

## Article

# Soft Segmentation and Reconstruction of Tree Crown from Laser Scanning Data

Mingrui Dai \* and Guohua Li

Institute of Computing Technology, China Academy of Railway Sciences Corporation Limited,  
Beijing 100081, China

\* Correspondence: [daimr@rails.cn](mailto:daimr@rails.cn)

**Abstract:** Point cloud data obtained by laser scanning can be used for object shape modeling and analysis, including forest inventory. One of the inventory tasks is individual tree extraction and measurement. However, individual tree segmentation, especially tree crown segmentation, is challenging. In this paper, we present a novel soft segmentation algorithm to segment tree crowns in point clouds automatically and reconstruct the tree crown surface from the segmented crown point cloud. The soft segmentation algorithm mainly processes the overlapping region of the tree crown. The experimental results showed that the segmented crown was accurate, and the reconstructed crown looked natural. The reconstruction algorithm was highly efficient in calculating the time and memory cost aspects since the number of the extracted boundary points was small. With the reconstructed crown geometry, the crown attributes, including the width, height, superficial area, projecting ground area, and volume, could be estimated. The algorithm presented here is effective for tree crown segmentation.

**Keywords:** tree crown segmentation; soft segmentation; tree crown reconstruction; tree crown shape; point cloud; laser scanning data



**Citation:** Dai, M.; Li, G. Soft Segmentation and Reconstruction of Tree Crown from Laser Scanning Data. *Electronics* **2023**, *12*, 2300. <https://doi.org/10.3390/electronics12102300>

Academic Editor: Bahman Javadi

Received: 28 April 2023

Revised: 14 May 2023

Accepted: 17 May 2023

Published: 19 May 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

The forest is closely related to human life and is an essential element of the metaverse and other virtual spaces, including three-dimensional (3D) computer games and movies. Whether studying forestry or modeling geometrical tree shapes, it is necessary to obtain forest data. A convenient and efficient way to obtain forest data is to use remote sensing technology, such as terrestrial fixed-point or airborne LiDAR, to scan a forest. These kind of data consist of a large number of discrete points in 3D space, called point clouds. In point cloud data, each point is recorded as 3D coordinates  $(x, y, z)$ . Compared to 2D image data, it has accurate three-dimensional position information without scaling or deformation. As pointed out in previous studies, the successful detection and delineation of individual trees from forest point cloud is critical, allowing for studies of individual tree demography, growth modeling, and more precise measures of biomass [1]. However, the segmentation of an individual tree, especially tree crown segmentation, is not easy. Apart from the coordinates of the points, we should estimate other information, such as the connection relationship and neighborhood relationship of these points.

As for the individual trees and the crown segmentation from LiDAR data, many methods have been presented in the past ten years. These existing methods can be roughly divided into three categories.

The first category method is based on treetop detection. When the tree spacing at the upper lever is large, exploiting the spacing between the tops of trees is effective to identify and group points into a single individual tree [2]. After a treetop has been detected, the tree climbing method can be applied to identify the individual tree, and then a donut expanding and sliding method tree can be used to detect the crown boundary and isolate

the individual trees [3]. Another way is that after the treetop has been detected, it is consecutively connected and accumulated by vertically traversing the point layers, which results in the individual tree delineation [4]. The canopy height model can be used for coarse segmentation, and the segmented results can be refined using a multi-direction 3D profile analysis and K-mean clustering [5]. In addition, tree segmentation can be practiced by combining the local maxima (treetop) with the growing region algorithm or Voronoi tessellation method. However, in some cases, these combination methods do not provide better results than the tree relative distance algorithm [6].

The second category is based on horizontally cut clusters. A representative method is layer stacking, which slices the entire forest point cloud at 1-m height intervals and isolates the trees in each layer. Then, merging the results from all the layers produces the representative tree profiles [7]. A simpler method than layer stacking is the projection algorithm, which projects the point cloud onto the  $xOy$  plane and employs the hybrid clustering technique, including a combination of DBSCAN and K-means, to segment the individual trees from the forest point cloud [8]. It has been observed that the point density decreases with an increasing distance from the trajectory. A distance-dependent algorithm that considers the inhomogeneities in point density was developed for the segmentation of the forest point clouds [9]. To overcome the limitation of the method that takes the highest point in a filtering window as the tree position, H. Liu et al. used the cluster center of the higher points to detect the tree position [10].

The third category method is based on the fusion of multiple algorithms. Given the complexity of forest point cloud segmentation, a processing chain consisting of the stand delineation, canopy height model, characterization, and point clustering with an adaptive mean shift was proposed [11]. For terrestrial backpack LiDAR data, the individual trees were extracted based on DBSCAN clustering and a cylinder voxelization of the volume, which showed a high detection rate for the tree locations [12]. To improve the results of the individual tree detection algorithms, M. Lisiewicz et al. proposed a three-step approach to correct the segmentation errors [13]. Recently, the structure and geometry shape were considered for improving the segmentation effect. For example, for individual tree crown segmentation from laser data that focuses on overstory and understory trees, a framework that combines the detection of the symmetrical structure of the trees and mean shift clustering was proposed [14]. With branch–trunk constraints, a hierarchical clustering method was proposed to extract street trees from mobile laser scanning (MLS) point clouds [15]. Using the shape of the scanline and circle fitting, an individual tree can be segmented, and the stem attributes can be estimated [16]. In addition, the application of a convolutional neural network to perform the tree crown detection and delineation has also been developed [17].

The reconstruction of a tree and a tree crown is a direct follow-up to point cloud segmentation, which measures tree properties more accurately and conveniently. The reconstructed geometrical models also show reality and can be used in virtual scenes in the virtual community, 3D computer games, movies, etc. In the past decades, much literature has proposed methods for reconstructing trees and tree crowns. Here, we only investigate some of the typical existing methods published in recent years.

Most of the reconstruction algorithms consist of the following four steps: (i) the segmentation of a TLS tree point cloud separating the wooden parts from foliage, (ii) the reconstruction of the trunk and branches, (iii) the distribution of the foliage within the tree crown using the points of the foliage cloud as the attractors, and (iv) the generation of a 3D representation [18]. The points from the leaves and branches can be segmented based on the convergence of the local principal curvature directions and the region growing method [19]. The tree trunk and branch geometries and topological structures are constructed based on skeleton extraction and then fitted with cylinders [20,21]. Therefore, the structural analysis and optimization of the extracted skeleton are often emphasized [22].

Tree crown segmentation and reconstruction is one of the key points in tree segmentation, which has received more and more attention. Although terrestrial laser scan point

clouds can provide precise feature observation of vegetation architecture and improve agricultural monitoring and management [23], the incompleteness that stems from the branch or leaf occlusion impairs the detection accuracy of the tree attributes. Reconstructing the tree crown geometry from the point clouds can compensate for this incompleteness for a better tree attribute estimation.

For the tree crown reconstruction, the crown points should be detected first from the point cloud. The distribution of the foliage within the tree crown uses the points of the foliage cloud as the attractors [18]. Paris et al. [24] presented a data fusion approach to extract the crown structures by exploiting the complementary perspective of airborne laser scanning. Cici et al. [25] delineated the tree crowns from the airborne laser scanning data using a Delaunay triangulation. The early crown reconstruction methods mainly employ surface fitting with a cylinder or paraboloid fitting surfaces [26]. Kato et al. [27] used radial basis functions (RBFs) to reconstruct the implicit surfaces approximating the individual tree crown shapes, and the tree crown formation was captured through the implicit surface reconstruction [28]. Lin et al. [29] provided a new method by combining the mobile mapping mode and a multi-echo-recording laser scanner to enhance the integrity of the individual tree crown reconstruction.

The other typical methods for reconstructing the tree crown geometry include the  $\alpha$ -shape method [30], the region-based level set method [31], and a voxel-based method to add leaves to a reconstructed 3D branch structure [32].

The direct surface fitting method for the crown surface reconstruction is short on details, and current other methods have a relatively high time cost for the computation. However, in some cases, the time efficiency of the reconstruction algorithm must be considered. For example, in vehicle [33] and airborne [34] laser scanning, the fast reconstruction of objects is very important for real-time monitoring and analyzing the scene [9,35]. The existing experiment results showed that crown segmentation in a multi-layered closed canopy forest could be improved using 3D segmentation methods rather than primarily relying on the canopy surface model [36].

The above methods were effective for tree counting and single tree separation. However, most of the methods did not pay attention to the crown shape, especially the overlapping of adjacent tree crowns, causing the shape and size of the crown to be overlooked and the shape reconstruction effect to lack realism. The former is essential for forest inventory, and the latter is important for digital model construction.

In this paper, we propose a new tree crown segmentation and reconstruction method based on terrestrial laser scan point clouds to obtain accurate tree crown segmented point clouds and their realistic geometrical reconstruction. Furthermore, our method can enhance the visual effects of reconstructed crowns with improved surface shape accuracy and enhance the reconstruction process.

The main highlights of this work include the following.

