunk point cloud. Finally, BIRCH and Mean Shift are used to calculate the centroid of the trunk point cloud and automatically obtain the contamination as a way to improve the iForest algorithm, which removes the noise points immediately adjacent to the trunk.

Through the experiments with five datasets of different tree species and point cloud quality, the results show that the proposed method has better results for both broad-leaved and coniferous street trees. The five datasets total 124 street trees, of which 115

trees were correctly segmented, achieved better results for both vehicle-borne and SIAM data. After the optimal circle segmentation process, the extraction of the trunk is more complete; the noise points immediately adjacent to the trunk also have a better denoising effect. The method improves the accuracy of trunk segmentation and optimizes both the completeness and fineness of the extracted trunks, providing more refined data for subsequent information acquisition, parameter calculation and 3D model construction of street trees. However, there are still trunks that are not fully recognized in the street tree trunk extraction, and a combination of deep learning is considered necessary to improve the robustness of the algorithm.

Author Contributions: Conceptualization, J.W.; methodology, Z.L. and Z.Z.; validation, Z.Z.; data curation, Y.C.; writing—original draft preparation, Z.L. and Z.Z.; writing—review and editing, J.W., F.J., J.Y. and W.S.; visualization, Z.L. and Y.C.; supervision, F.J. and W.S.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Introduction plan of high-end foreign experts (Grant No. G2021025006L).

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: We want to thank Ayman Habib for his assistance and the anonymous reviewers for their relevant and in-depth comments. And we also want to thank the Qingdao Key Laboratory of Beidou Navigation and Intelligent Spatial Information Technology Application for funding support.

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

References

1. Nguyen, V.-T.; Constant, T.; Colin, F. An innovative and automated method for characterizing wood defects on trunk surfaces using high-density 3D terrestrial LiDAR data. *Ann. For. Sci.* **2021**, *78*, 32. [\[CrossRef\]](#)
2. Chen, X.; Neubert, B.; Xu, Y.-Q.; Deussen, O.; Kang, S.B. Sketch-based tree modeling using Markov random field. *ACM Trans. Graph.* **2008**, *27*, 1–9.
3. Di Stefano, F.; Chiappini, S.; Gorreja, A.; Balestra, M.; Pierdicca, R. Mobile 3D scan LiDAR: A literature review. *Geomat. Nat. Hazards Risk* **2021**, *12*, 2387–2429. [\[CrossRef\]](#)
4. Liu, T.; Im, J.; Quackenbush, L.J. A novel transferable individual tree crown delineation model based on Fishing Net Dragging and boundary classification. *ISPRS J. Photogramm. Remote Sens.* **2015**, *110*, 34–47. [\[CrossRef\]](#)
5. Dai, W.; Yang, B.; Dong, Z.; Shaker, A. A new method for 3D individual tree extraction using multispectral airborne LiDAR point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *144*, 400–411. [\[CrossRef\]](#)
6. Fan, W.; Yang, B.; Dong, Z.; Liang, F.; Xiao, J.; Li, F. Confidence-guided roadside individual tree extraction for ecological benefit estimation. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *102*, 102368. [\[CrossRef\]](#)
7. Balenovi, I.; Liang, X.; Jurjevic, L.; Hyypä, J.; Kukko, A. Hand-Held Personal Laser Scanning: Current Status and Perspectives for Forest Inventory Application. *Croat. J. For. Eng.* **2021**, *42*, 165–183. [\[CrossRef\]](#)
8. Vandendaele, B.; Martin-Ducup, O.; Fournier, R.A.; Pelletier, G.; Lejeune, P. Mobile Laser Scanning for Estimating Tree Structural Attributes in a Temperate Hardwood Forest. *Remote Sens.* **2022**, *14*, 4522. [\[CrossRef\]](#)
9. Sun, W.; Wang, J.; Jin, F.; Yang, Y. A quality improvement method for 3D laser slam point clouds based on geometric primitives of the scan scene. *Int. J. Remote Sens.* **2020**, *42*, 378–388. [\[CrossRef\]](#)
10. Wang, Y.; Jiang, T.; Liu, J.; Li, X.; Liang, C. Hierarchical Instance Recognition of Individual Roadside Trees in Environmentally Complex Urban Areas from UAV Laser Scanning Point Clouds. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 595. [\[CrossRef\]](#)
11. Liu, H.; Wu, C. Tree Crown Width Estimation, Using Discrete Airborne LiDAR Data. *Can. J. Remote Sens.* **2016**, *42*, 610–618. [\[CrossRef\]](#)
12. Kwong, I.H.Y.; Fung, T. Tree height mapping and crown delineation using LiDAR, large format aerial photographs, and unmanned aerial vehicle photogrammetry in subtropical urban forest. *Int. J. Remote Sens.* **2020**, *41*, 5228–5256. [\[CrossRef\]](#)
13. Mokroš, M.; Mikita, T.; Singh, A.; Tomaščík, J.; Chudá, J.; Wężyk, P.; Liang, X. Novel low-cost mobile mapping systems for forest inventories as terrestrial laser scanning alternatives. *Int. J. Appl. Earth Obs. Geoinf.* **2021**, *104*, 102512. [\[CrossRef\]](#)
14. Ferraz, A.; Saatchi, S.; Mallet, C.; Meyer, V. Lidar detection of individual tree size in tropical forests. *Remote Sens. Environ.* **2016**, *183*, 318–333. [\[CrossRef\]](#)
15. Lin, W.; Meng, Y.; Qiu, Z.; Zhang, S.; Wu, J. Measurement and calculation of crown projection area and crown volume of individual trees based on 3D laser-scanned point-cloud data. *Int. J. Remote Sens.* **2017**, *38*, 1083–1100. [\[CrossRef\]](#)
16. Khosravipour, A.; Skidmore, A.K.; Isenburg, M. Generating spike-free digital surface models using LiDAR raw point clouds: A new approach for forestry applications. *Int. J. Appl. Earth Obs. Geoinf.* **2016**, *52*, 104–114. [\[CrossRef\]](#)