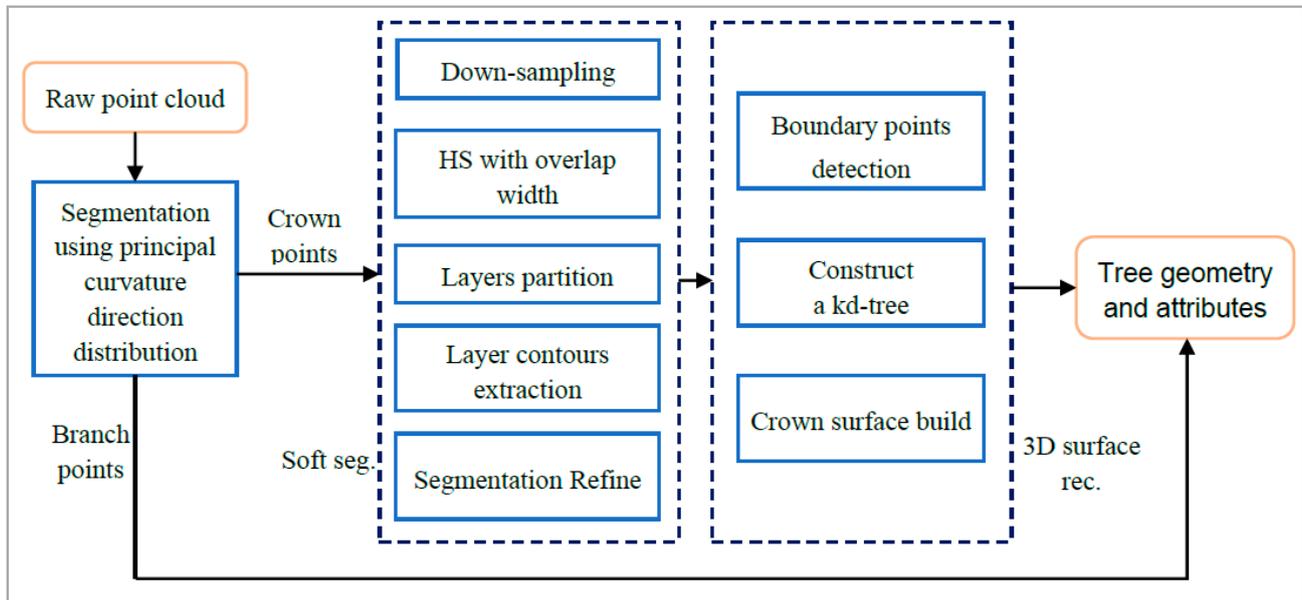
- (1) Construct an algorithm for segmenting and reconstructing tree crowns from laser scanning data, which can be applied to the forest inventory.
- (2) Propose a soft segmentation algorithm for making the reconstructed tree crown more natural and accurate.
- (3) Propose a fast reconstruction algorithm that fuses down-sampling and constructs a kd-tree.

The rest of this article is organized as follows. Section 2 describes the data we used and the proposed method in detail. Section 3 discusses the experimental results and the analysis for evaluating the proposed method. In the last section, Section 4, we conclude and list several limitations.

## 2. Data and Methods

Figure 1 illustrates the overview of our method's framework. The input data consisted of a raw point cloud. Only 3D coordinates of all the points were used. After segmenting the crown points from the branch points using the principal curvature direction distribution

method, the proposed soft segmentation algorithm was used to obtain the individual tree points. Then, the 3D silhouette surface was reconstructed. Finally, the tree geometry and the other attributes were estimated using the reconstructed trees.



**Figure 1.** The framework of the proposed method.

### 2.1. Data

The point cloud of the two pine trees used in our experiment was captured using a Cyrax or RIGEL scanner on the Peking University campus. The number of points in each scan was approximately 269,366 and 487,555, respectively. Each point was represented by 3D coordinates  $(x, y, z)$ . The length (unit: meter) in directions  $x$ ,  $y$ , and  $z$  were 8.47/11.19/11.12 and 10.19/9.31/11.94 in the two scans, respectively. The information is listed in detail under the tree images, as shown in Figure 2.

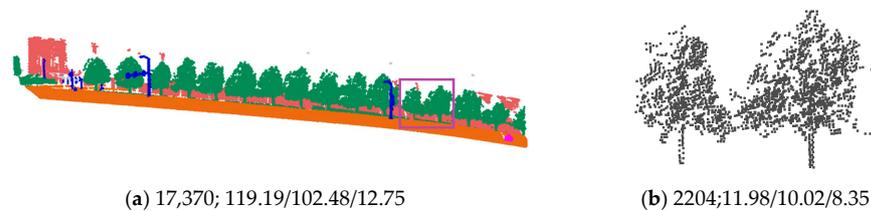


**Figure 2.** Point cloud data were all scanned on the Peking University campus. The length unit was listed in meters in these experimental data.

For testing the proposed soft segmentation algorithm, we combined two point clouds (Figure 2a,b) to form a large point cloud (Figure 2c). The experimental purpose of combining two trees at different intervals for segmentation was twofold. Firstly, it was done to obtain the simulated data of the overlapping tree crowns. On the other hand, it was used to study whether the accuracy of the proposed method was affected by the two trees with different degrees of overlap.

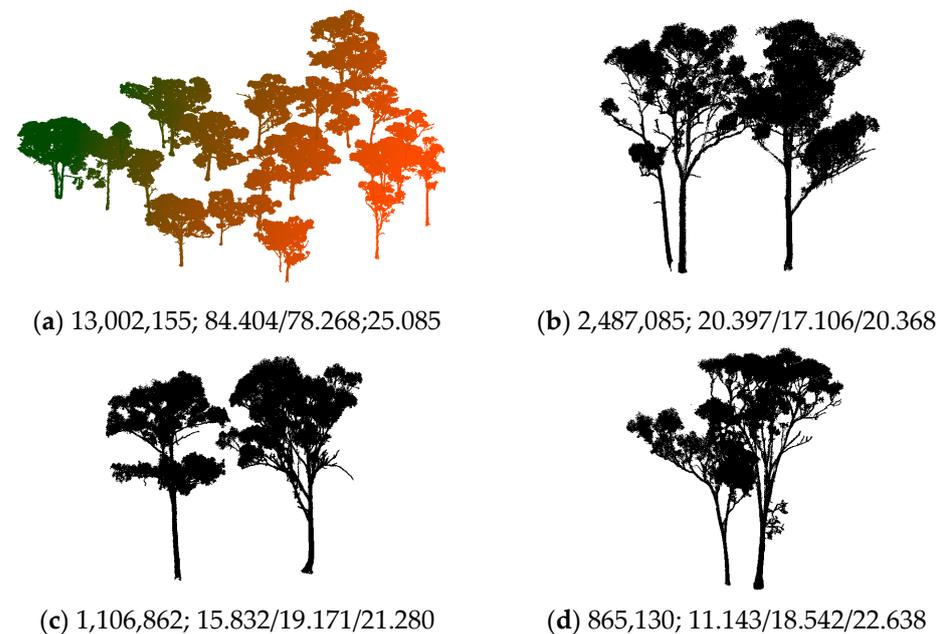
The second set of experimental data was street trees extracted from the Oakland point cloud data (Figure 3a). The Oakland point cloud data were collected around the Carnegie Mellon University (CMU) campus in Oakland, using Navlab11 equipped with side looking

SICK LMS laser scanners, and were opened by D. Munoz et al. [37]. The distance between two adjacent trees on the roadside was manually planned, and the distance between them was relatively large. Most of the adjacent trees had no intersecting areas between their crowns, resulting in accurate individual trees isolated using most of the existing methods, such as region growth or clustering. Therefore, we only chose two trees with overlapping crowns, as shown in Figure 3b, and named OaklandTrees to test our proposed soft segmentation algorithm.



**Figure 3.** Point cloud data from Oakland [37]. Subfigure (b) shows two trees, named OaklandTrees, labeled with a purple square in Subfigure (a).

Another set of experimental data was three pair of trees, which were extracted from RUSH07, as shown in Figure 4a. The point cloud data were acquired in the native Eucalypt open forest (dry sclerophyll box-ironbark forest) [38] in Victoria, Australia. Please refer to the introduction of this database for the geographic information and the data composition of the scanned plot.



**Figure 4.** Point cloud data, RUSH07 [38], and the three pair of trees were extracted from RUSH07. (a) RUSH07; (b) RUSH07treesA; (c) RUSH07treesB; and (d) RUSH07treesC.

Three pair of trees were selected as the experimental data because our main interest was in the segmentation of trees with overlapping crowns. The data, RUSH07, was segmented into several groups, and there were no overlapping areas between the tree crowns of the groups. Although there were more than three pair of groups with overlapping tree crowns, the three selected pairs, named *RUSH07treesA*, *RUSH07treesB*, and *RUSH07treesC*, as shown in Figure 4b–d, respectively, were representative. The difficulty of segmenting the three pair of trees was different. *RUSH07treesA* consisted of two trees with the same height. *RUSH07treesB* displayed an interlocking phenomenon between adjacent

tree crowns. *RUSH07treesC* was composed of two trees with different heights, which were relatively close together. The information about those three pair of trees can be found in Figure 4b–d.

## 2.2. Soft Segmentation

The proposed soft segmentation (SoftSeg) algorithm mainly deals with the crown points from two adjacent trees. SoftSeg consists of four steps: the crown points extraction; the crown layers partition; the vertical partition; and the layer contour extraction and refinement.

### 2.2.1. Crown Points Extraction

This step segments the crown points  $\Omega_C$  from the trunk and branch points  $\Omega_B$  in the point cloud  $\Omega$ , which consists of two trees, as shown in Figure 2c, for example.

$$\Omega = \Omega_C \cup \Omega_B \quad (1)$$

The segmentation algorithm for the branch and leaf separation can be any one of the existing approaches. We employed the principal curvature direction-based method proposed in the literature [19]. Based on the key assumption that the neighbor points from the branches usually have similar principal directions and the neighbor points from the leaves do not have this feature, the curvature direction-based algorithm consists of several steps. However, the first three steps can be used to the separate trunk and branch points from the leaf points. The first step is to estimate the principal directions and the principal curvatures of each point. The second step is to build the axis distribution of each point. The last step is to discriminate the point from a branch or a leaf by employing a threshold. Figure 5 shows the segmented crown points from Figure 2c.



**Figure 5.** The extracted crown shown in two different views.

As shown in Figure 5b, the top view of the segmented crown highlights the density of the point cloud in different regions of the extracted crown. The non-uniformity of the point cloud was potentially caused by unilateral laser scanning and self-occlusion, resulting in an unstable distance threshold value when the 3D surface was reconstructed.

We used the down-sampling method to overcome the influence of the non-uniformity of the point cloud density, similar to the voxel-based method [32]. Then, the sparse and uniform crown part points  $\Omega_C^{(s)}$  were obtained from  $\Omega_C$ . Figure 6a (6176 points) shows the s down-sample visual effect from Figure 5a (708,504 points).

In this way, the number of sampling points was dramatically reduced. Only 0.872% of the points were used for the crown segmentation and reconstruction. The shape of the tree crown point cloud after down-sampling was, in general, consistent with its original figure.

### 2.2.2. Vertical Partition

According to the segmentation results of the branch and leaf point cloud (Section 2.2.1), the tree roots ( $C_1$  and  $C_2$ ) and the trunk top positions were detected. Therefore, we used the minimum cut plane to construct the coarse segmentation of the two tree point clouds. The

approach consisted of three steps: partitioning the points into slices, finding the optimal dividing line, and obtaining the initial segmentation of the point cloud.



Figure 6. The down-sampling result  $\Omega_C^{(s)}$  of the extracted crown  $\Omega_C$ .

The first step was to partition points into slices. Assuming that the thickness  $\Delta_{vb}$  of each slice was 0.2 m, then the number of slices was the quotient obtained by dividing the distance between  $C_1$  and  $C_2$  by  $\Delta_{vb}$ , as shown in Formula (2).

$$N_{vb} = \frac{|C_1C_2|}{\Delta_{vb}} \tag{2}$$

Figure 7 illustrates the method for generating slices along the line  $C_1$  to  $C_2$ . Note that the partition was performed in 3D space, and the local coordinate system  $[C_1; \vec{C_1C_2}^o, \vec{C_1Y}, \vec{C_1Z}]$  was not the same as that of the laser scanner, where only the direction of the z-axis was the same.  $\vec{C_1C_2}^o$  is a unit vector parallel to  $\vec{C_1C_2}$ . Let  $C_1(x_c, y_c, z_c)$ , then any point  $p_i(x_i, y_i, z_i)$  belongs to the  $i^{th}$  slice  $B_i$ , if  $i = \left\lfloor \frac{\sqrt{(x_i-x_c)^2 + (y_i-y_c)^2}}{\Delta_{vb}} \right\rfloor$ . In this way, all the points in  $\Omega_C^{(s)}$  were partitioned into  $m$  slices.

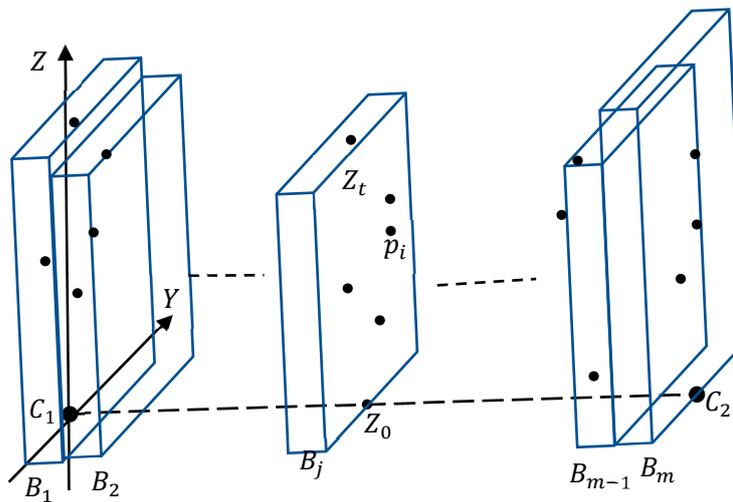


Figure 7. Generated slices using vertical partitioning.

The second step was to find the optimal dividing line, which was defined by the minimum cut plane and could be obtained using the solution of the minimum–maximum optimal problem, as shown in Formula (3).

$$i^* = \operatorname{argmin}_i \operatorname{max}_j \{z_j | \forall p_j(z_j, z_j, z_j) \in B_i\} \tag{3}$$

The last step was to segment the points into three categories: the unlabeled set, the point set from the first tree, and the point set from the second tree, labeled as 0, 1, and 2, respectively.

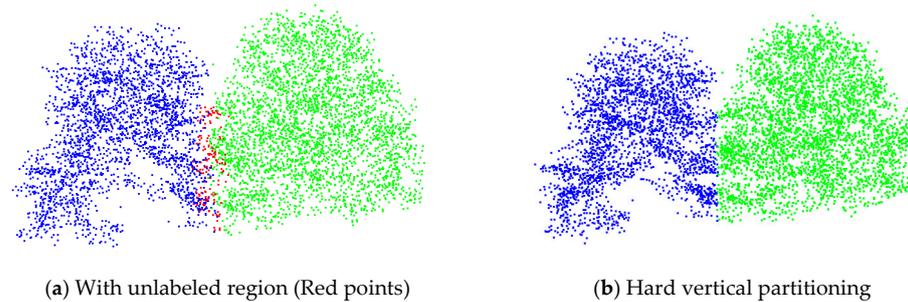
The label  $L(p_i)$  of a point  $p_i(x_i, y_i, z_i)$  was determined by a partitioning rule. If  $p_i \in B_j$ , then

$$L(p_i) = \begin{cases} 0, & \text{otherwise} \\ 1, & j < i^* - \varepsilon \\ 2, & j > i^* + \varepsilon \end{cases} \quad (4)$$

where  $\varepsilon \in N$ , known as the *overlap width*, is a parameter specified by the user and related to the overlap level of two crowns. The vertical partition result is shown in Figure 6a, where  $\varepsilon = 1$ . If the unlabeled case is deleted, Formula (4) is transformed into the following.

$$l(p_i) = \begin{cases} 1, & j \leq i^* \\ 2, & j > i^* \end{cases} \quad (5)$$

Formula (5) defines a kind of hard vertical partitioning, known as the *hard segmentation method* (HS), which segments the point cloud (Figure 6) into Figure 8b. From the subfigure, it can be found that after segmenting, the shape between the two segmented crowns was a straight line, which is generally not a natural shape since the adjacent tree crowns often overlap.



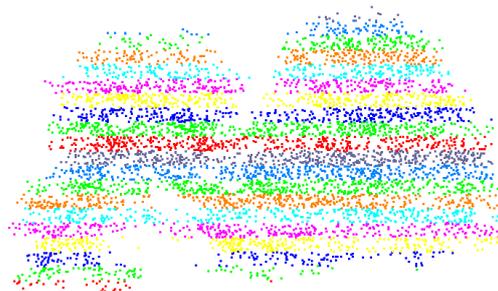
**Figure 8.** The vertical partitioning of the point cloud.

To improve the visual effect of the crown shape, Formula (4) was employed in our soft segmentation algorithm. The point in the intersection region was classified into the unlabeled point set, which was processed in the following subsections.

### 2.2.3. Crown Layers Partition

Unlike the methods that project 3D points on the ground or use an  $xOy$  plane to achieve the biggest crown contour, we used a method similar to the layer stacking method [7]. In this way, the shape of each layer of the crown contour was tighter.

For the crown point set  $\Omega_C^{(s)}$ , the layer bins  $\delta_i (i = 1, 2, \dots, k)$  were generated from horizontal slices. Figure 9 shows the partitioning effect.



**Figure 9.** The horizontal partitioning of the point cloud.

Let  $k$  be the number of bins.

$$\Omega_C^{(s)} = \delta_1 \cup \delta_2 \cup \dots \cup \delta_k \tag{6}$$

The cutting was uniform in the upward direction, and the thickness  $\Delta_b$  of each bin was calculated using Formula (7).

$$\Delta_b = \frac{1}{k}(Z_{\max} - Z_{\min}) \tag{7}$$

where  $Z_{\max} = \max\{z | \forall P(x, y, z) \in \Omega_C^{(s)}\}$  and  $Z_{\min} = \min\{z | \forall P(x, y, z) \in \Omega_C^{(s)}\}$ . The number of bins  $k$  is a parameter whose value depends on the laser scanning resolution. If the value of  $k$  is appropriate, each bin  $\delta_i$  is neither too thick nor too thin. In our experiment, the thickness  $\Delta_b$  of each bin was set to approx. 0.5 m, and the parameter  $k = 20$ , specified according to the experiment data that the height of the trees used in our experiments were approx. 10 m. In this way, all the points were partitioned into these bins.

#### 2.2.4. Layer Contour Extraction and Refinement

The unlabeled points in the intersection region were processed, as shown in Figure 10. This problem was similar to recovering the missing part of the crown. By referring to an idea that repairs the missing points based on their shape and structure [39], we also fully used the point cloud contour and optimized the shape of each layer bin.

The contour of each layer bin was extracted by projecting the points in the bin onto the horizontal cross-section in the middle of the bin and employing the two-dimensional (2D) contour extraction algorithm. The 2D contour was obtained sequentially, connecting the farthest projection point in each direction. For example, as shown in Figure 10, the farthest projection point in angle  $\angle A_1OA_2$  was  $A_1$ . The number of directions ( $m = 9, 18$  or  $36$ ) was specified by the experimental test.

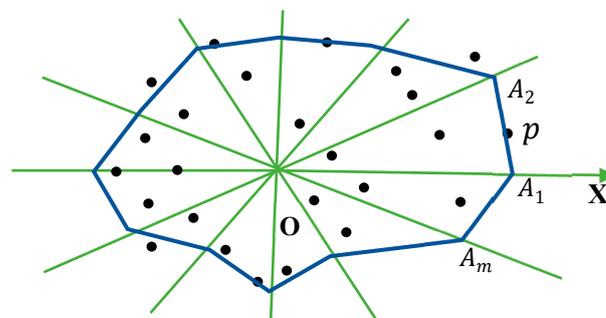


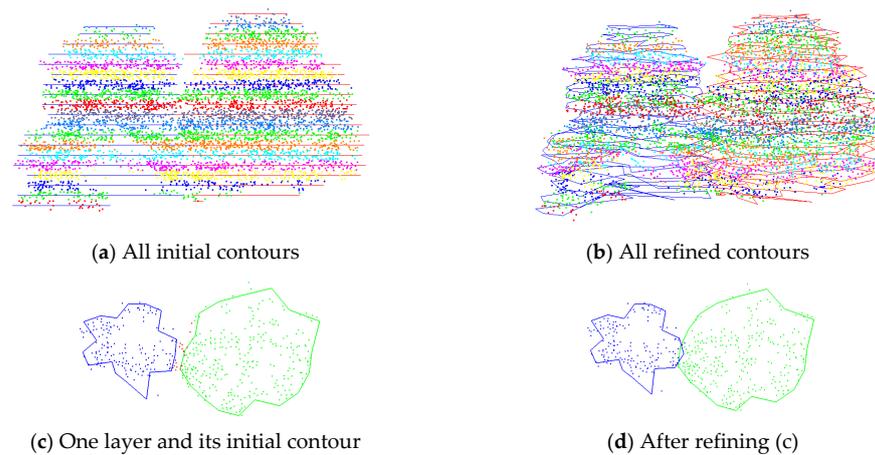
Figure 10. The 2D contour extraction of the points.