17. Stovall, A.E.L.; Anderson-Teixeira, K.J.; Shugart, H.H. Assessing terrestrial laser scanning for developing non-destructive biomass allometry. *For. Ecol. Manag.* **2018**, *427*, 217–229. [[CrossRef](#)]
18. Bauwens, S.; Bartholomeus, H.; Calders, K.; Lejeune, P. Forest Inventory with Terrestrial LiDAR: A Comparison of Static and Hand-Held Mobile Laser Scanning. *Forests* **2016**, *7*, 127. [[CrossRef](#)]
19. Rutzinger, M.; Pratihast, A.K.; Oude Elberink, S.J.; Vosselman, G. Tree modelling from mobile laser scanning data-sets. *Photogramm. Rec.* **2011**, *26*, 361–372. [[CrossRef](#)]
20. Luo, D.; Yan-Min, W. Rapid extracting pillars by slicing point clouds. *International Archives of Photogrammetry. Remote Sens. Spat. Inf. Sci.* **2008**, *37*, 215–218.
21. Lehtomäki, M.; Jaakkola, A.; Hyyppä, J.; Kukko, A.; Kaartinen, H. Detection of Vertical Pole-Like Objects in a Road Environment Using Vehicle-Based Laser Scanning Data. *Remote Sens.* **2010**, *2*, 641–664. [[CrossRef](#)]
22. Cabo, C.; Ordoñez, C.; García-Cortés, S.; Martínez, J. An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 47–56. [[CrossRef](#)]
23. Yu, Y.; Li, J.; Guan, H.; Wang, C.; Yu, J. Semiautomated Extraction of Street Light Poles from Mobile LiDAR Point-Clouds. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1374–1386. [[CrossRef](#)]
24. Huang, J.; You, S. Pole-like object detection and classification from urban point clouds. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 3032–3038.
25. Kang, Z.; Yang, J. A probabilistic graphical model for the classification of mobile LiDAR point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 108–123. [[CrossRef](#)]
26. Monnier, F.; Vallet, B.; Soheilian, B. Trees Detection from Laser Point Clouds Acquired in Dense Urban Areas by a Mobile Mapping System. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2012**, *3*, 245–250. [[CrossRef](#)]
27. Lamprecht, S.; Stoffels, J.; Dotzler, S.; Haß, E.; Udelhoven, T. aTrunk—An ALS-Based Trunk Detection Algorithm. *Remote Sens.* **2015**, *7*, 9975–9997. [[CrossRef](#)]
28. Li, L.; Li, D.; Zhu, H.; Li, Y. A dual growing method for the automatic extraction of individual trees from mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2016**, *120*, 37–52. [[CrossRef](#)]
29. Xu, S.; Xu, S.; Ye, N.; Zhu, F. Individual stem detection in residential environments with MLS data. *Remote Sens. Lett.* **2017**, *9*, 51–60. [[CrossRef](#)]
30. Hyyppä, E.; Kukko, A.; Kaijaluoto, R.; White, J.C.; Wulder, M.A.; Pyörälä, J.; Liang, X.; Yu, X.; Wang, Y.; Hyyppä, J.; et al. Accurate derivation of stem curve and volume using backpack mobile laser scanning. *ISPRS J. Photogramm. Remote Sens.* **2020**, *161*, 246–262. [[CrossRef](#)]
31. Ai, M.; Yao, Y.; Hu, Q.; Wang, Y.; Wang, W. An Automatic Tree Skeleton Extraction Approach Based on Multi-View Slicing Using Terrestrial LiDAR Scans Data. *Remote Sens.* **2020**, *12*, 3824. [[CrossRef](#)]
32. Xu, S.; Sun, X.; Yun, J.; Wang, H. A New Clustering-Based Framework to the Stem Estimation and Growth Fitting of Street Trees from Mobile Laser Scanning Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 3240–3250. [[CrossRef](#)]
33. Neuville, R.; Bates, J.S.; Jonard, F. Estimating Forest Structure from UAV-Mounted LiDAR Point Cloud Using Machine Learning. *Remote Sens.* **2021**, *13*, 352. [[CrossRef](#)]
34. Alvites, C.; Santopuoli, G.; Hollaus, M.; Pfeifer, N.; Maesano, M.; Moresi, F.V.; Marchetti, M.; Lasserre, B. Terrestrial Laser Scanning for Quantifying Timber Assortments from Standing Trees in a Mixed and Multi-Layered Mediterranean Forest. *Remote Sens.* **2021**, *13*, 4265. [[CrossRef](#)]
35. Li, M.; Sun, C. Refinement of LiDAR point clouds using a super voxel based approach. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 213–221. [[CrossRef](#)]
36. Balta, H.; Velagic, J.; Bosschaerts, W.; Cubber, G.; Siciliano, B. Fast Statistical Outlier Removal Based Method for Large 3D Point Clouds of Outdoor Environments. *IFAC-Pap. Line* **2018**, *22*, 348–353. [[CrossRef](#)]
37. Pirotti, F.; Ravanelli, R.; Fissore, F.; Masiero, A. Implementation and assessment of two density-based outlier detection methods over large spatial point clouds. *Open Geospat. Data Softw. Stand.* **2018**, *3*, 14. [[CrossRef](#)]
38. Hua, Z.; Xu, S.; Liu, Y. Individual Tree Segmentation from Side-View LiDAR Point Clouds of Street Trees Using Shadow-Cut. *Remote Sens.* **2022**, *14*, 5742. [[CrossRef](#)]
39. De Conto, T.; Olofsson, K.; Görgens, E.B.; Rodriguez, L.C.E.; Almeida, G. Performance of stem denoising and stem modeling algorithms on individual tree point clouds from terrestrial laser scanning. *Comput. Electron. Agric.* **2017**, *143*, 165–176. [[CrossRef](#)]
40. Raunonen, P.; Kaasalainen, M.; Åkerblom, M.; Kaasalainen, S.; Kaartinen, H.; Vastaranta, M.; Holopainen, M.; Disney, M.; Lewis, P. Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data. *Remote Sens.* **2013**, *5*, 491–520. [[CrossRef](#)]
41. Méndez, V.; Rosell-Polo, J.R.; Sanz, R.; Escolà, A.; Catalán, H. Deciduous tree reconstruction algorithm based on cylinder fitting from mobile terrestrial laser scanned point clouds. *Biosyst. Eng.* **2014**, *124*, 78–88. [[CrossRef](#)]
42. Liang, X.; Kankare, V.; Hyyppä, J.; Wang, Y.; Kukko, A.; Haggrén, H.; Vastaranta, M. Terrestrial laser scanning in forest inventories. *ISPRS J. Photogramm. Remote Sens.* **2016**, *115*, 63–77. [[CrossRef](#)]
43. Cao, W.; Wu, J.; Shi, Y.; Chen, D. Restoration of Individual Tree Missing Point Cloud Based on Local Features of Point Cloud. *Remote Sens.* **2022**, *14*, 1346. [[CrossRef](#)]
44. Roynard, X.; Deschaud, J.-E.; Goulette, F. Paris-Lille-3D: A large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification. *Int. J. Robot. Res.* **2018**, *37*, 545–557. [[CrossRef](#)]