Using the 2D contour extraction approach, we obtained the contour of each layer tree by tree,  $L(p_i) = 1$  or  $2$ , as shown in Figure 11a. Figure 11c provides an example of a layer where the contours of the two trees did not overlap, and the unlabeled (red) points were not determined as to which tree they belonged to.

For an unlabeled point  $p$ ,  $L(p) = 0$ , we inferred its new label, according to the contour shape, distance, and randomized assignment. We first calculated  $d_1$  and  $d_2$ , which were the distances from  $p$  to the contours of Tree 1 and Tree 2, respectively. Then we used the Monte Carlo method to label  $p$  as one or two, according to the probability.

$$P_{label} = e^{-d_2/d_1} \tag{8}$$

The idea behind this formula was that the farther away from the contour of one tree, the lower the probability of belonging to this tree. Figure 11b,d show the effect of the updated segmentation and the refined contour.



**Figure 11.** The segmentation was refined by partitioning all the unclassified points and optimizing the layer contour.

Since our algorithm included a random classification strategy for the overlap region points, its effect achieved the simulation of the intersection of the crown contour. The algorithm was called the *soft segmentation* (SoftSeg) algorithm.

### 2.3. Reconstruction

After the terrestrial laser scan point clouds have been soft segmented, the points of a tree are taken as the input for crown surface reconstruction. The input data, denoted as  $\Omega$ , include  $N$  points, and each point  $P_i (i = 1, 2, \dots, N)$  were represented only by its position coordinates  $(x, y, z)$ , without regard to other information that laser scanners may generate besides the position coordinates. The direction of the  $z$ -axis is upward, which will guide the horizontal cutting of our method.

After inputting the laser scan point cloud of a tree, we first segment crown points (denoted as  $C$ ) out from branches, then reconstruct the crown as following steps.

#### 2.3.1. Detecting Boundary Points of Bins

For each layer bin, the boundary points were detected based on a 2D  $\alpha$ -shape. The classical  $\alpha$ -shape method was introduced by H. Edelsbrunner in 1983 [40], and Bernardini et al. [41] improved it to probe the boundary of a point set by setting  $\alpha = r^2$ , in which  $r$  is the radius of a disc in a 2D plane or a ball in 3D space. In our case, we projected all the points of a bin onto a horizontal plane  $xOy$ , as shown in Figure 10, and obtained the boundary points of each original bin based on Bernardini's method [41].

#### 2.3.2. Building the Crown Surface

After obtaining the boundary points of each bin, the crown surface geometry was constructed based on a 3D  $\alpha$ -shape [42]. Let  $\Lambda$  be the set of all the boundary points  $Q_j (j = 1, 2, \dots, n_b)$  of the crown. We constructed a kd-tree data structure [43] on  $\Lambda$  to speed up the procedure of constructing the surface. The Algorithm 1 known as the "Building Crown Surface with Boundary Points" (or BCSwBPs) was given in pseudocode as follows.

**Algorithm 1:** BCSwBPs.Input:  $\Lambda, \alpha$ 

Output: crown surface geometry

S1: construct a kd-tree with all points in the set  $\Lambda$ S2: for  $\forall Q_i, Q_j, Q_k \in \Lambda$ S3: calculate two centers  $C_1, C_2$  of two balls  $B_1, B_2$ . Note that  $Q_i, Q_j, Q_k$  are on  $B_m$  and  $|C_m Q_n| = \alpha, m = 1, 2; n = i, j, k$ .S4: Find a nearest neighbor  $Q_{d1}$  of  $C_1$ S5: Find a nearest neighbor  $Q_{d2}$  of  $C_2$ S6: if  $Q_{d1} \notin B_1$  or  $Q_{d2} \notin B_2$ , thenS7:  $\Delta Q_i Q_j Q_k$  is a boundary triangle for output

S8: end if

S9: end for

**2.3.3. Estimating the Attributes**

With the reconstructed crown surface, several crown attributes, including the width, height, superficial area, and projecting ground area, were estimated.

The width  $W_{crown}$  of a tree crown was estimated according to Formula (9).

$$W_{crown} = \max \left\{ \left| \overline{P'_i P'_j} \right|, \forall P'_i, P'_j \in \Gamma \right\} \quad (9)$$

where  $\Gamma$  is the boundary of the projection points set that vertically mapped  $\Lambda$  onto the  $xOy$  plane.

The height  $H_{crown}$  of the crown was estimated using the laser scanning point data.

$$H_{crown} = Z_{\max} - Z_{\min} \quad (10)$$

where  $Z_{\max}$  and  $Z_{\min}$  are the maximum and minimum z-coordinates of the crown.

The superficial area of the crown was the sum of the areas of all the triangles on the surface of the crown. In the traditional  $\alpha$ -shape method, the triangles in the crown were turned into boundary triangles if the radius  $r$  of the probing ball was too small. This led to a bigger superficial area than the real case. If the  $r$  was set too large, the reconstructed crown would become a convex hull, which would also induce a significant error. As our method only used the boundary points to build the crown surface, the superficial area only consisted of the boundary triangles, making the error small.

For calculating the projecting ground area, all the points of  $\Lambda$  were projected onto the  $xOy$  plane, and a polygon  $\mathcal{P}$  was built on these projected points by employing a 2D  $\alpha$ -shape method. Note that the polygon  $\mathcal{P}$  may not always be convex, and the built  $\mathcal{P}$  based on projecting  $C$  or  $\Lambda$  onto the  $xOy$  plane is the same.

For calculating the reconstructed tree crown volume, the geometrical information of each layer bin could be used again. The volume was estimated using Formula (11).

$$V_{tree} = \frac{1}{3} \Delta_b \sum_{i=0}^{k-1} \left( S_i + S_{i+1} + \sqrt{S_i S_{i+1}} \right) \quad (11)$$

where  $S_i$  is the cross-sectional area of the  $i^{th}$  layer bin.

**3. Results**

We demonstrated the effectiveness of the SoftSeg and BCSwBPs algorithms based on the point cloud segmentation and reconstruction experiments. Our algorithm was written in C++ language with the support of OpenGL for visualization. The experiments were conducted on a laptop with an Intel(R) Core(TM) i7-4710MQ CPU@2.5GHz and a 4 GB RAM.

### 3.1. Segmentation of the Tree Crown with Different Overlap Degrees

A pair of trees composed of the same two trees, if their distances are different, means that the degree of overlap between tree crowns is also different. The experiment was to test the segmentation effect of the point cloud from two trees at different distances. As shown in the first column of Figure 12, the distance between two tree roots was 6.464, 6.962, 7.46, 7.959, 8.457, 8.956, and 9.455 m, respectively. These seven point clouds were segmented using our SoftSeg algorithm, and the result of each step is displayed in Figure 12. It can be found that the overlap region was classified in the view of the crown shape, which showed a realistic silhouette.

For quantitatively evaluating the effect of the segmentation of our method, we computed the accuracy (Ac) and precision (Pre) and compared our results to those from the hard segmentation (Formula (5)).

The accuracy was calculated using Formula (12).

$$Ac = (m11 + m22) / N \quad (12)$$

The sensitivity was calculated using Formula (13).

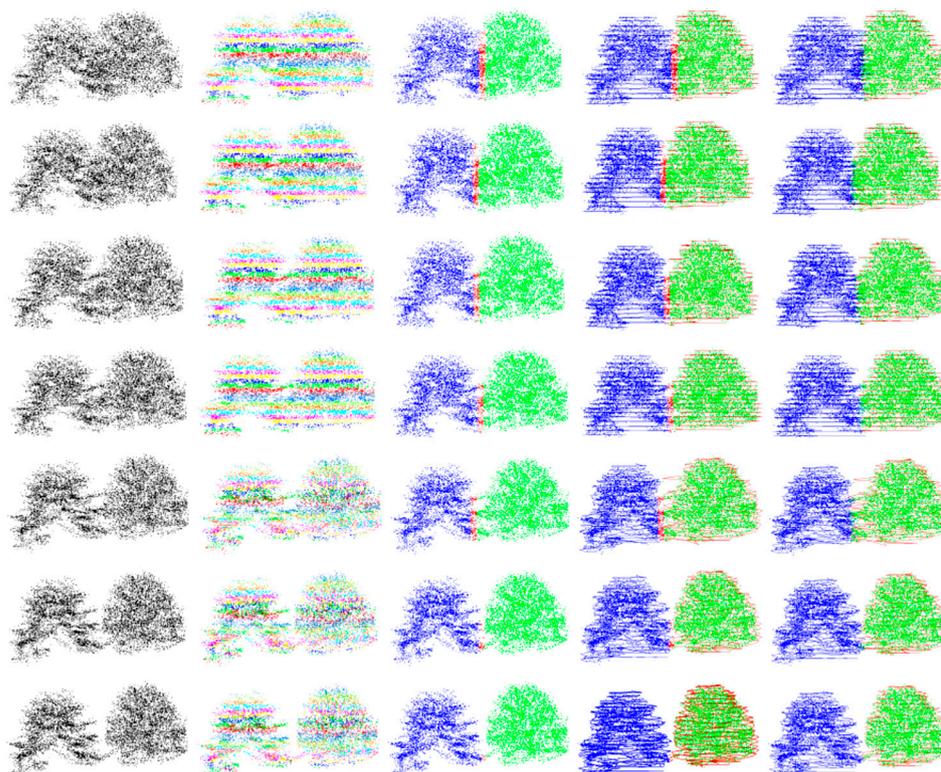
$$Pre = m11 / (m11 + m21) \quad (13)$$

where  $m11$  is the number of points correctly classified to Tree 1;  $m22$  is the number of points correctly classified to Tree 2;  $m21$  is the number of points wrongly classified to Tree 1; and  $N$  is the total number of points of the input data.

We compared the results obtained by the proposed soft segmentation method (SS, or SoftSeg) to the results obtained by the hard segmentation method (HS, Formula (5) in Section 2.2.2). The quantitative results are listed in Table 1. In the table, the number of points in Tree 1 and Tree 2 were 3253 and 2923, respectively.

**Table 1.** The confusion matrixes of the hard segmentation and soft segmentation methods. Each matrix was  $2 \times 2$ , and the elements on the diagonal of a matrix represent the correct number of segmentation points or the corresponding ratio of points.

SN	Class	HS.T1	HS.T2	HS.T1p	HS.T2p	SS.T1	SS.T2	SS.T1p	SS.T2p
Row1	Tru.T1	3092	161	0.9505	0.0495	3078	175	0.9462	0.0538
	Tru.T2	238	2685	0.0814	0.9186	227	2696	0.0777	0.9223
Row2	Tru.T1	3250	3	0.9991	0.0009	3242	11	0.9966	0.0034
	Tru.T2	365	2558	0.1249	0.8751	341	2582	0.1167	0.8833
Row3	Tru.T1	3253	0	1	0	3251	2	0.9994	0.0006
	Tru.T2	280	2643	0.0958	0.9042	270	2653	0.0924	0.9076
Row4	Tru.T1	3253	0	1	0	3251	2	0.9994	0.0006
	Tru.T2	180	2743	0.0616	0.9384	162	2761	0.0554	0.9446
Row5	Tru.T1	3253	0	1	0	3253	0	1	0
	Tru.T2	285	2638	0.0975	0.9025	279	2644	0.0954	0.9046
Row6	Tru.T1	3253	0	1	0	3253	0	1	0
	Tru.T2	52	2871	0.0178	0.9822	42	2881	0.0144	0.9856
Row7	Tru.T1	3253	0	1	0	3253	0	1	0
	Tru.T2	22	2901	0.0075	0.9925	18	2905	0.0062	0.9938



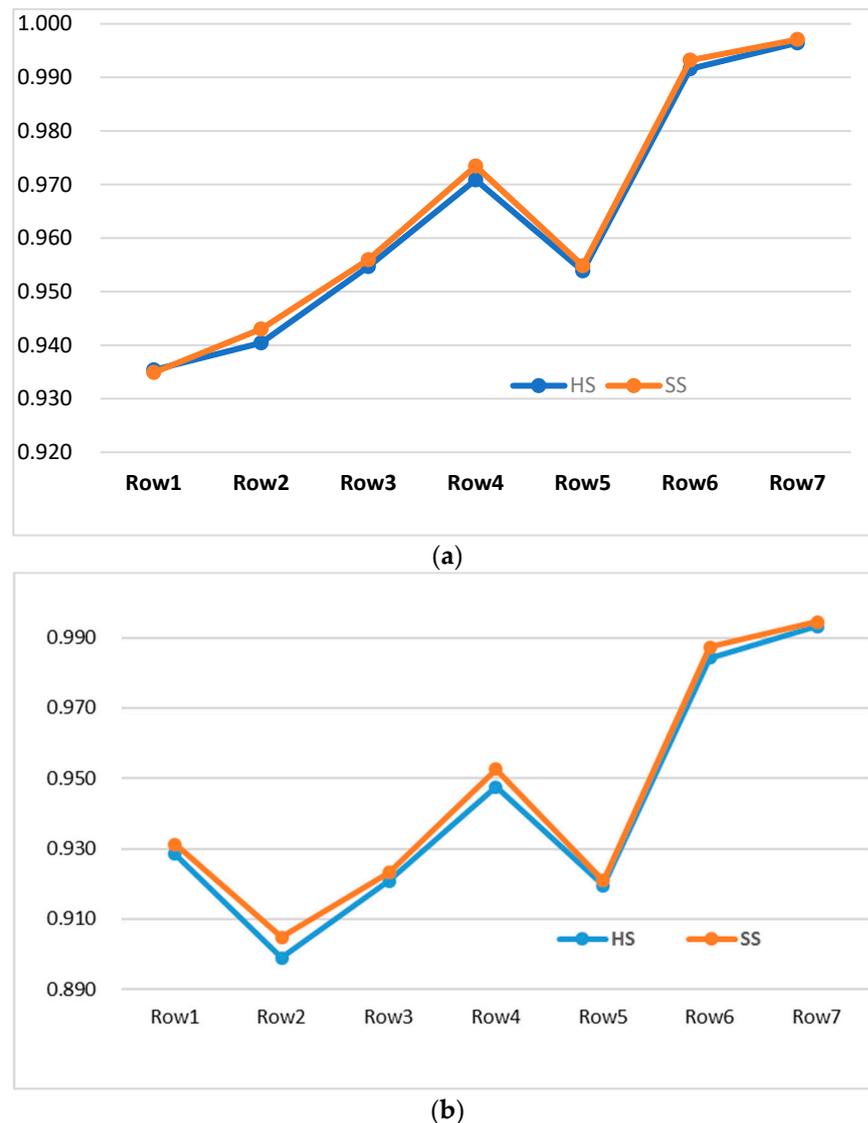
**Figure 12.** The segmentation of two trees pieced together at different distances. From top to bottom, the distance between the two tree roots was 6.464, 6.962, 7.46, 7.959, 8.457, 8.956, and 9.455 m, respectively. From left to right: the input, layer bins, initial segmentation, initial contours of layer bins, refined segmentation, and contours.

As for the symbols in the table in the first column (series number, SN) of the table, Row  $i$  ( $i = 1, 2, \dots, 7$ ) consisted of the point cloud data from the  $i^{\text{th}}$  row in Figure 12. Tru.T1 is the number of points that belongs to Tree 1. HS.T1 means that those points are classified into Tree 1 using the hard segmentation method. HS.T1p is the ratio of HS.T1 to the number of points from Tree 1. “SS” is the abbreviation of “SoftSeg”. Similarly, we defined the other symbols, including HS.T2, HS.T2p, SS.T1, SS.T1p, SS.T2, and SS.T2p.

Using the data listed in Table 1 and with Formulas (12) and (13), the values of the Ac and Pre were calculated and are displayed in Figure 13. From the line charts, we can conclude that as the overlapping area of the two trees decreased, the accuracy of segmentation continued to increase. Both line charts showed that compared to the HS algorithm, the SS algorithm improved slightly in both the accuracy and precision. Due to the small proportion of the points in the overlapping part of the crown of the two trees to the total number of points, the segmentation accuracy and precision seemed to improve only slightly. However, in view of the sum of misclassification points in Table 1, the misclassification rate of the SS is approx. 4% lower than that of the HS.

### 3.2. Segmentation and Comparison

Four pair of trees, OaklandTrees (Figure 3b), RUSH07treesA (Figure 4b), RUSH07treesB (Figure 4c), and RUSH07treesC (Figure 4d), were used for testing the performance of our proposed method (SoftSeg).

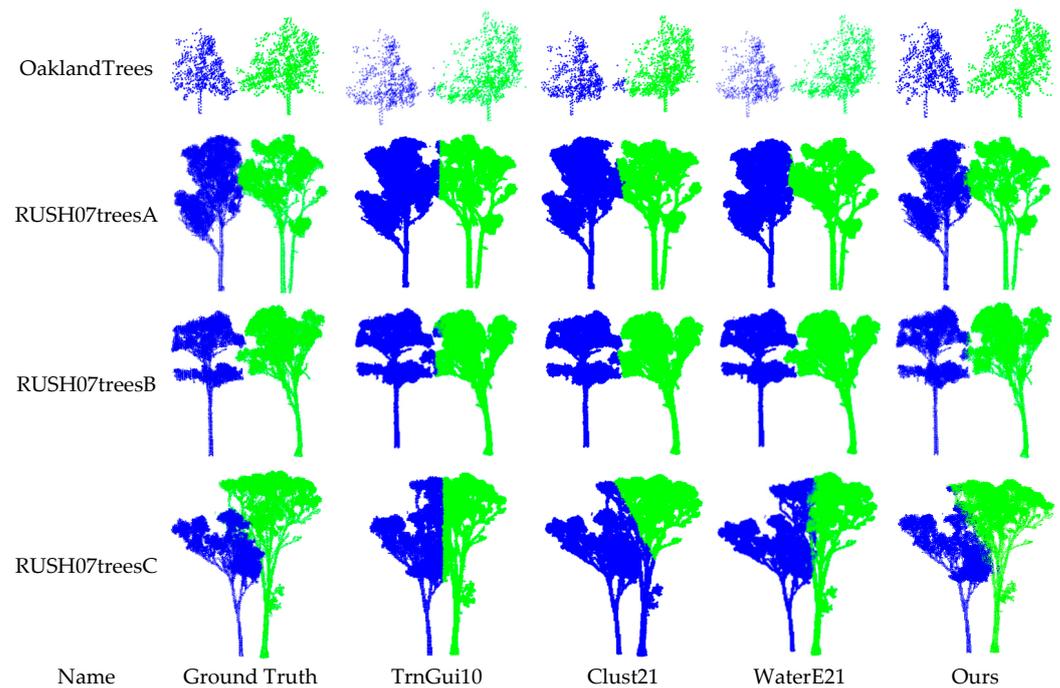


**Figure 13.** Comparison of the Ac and Pre of the segmentation using the hard segmentation (HS) method and the soft segmentation (SS) method. (a) The accuracy (Ac) of the segment results of the HS and SS methods. (b) The precision (Pre) of the segment results of the HS and SS methods.

The experimental results of our method were compared to those of three representative methods: trunk guided crown segmentation (TrnGui10) [44], DBSCAN-based clustering (Clust21) [8], and the water expansion method (WaterE21) [45]. The segmentation results of the four point clouds after down-sampling are shown in Figure 14.

By comparing the segmented results to the ground truth (the second column in Figure 14), it can be found that when there was an apparent valley between the adjacent trees, WaterE21 showed the best performance in the view of the visual effect. Our method achieved the best performance among the four listed methods when RUSH07treesC was segmented.

For the quantitative comparison, the confusion matrixes of the four pair of trees segmented using the four methods are listed in Table 2. In the table, “Pts.N” means the number of points of the corresponding point cloud data. The bold confusion matrixes achieved the best results among the four methods. This table showed that both the WaterE21 and our methods achieved the best segmentation results since the diagonal elements in their confusion matrix were relatively large.



**Figure 14.** Visual comparison of four pair trees which are segmented using four methods.

**Table 2.** The confusion matrixes of the four pair of trees segmented using the four methods.

Name	Pts.N	TrnGui10	Clust21	WaterE21	Ours
OaklandTrees	1370	1351	19	1319	51
	834	0	834	0	834
RUSH07TreesA	33,424	27,505	5919	29,072	4352
	43,286	0	43,286	0	43,286
RUSH07TreesB	33,849	32,592	1257	33,524	325
	27,065	0	27,065	1	27,064
RUSH07TreesC	34,053	23,255	10,798	25,063	8990
	24,057	2106	21,951	0	24,057

According to the confusion matrixes and Formula (11), the accuracy (Ac) of the segmentation results could be calculated. The values of the segmentation accuracy of the four pair point clouds segmented using the four methods are displayed as line chart in Figure 15. From the perspective of the segmentation accuracy, these four segmentation methods accurately segmented the three point cloud data, OaklandTrees, RUSH07treesA, and RUSH07treesB, and the segmentation accuracies were all greater than 90%. However, for the segmentation of RUSH07treesC, only our method had a segmentation accuracy exceeding 90%, showing an optimal robustness.