45. Li, W.; Guo, Q.; Jakubowski, M.K.; Kelly, M. A New Method for Segmenting Individual Trees from the Lidar Point Cloud. *Photogramm. Eng. Remote Sens.* **2012**, *78*, 75–84. [[CrossRef](#)]
46. Chave, J.; Andalo, C.; Brown, S.; Cairns, M.A.; Chambers, J.Q.; Eamus, D.; Fölster, H.; Fromard, F.; Higuchi, N.; Yamakura, T. Tree allometry and improved estimation of carbon stocks and balance in tropical forests. *Oecologia* **2005**, *145*, 87–99. [[CrossRef](#)]
47. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *SIGMOD Rec.* **1996**, *25*, 103–114. [[CrossRef](#)]
48. Charest, L.; Plante, J.-F. Using balanced iterative reducing and clustering hierarchies to compute approximate rank statistics on massive datasets. *J. Stat. Comput. Simul.* **2013**, *84*, 2214–2232. [[CrossRef](#)]
49. Liu, F.T.; Ting, K.M.; Zhou, Z.-H. Isolation forest. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 413–422.
50. Karczmarek, P.; Kiersztyn, A.; Pedrycz, W.; Al, E. K-Means-based isolation forest. *Knowl.-Based Syst.* **2020**, *195*, 105659. [[CrossRef](#)]
51. Zhang, Z.; Chen, G.; Wang, X.; Shu, M. Fore-Net: Efficient inlier estimation network for large-scale indoor scenario. *ISPRS J. Photogramm. Remote Sens.* **2022**, *184*, 165–176. [[CrossRef](#)]

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.