The average accuracies of the four segmentation results were 91.79%, 94.00%, 95.05%, and 97.06%, obtained by the four methods, TrnGui10, Clust21, WaterE21, and ours, respectively. This means that in the experiment of segmenting these four point cloud data, our algorithm achieved the best performance.

### 3.3. Reconstruction Results

For the two pine trees, we used our method to construct each tree crown, respectively, as shown in Figure 16.

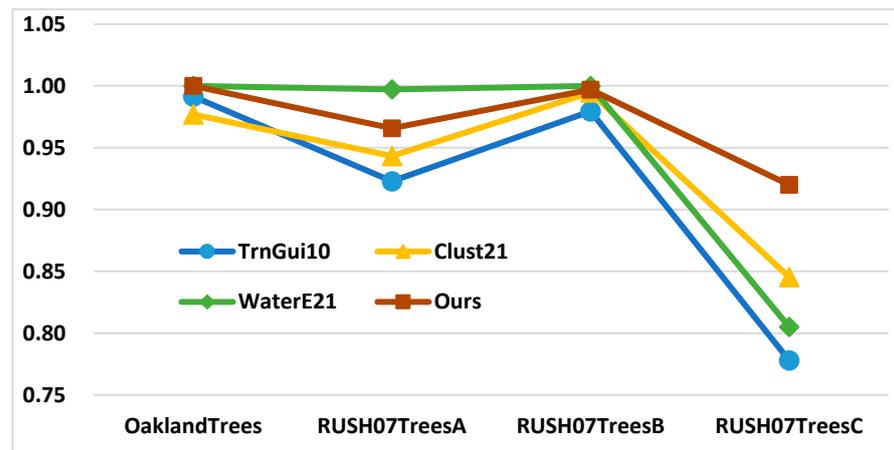


Figure 15. The accuracy (Ac) comparison of the four pair of trees which were segmented using the four methods.

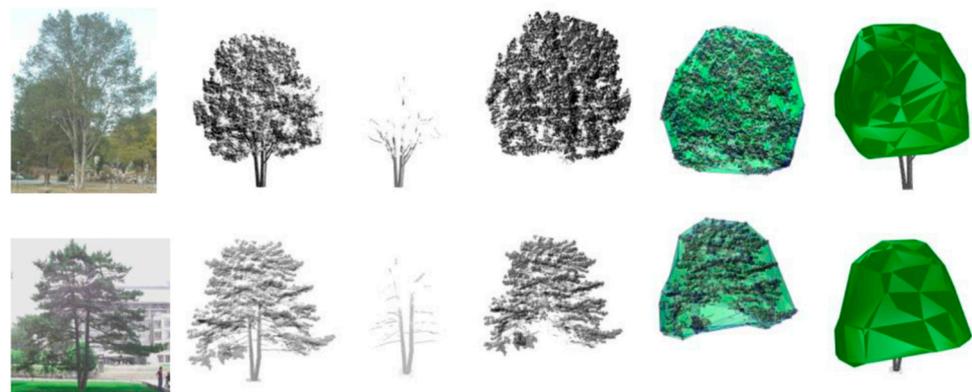


Figure 16. Crown reconstruction process illustrated with pine trees denoted as Pine 1 and Pine 2. Each row shows the tree crown reconstruction results. The first column is a photo of the tree. The second column is the laser scan point cloud. The third points from branches, the fourth the points from the crown, and the fifth shows the reconstructed crown silhouette surface, which was merged with the crown points. The last column shows the reconstructed crown merged with the branch points.

The related attributes are listed in Table 3, with “SN” denoting the serial number of the tree, “N.Pts” the number of scanning points, “H.Tree” the height (m) of the tree, “W.Tree” the width (m) of the tree, “A.Sup” the superficial area (m<sup>2</sup>) of the reconstructed crown, and “A.proj” the projection area (m<sup>2</sup>) of the reconstructed crown.

Table 3. Attributes of the two pine trees (Figure 16).

SN	N.Pts	H.Tree	W.Tree	A.Sup	A.proj
Tree1	487,555	11.9	10.1	292.5	71.7
Tree2	269,366	11.1	11.1	303.3	73.1

For the point cloud, Oaklandtrees (Figure 3), after the segmentation with our SoftSeg method, all the tree shape silhouette meshes were reconstructed, as illustrated in Figure 17. The attributes of all the trees are listed in Table 4. In this table, the volume was calculated using Formula (11), and its unit was m<sup>3</sup>. As shown in Table 4, the point cloud of each segmented tree was very sparse, and the accuracy of the calculated attribute values needs further verification in future research.

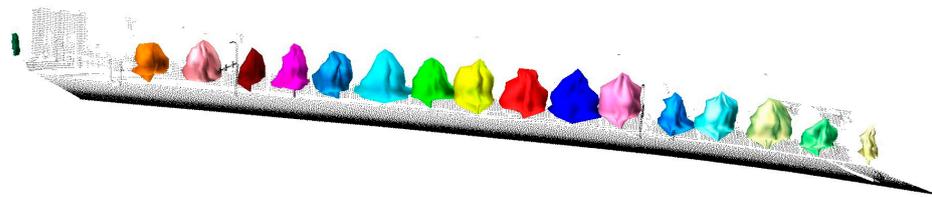


Figure 17. Crown reconstruction results using the Oakland data.

Table 4. Attributes of the segmented 17 trees (Figure 17).

Tree SN	N.Pts	H.Tree	W.Tree	A.Sup	Volume
1	977	7.138	6.515	113.109	64.803
2	435	6.747	3.334	46.002	12.153
3	978	6.654	6.454	108.756	48.789
4	1527	8.745	7.293	176.111	123.144
5	1370	8.601	7.687	158.965	108.512
6	834	7.540	5.727	112.835	54.985
7	1267	10.064	7.283	185.166	121.502
8	1427	9.610	9.353	220.818	153.500
9	1319	9.003	8.101	176.291	115.188
10	1042	9.816	7.222	172.671	108.886
11	1083	9.157	8.454	168.591	109.401
12	1333	9.507	9.000	273.591	174.541
13	1024	8.858	7.363	182.162	98.426
14	777	7.880	5.909	112.269	49.672
15	677	8.055	4.576	96.327	37.486
16	834	7.437	6.111	130.311	65.083
17	122	4.265	1.394	13.499	1.379

For the three point clouds of the three pair of trees, as shown in Figure 4, after the segmentation using the SoftSeg method, the reconstructed mesh models of the individual trees are displayed in Figure 18. To enhance the visual differentiation, the different pair trees models used different colors. Visually, these silhouette mesh models have a more natural appearance.

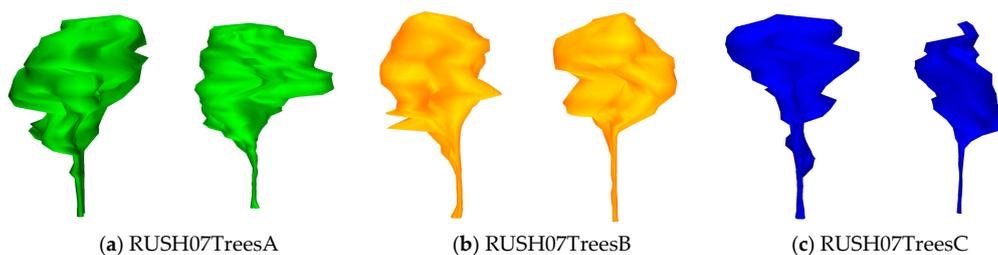


Figure 18. The reconstructed tree silhouette mesh model from the point clouds after using the soft segmentation method.

### 3.4. Discussion

#### 3.4.1. Time Efficiency with Down-Sampling

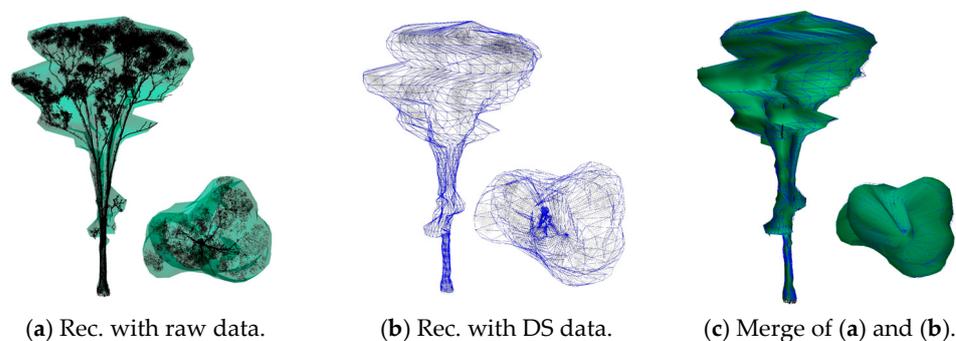
A down-sampling step was employed to improve the time efficiency of the individual tree segmentation processing. The time (seconds) of the segmentation of RUSH07TreesA is recorded in Table 5. In the table, “DS” means the down-sampling step. From the table, we can safely conclude that the down-sampling step significantly improved the algorithm efficiency because the time cost of the method without the DS step was approx. 28 times of the efficiency with the DS step.

**Table 5.** The running time (s) of each step of the algorithm for segmenting RUSH07TreesA.

Method	DS	Roots Detect	Layer Bin Build	Vertical Partitioning	Contour Build	Segmentation Refine	Total Time
With DS	0.068	0.3446	0.0189	0.0107	0.0119	0.0023	0.4564
Without DS	0	12.4051	0.2433	0.1549	0.2651	0.0215	13.0899

### 3.4.2. Error Caused by Down-Sampling

As for the implications of down-sampling, it is possible to cause minor errors when the attributes, including the volume and surface area of the tree crowns, are estimated. However, the impact of down-sampling on the silhouette shape of the tree crowns is relatively tiny. We used one tree extracted from RUSH07TreesC to check the implications. In this experiment, the layer number  $k$  was set to 38 (different from the value used in Figure 18), about 0.5 m thick. The reconstructed silhouette surface mesh is illustrated in Figure 19. The reconstructed mesh consisted of 1369 vertices and 2700 triangles. The average length of the triangle edge was 0.8481 m, and the average area of all the triangles was  $0.202 \text{ m}^2$ .



**Figure 19.** The reconstructed mesh models used point clouds with/without down-sampling (DS). A subfigure at the bottom right of each figure is the top view of the reconstructed model.

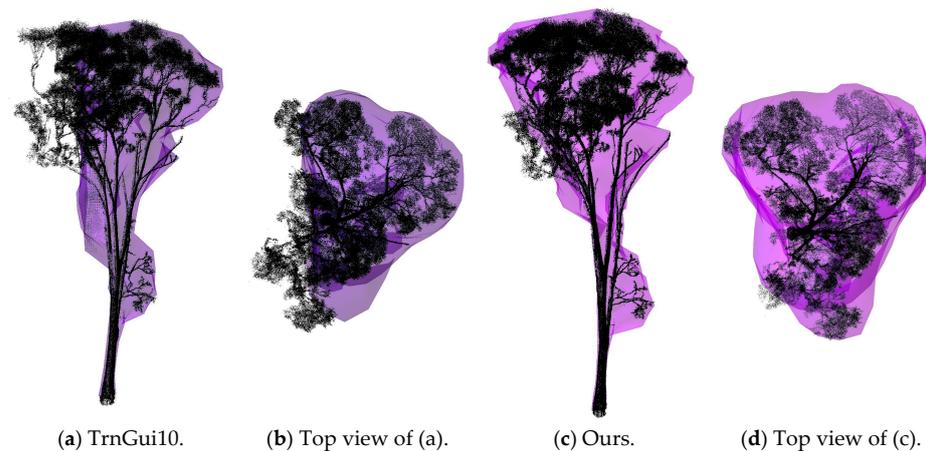
With the reconstructed shape, the attributes were estimated and are listed in Table 6. The table shows that the relative error was very small (less than 1.6%), even though the number of the model points was compressed by 93%. Therefore, the down-sampling step can be introduced into the algorithm process to improve the algorithm efficiency.

**Table 6.** Attributes of one tree from RUSH07TreesC (Figure 18).

Method	N.Pts	N.Polygon	H.Tree	W.Tree	A.Sup	Volume
With DS	34,053	2700	23.176	13.333	545.831	532.967
Without DS	536,461	2700	23.318	13.412	549.142	525.064
Error	−93.65%	0.00%	−0.61%	−0.59%	−0.60%	1.51%

### 3.4.3. Visual Effect of the Reconstructed Crown

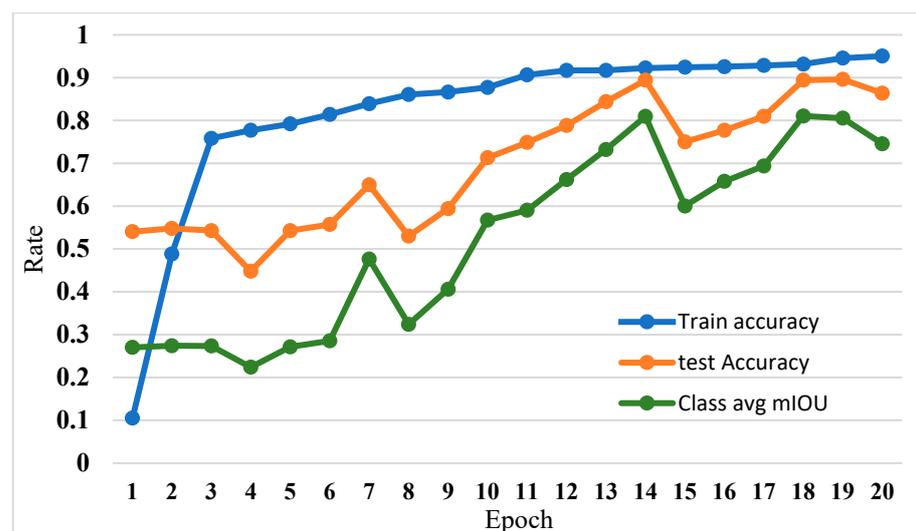
In addition, the proposed SoftSeg method had an advantage over the hard segmentation methods, for example, TrnGui10 [44], in that the visual effect of the crown shape that was reconstructed using the point cloud segmented by SoftSeg was often better than that of the hard segmentation method. Figure 20 illustrates the visual effect by comparing our method to the reconstructed crown from the segmented point cloud using TrnGui10 method. The points were ground truth. The purple silhouette surface was the reconstructed tree shape. Obviously, the hard segmentation method often caused the vertical segmentation plane, decreasing the visual reality.



**Figure 20.** Visual comparison of the reconstructed tree silhouette mesh models from the point cloud RUSH07TreesC using the TrnGui10 and our SoftSeg methods.

#### 3.4.4. Segmentation Using the Deep Learning Method

We did not consider deep learning methods as a comparative method but rather part of the discussion. This was primarily because we did not have enough labeled data for the training set. We adopted the Pointnet++ [46] to segment the trees. The test set consisted of one point cloud in Figure 12 and four point clouds in Figure 14. The training set consisted of six point clouds in Figure 12 and one point cloud in Figure 14. After setting the parameters, `batch_size = 2`, `decay_rate = 0.0001`, `epoch = 20`, `learning_rate = 0.001`, `lr_decay = 0.5`, `optimizer = 'Adam'`, `step_size = 20`, the line charted the training accuracy, the test accuracy, and the class average mIOU in the training and the testing processes are displayed in Figure 21. The last test accuracy was approx. 90.0%, which was far less than the segmentation accuracy of our method. However, with the establishment of labeled datasets for forest segmentation in the future, deep learning methods will provide excellent performance.



**Figure 21.** The training accuracy, the test accuracy, and the class average mIOU varied in the training and testing process.

## 4. Conclusions

In this paper, we proposed a novel tree crown soft segmentation algorithm called SoftSeg, which was automatic if the number of layer bins and the overlap width was assigned as the default values. The experiments showed that the new algorithm could

effectively segment the different overlapping degrees of the crown of two trees. Compared to hard segmentation, it could improve the segmentation accuracy, over 90%, and reduce the misclassification point. The crown silhouette shape reconstructed from the points segmented by SoftSeg was more realistic than that obtained using the hard segmentation method. In addition, to improve the algorithm efficiency, two strategies, down-sampling and kd-tree, were used in the proposed method. In practical applications, if it was necessary to study the distribution characteristics of the branches and leaves in the overlapping area of the tree crowns, the proposed method could be an alternative solution.

Our algorithm had several limitations. First, the reconstructed mesh quality should be given more attention, which could be improved by employing an existing mesh optimal algorithm. Second, when the point cloud was severely missing, down-sampling could not supplement enough missing information, so the reconstructed crown shape was also incomplete. Solving the reconstruction of the severely missing point cloud data could also be a challenging task. Another limitation was that the segmentation results had apparent errors, and further improvement is needed in the future.

**Author Contributions:** Conceptualization, M.D.; methodology, M.D. and G.L.; software, M.D.; validation, M.D.; formal analysis, M.D.; investigation, M.D.; resources, M.D. and G.L.; data curation, M.D.; writing—original draft preparation, M.D.; writing—review and editing, M.D. and G.L.; visualization, M.D.; supervision, G.L.; project administration, G.L.; funding acquisition, M.D. and G.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding. This research was funded by the China Academy of Railway Sciences Corporation Limited, grant number 2021YJ197.

**Data Availability Statement:** The first author can provide point clouds of two pine trees, please send an email to get in contact; for RUSH07, please refer to TERN, <https://portal.tern.org.au/metadata/23868>, accessed on 1 March 2023; for OaklandTrees, please visit [http://www.cs.cmu.edu/~vmr/datasets/oakland\\_3d/cvpr09/doc/](http://www.cs.cmu.edu/~vmr/datasets/oakland_3d/cvpr09/doc/), accessed on 18 May 2023.

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

## References

1. Jakubowski, M.K.; Li, W.; Guo, Q.; Kelly, M. Delineating Individual Trees from LiDAR Data: A Comparison of Vector- and Raster-based Segmentation Approaches. *Remote Sens.* **2013**, *5*, 4163–4186. [CrossRef]
2. 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]
3. Zhang, C.; Zhou, Y.; Qiu, F. Individual Tree Segmentation from LiDAR Point Clouds for Urban Forest Inventory. *Remote Sens.* **2015**, *7*, 7892–7913. [CrossRef]
4. Wang, J.; Lindenbergh, R.; Menenti, M. Scalable individual tree delineation in 3D point clouds. *Photogramm. Rec.* **2018**, *33*, 315–340. [CrossRef]
5. Yang, J.; Kang, Z.; Cheng, S.; Yang, Z.; Akwensi, P.H. An Individual Tree Segmentation Method Based on Watershed Algorithm and Three-Dimensional Spatial Distribution Analysis From Airborne LiDAR Point Clouds. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 1055–1067. [CrossRef]
6. Irlan, I.; Saleh, M.B.; Prasetyo, L.B.; Setiawan, Y. Evaluation of Tree Detection and Segmentation Algorithms in Peat Swamp Forest Based on LiDAR Point Clouds Data. *J. Manaj. Hutan Trop. J. Trop. For. Manag.* **2020**, *26*, 123–132. [CrossRef]
7. Ayrey, E.; Fraver, S.; Kershaw, J.A., Jr.; Kenefic, L.S.; Hayes, D.; Weiskittel, A.R.; Roth, B.E. Layer Stacking: A Novel Algorithm for Individual Forest Tree Segmentation from LiDAR Point Clouds. *Can. J. Remote Sens.* **2017**, *43*, 16–27. [CrossRef]
8. Chen, Q.; Wang, X.; Hang, M.; Li, J. Research on the improvement of single tree segmentation algorithm based on airborne LiDAR point cloud. *Open Geosci.* **2021**, *13*, 705–716. [CrossRef]
9. Bienert, A.; Georgi, L.; Kunz, M.; von Oheimb, G.; Maas, H.-G. Automatic extraction and measurement of individual trees from mobile laser scanning point clouds of forests. *Ann. Bot.* **2021**, *128*, 787–804. [CrossRef]
10. Liu, H.; Dong, P.; Wu, C.; Wang, P.; Fang, M. Individual tree identification using a new cluster-based approach with discrete-return airborne LiDAR data. *Remote Sens. Environ.* **2021**, *258*, 112382. [CrossRef]
11. Qin, Y.; Ferraz, A.; Mallet, C.; Iovan, C. Individual tree segmentation over large areas using airborne LiDAR point cloud and very high resolution optical imagery. In Proceedings of the 2014 IEEE Geoscience and Remote Sensing Symposium, Quebec City, QC, Canada, 13–18 July 2014; pp. 800–803. [CrossRef]
12. Comesaña-Cebral, L.; Martínez-Sánchez, J.; Lorenzo, H.; Arias, P. Individual Tree Segmentation Method Based on Mobile Backpack LiDAR Point Clouds. *Sensors* **2021**, *21*, 6007. [CrossRef] [PubMed]

13. Lisiewicz, M.; Kamińska, A.; Kraszewski, B.; Stereńczak, K. Correcting the Results of CHM-Based Individual Tree Detection Algorithms to Improve Their Accuracy and Reliability. *Remote Sens.* **2022**, *14*, 1822. [[CrossRef](#)]
14. Huo, L.; Lindberg, E.; Holmgren, J. Towards low vegetation identification: A new method for tree crown segmentation from LiDAR data based on a symmetrical structure detection algorithm (SSD). *Remote Sens. Environ.* **2022**, *270*, 112857. [[CrossRef](#)]
15. Li, J.; Cheng, X.; Xiao, Z. A branch-trunk-constrained hierarchical clustering method for street trees individual extraction from mobile laser scanning point clouds. *Measurement* **2021**, *189*, 110440. [[CrossRef](#)]
16. Pires, R.D.P.; Olofsson, K.; Persson, H.J.; Lindberg, E.; Holmgren, J. Individual tree detection and estimation of stem attributes with mobile laser scanning along boreal forest roads. *ISPRS J. Photogramm. Remote Sens.* **2022**, *187*, 211–224. [[CrossRef](#)]
17. Braga, J.R.G.; Peripato, V.; Dalagnol, R.; Ferreira, M.P.; Tarabalka, Y.; Araújo, L.E.O.C.; Velho, H.F.d.C.; Shiguemori, E.H.; Wagner, F.H. Tree Crown Delineation Algorithm Based on a Convolutional Neural Network. *Remote Sens.* **2020**, *12*, 1288. [[CrossRef](#)]
18. Janoutová, R.; Homolová, L.; Novotný, J.; Navrátilová, B.; Píkl, M.; Malenovský, Z. Detailed reconstruction of trees from terrestrial laser scans for remote sensing and radiative transfer modelling applications. *Silico Plants* **2021**, *3*, diab026. [[CrossRef](#)]
19. Dai, M.; Li, H.; Zhang, X. Tree Modeling through Range Image Segmentation and 3D Shape Analysis. In *Lecture Notes in Electrical Engineering Book Series (LNEE)2010*; Springer: Berlin/Heidelberg, Germany, 2010; Volume 67, pp. 413–422. [[CrossRef](#)]
20. Livny, Y.; Yan, F.; Olson, M.; Chen, B.; Zhang, H.; El-Sana, J. Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graph.* **2010**, *29*, 151:1–151:8. [[CrossRef](#)]
21. Zhang, X.; Li, H.; Dai, M.; Ma, W.; Quan, L. Data-driven synthetic modeling of trees. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 1214–1226. [[CrossRef](#)]
22. Wang, Z.; Zhang, L.; Fang, T.; Mathiopoulos, P.T.; Qu, H.; Chen, D.; Wang, Y. A Structure-Aware Global Optimization Method for Reconstructing 3-D Tree Models From Terrestrial Laser Scanning Data. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 5653–5669. [[CrossRef](#)]
23. Moorthy, I.; Miller, J.R.; Berni, J.A.J.; Zarco-Tejada, P.; Hu, B.; Chen, J. Field characterization of olive (*Olea europaea* L.) tree crown architecture using terrestrial laser scanning data. *Agric. For. Meteorol.* **2011**, *151*, 204–214. [[CrossRef](#)]
24. Paris, C.; Kelbe, D.; van Aardt, J.; Bruzzone, L. A Novel Automatic Method for the Fusion of ALS and TLS LiDAR Data for Robust Assessment of Tree Crown Structure. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3679–3693. [[CrossRef](#)]
25. Alexander, C. Delineating tree crowns from airborne laser scanning point cloud data using delaunay triangulation. *Int. J. Remote Sens.* **2009**, *30*, 3843–3848. [[CrossRef](#)]
26. Morsdorf, F.; Meier, E.; Kötz, B.; Itten, K.I.; Dobbertin, M.; Allgöwer, B. LIDAR-based geometric reconstruction of boreal type forest stands at single tree level for forest and wildland fire management. *Remote Sens. Environ.* **2004**, *92*, 353–362. [[CrossRef](#)]
27. Kato, A.; Schreuder, G.F.; Calhoun, D.; Schiess, P.; Stuetzle, W. Digital surface model of tree canopy structure from LiDAR data through implicit surface reconstruction. In Proceedings of the ASPRS 2007 Annual Conference, Tampa, FL, USA, 7–11 May 2007.
28. Kato, A.; Moskal, L.M.; Schiess, P.; Swanson, M.E.; Calhoun, D.; Stuetzle, W. Capturing tree crown formation through implicit surface reconstruction using airborne lidar data. *Remote Sens. Environ.* **2016**, *113*, 1148–1162. [[CrossRef](#)]
29. Lin, Y.; Hyypä, J. Multiecho-Recording Mobile Laser Scanning for Enhancing Individual Tree Crown Reconstruction. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 4323–4332. [[CrossRef](#)]
30. Zhu, C.; Zhang, X.; Hu, B.; Jaeger, M. *Reconstruction of Tree Crown Shape from Scanned Data*; Springer: Berlin/Heidelberg, Germany, 2008.
31. Tang, S.; Dong, P.; Buckles, B.P. Three-dimensional surface reconstruction of tree canopy from lidar point clouds using a region-based level set method. *Int. J. Remote Sens.* **2012**, *34*, 1373–1385. [[CrossRef](#)]
32. Xie, D.; Wang, X.; Qi, J.; Chen, Y.; Mu, X.; Zhang, W.; Yan, G. Reconstruction of Single Tree with Leaves Based on Terrestrial LiDAR Point Cloud Data. *Remote Sens.* **2018**, *10*, 686. [[CrossRef](#)]
33. Kim, D.; Jo, K.; Lee, M.; Sunwoo, M. L-shape model switching-based precise motion tracking of moving vehicles using laser scanners. *IEEE Trans. Intell. Transp. Syst.* **2017**, *19*, 598–612. [[CrossRef](#)]
34. Ma, Q.; Su, Y.; Tao, S.; Guo, Q. Quantifying individual tree growth and tree competition using bi-temporal airborne laser scanning data: A case study in the Sierra Nevada Mountains, California. *Int. J. Digit. Earth* **2017**, *11*, 485–503. [[CrossRef](#)]
35. Yu, X.; Hyypä, J.; Kaartinen, H.; Hyypä, H.; Maltamo, M.; Rnnholm, P. Measuring the growth of individual trees using multi-temporal airborne laser scanning point clouds. In Proceedings of the ISPRS Workshop on “Laser Scanning 2005”, Enschede, The Netherlands, 12–14 September 2005; pp. 204–208.
36. Aubry-Kientz, M.; Dutrieux, R.; Ferraz, A.; Saatchi, S.; Hamraz, H.; Williams, J.; Coomes, D.; Piboule, A.; Vincent, G. A Comparative Assessment of the Performance of Individual Tree Crowns Delineation Algorithms from ALS Data in Tropical Forests. *Remote Sens.* **2019**, *11*, 1086. [[CrossRef](#)]
37. Daniel Muñoz, J.; Bagnell, A.; Vandapel, N.; Hebert, M. Contextual Classification with Functional Max-Margin Markov Networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009.
38. Calders, K. *Terrestrial Laser Scans—Riegl VZ400, Individual Tree Point Clouds and Cylinder Models, Rushworth Forest*; Version 1; Terrestrial Ecosystem Research Network: Indooroopilly, QLD, Australia, 2014. [[CrossRef](#)]
39. Fang, H.; Li, H. *Counting of Plantation Trees Based on Line Detection of Point Cloud Data*; Geomatics and Information Science of Wuhan University: Wuhan, China, 2022; Volume 7. [[CrossRef](#)]

40. Edelsbrunner, H.; Kirkpatrick, D.; Seidel, R. On the shape of a set of points in the plane. *IEEE Trans. Inf. Theory* **1983**, *29*, 551–559. [[CrossRef](#)]
41. Bernardini, F.; Bajaj, C. Sampling and Reconstructing Manifolds Using Alphashapes. 1997. Available online: <https://docs.lib.purdue.edu/cgi/viewcontent.cgi?article=2349&context=cstech> (accessed on 12 April 2023).
42. Edelsbrunner, H.; Mücke, E.P. Three-dimensional alpha shapes. *ACM Trans. Graph.* **1994**, *13*, 43–72. [[CrossRef](#)]
43. Arya, S.; Malamatos, T.; Mount, D.M. Space-time tradeoffs for approximate nearest neighbor searching. *J. ACM* **2009**, *57*, 1–54. [[CrossRef](#)]
44. Li, H.; Zhang, X.; Jaeger, M.; Constant, T. Segmentation of forest terrain laser scan data. In Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI '10), Association for Computing Machinery, New York, NY, USA, 12–13 December 2010; pp. 47–54. [[CrossRef](#)]
45. Yun, T.; Jiang, K.; Li, G.; Eichhorn, M.P.; Fan, J.; Liu, F.; Chen, B.; An, F.; Cao, L. Individual tree crown segmentation from airborne LiDAR data using a novel Gaussian filter and energy function minimization-based approach. *Remote Sens. Environ.* **2021**, *256*, 112307. [[CrossRef](#)]
46. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–10.

**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.