forests

MDPI

Article

An Individual Tree Detection and Segmentation Method from TLS and MLS Point Clouds Based on Improved Seed Points

Qiuji Chen [1,*], Hao Luo [1], Yan Cheng [2], Mimi Xie [1] and Dandan Nan [1]

[1] College of Geomatics, Xi'an University of Science and Technology, Xi'an 710054, China; luohao@stu.xust.edu.cn (H.L.); 22210061016@stu.xust.edu.cn (M.X.); 23210010003@stu.xust.edu.cn (D.N.)
[2] School of Earth Sciences, Chengdu University of Technology, Chengdu 610059, China; chengyan@stu.cdut.edu.cn
* Correspondence: qiujichen@stu.xust.edu.cn

Abstract: Individual Tree Detection and Segmentation (ITDS) is a key step in accurately extracting forest structural parameters from LiDAR (Light Detection and Ranging) data. However, most ITDS algorithms face challenges with over-segmentation, under-segmentation, and the omission of small trees in high-density forests. In this study, we developed a bottom–up framework for ITDS based on seed points. The proposed method is based on density-based spatial clustering of applications with noise (DBSCAN) to initially detect the trunks and filter the clusters by a set threshold. Then, the K-Nearest Neighbor (KNN) algorithm is used to reclassify the non-core clustered point cloud after threshold filtering. Furthermore, the Random Sample Consensus (RANSAC) cylinder fitting algorithm is used to correct the trunk detection results. Finally, we calculate the centroid of the trunk point clouds as seed points to achieve individual tree segmentation (ITS). In this paper, we use terrestrial laser scanning (TLS) data from natural forests in Germany and mobile laser scanning (MLS) data from planted forests in China to explore the effects of seed points on the accuracy of ITS methods; we then evaluate the efficiency of the method from three aspects: trunk detection, overall segmentation and small tree segmentation. We show the following: (1) the proposed method addresses the issues of missing segmentation and misrecognition of DBSCAN in trunk detection. Compared to using DBSCAN directly, recall (r), precision (p), and F-score (F) increased by 6.0%, 6.5%, and 0.07, respectively; (2) seed points significantly improved the accuracy of ITS methods; (3) the proposed ITDS framework achieved overall r, p, and F of 95.2%, 97.4%, and 0.96, respectively. This work demonstrates excellent accuracy in high-density forests and is able to accurately segment small trees under tall trees.

Keywords: individual tree detection and segmentation; trunk detection; point clouds; seed points; seeds-based segmentation

check for updates

Citation: Chen, Q.; Luo, H.; Cheng, Y.; Xie, M.; Nan, D. An Individual Tree Detection and Segmentation Method from TLS and MLS Point Clouds Based on Improved Seed Points. *Forests* **2024**, *15*, 1083. https://doi.org/10.3390/f15071083

Academic Editor: Marcello Vitale

Received: 10 May 2024
Revised: 6 June 2024
Accepted: 19 June 2024
Published: 22 June 2024

1. Introduction

Forests are an important part of the global ecosystem, with a variety of irreplaceable ecological functions, such as reducing the impact of wind and fixing sand, conserving water, and regulating the climate. However, traditional forest inventory methods often require a large investment of human and material resources, limiting their scalability and efficiency. Over the past two decades, LiDAR technology has played an increasingly important role in forestry resource surveys with its active remote sensing characteristics [1–3]. In contrast to traditional optical remote sensing algorithms, LiDAR-acquired point cloud data can obtain accurate tree structural parameters, such as tree height, diameter at breast height, crown diameter, volume, etc. [4,5], which are useful for calculating forest biomass [6] and extracting quantitative structural models (QSM) [7]. Accurate extraction of structural parameters of forest trees relies on accurate ITDS, which has become a core direction of forest LiDAR point cloud research in recent years [8,9]. Small-footprint LiDAR can be classified into three types based on the platform where the scanner is mounted: airborne LiDAR, terrestrial LiDAR (TLS), and mobile LiDAR (MLS). Airborne LiDAR has a relatively low point density and is limited in the amount of point data it can acquire from a single tree. Furthermore, the top–down data acquiring mode cannot easily capture detailed information on trunks, branches, and understory vegetation, which somewhat constrains its potential to be applied to single-tree studies [10,11]. In contrast, TLS and MLS use side-view scanning mode to collect data, which has a higher density of point cloud data per unit area and can obtain rich tree-side information. This is extremely useful for detailed studies at the level of a single tree and provides important technical support for high-precision tree observation and analysis [12–14].

ITDS methods for airborne LiDAR data can be broadly categorized into CHM-based (Canopy Height Model) and point-cloud-based methods [15,16]. The CHM-based method identifies the exact location of the tree by setting up a search window to locate the highest point in the canopy, and then a segmentation algorithm is used to delineate the contours of individual canopies [17,18]. In contrast, point cloud-based methods directly manipulate and analyze a large number of points in 3D space, such as using clustering algorithms to classify the point cloud based on distance for accurate tree identification and segmentation [19,20]. Both CHM-based and point-cloud-based methods are effective methods for LiDAR ITDS [21–23], though the accuracy of these two types of methods tends to drop significantly when dealing with complex forests with irregular canopies, overlapping crowns, or multi-stem characteristics [24–28].

Many methods exist for the ITDS of TLS and MLS data. Lu et al. [29] proposed a new method to extract tree trunks based on the intensity information of LiDAR point clouds and segmented trees based on the relative spacing of each point. However, the need for LiDAR intensity information and the shading conditions of the trees limit the further application of the approach. Tao et al. [30] proposed a tree crown segmentation method based on ecological theory. The method first applies the DBSCAN algorithm to identify the trunk by means of density-dependent spatial clustering and then divides the canopy based on the shortest path algorithm, and this method achieves high segmentation accuracy in different test data. Lee et al. [31] proposed an adaptive clustering method similar to the Watershed Algorithm for single-timber segmentation in LiDAR point clouds of pine forests. However, it relies on sufficient training datasets for supervised learning and the segmentation performance for structurally complex forest environments has not been fully validated. Xia et al. [32] proposed a new method to automatically detect tree trunks in dense forests using single-scan TLS data, which achieved high trunk detection accuracy. Zhong et al. [13] proposed a hierarchical segmentation technique for top–down processing of TLS and MLS point clouds. The method first splits a large point cloud region into smaller localized point clusters by spatial clustering method, and then these localized point clusters are further segmented to distinguish between the core region of the trees and the overlapping region of the edges. Comesana-cebral et al. [33] proposed a cylindrical volume cluster analysis method based on the iterative DBSCAN algorithm for segmenting single trees. However, a considerable number of misidentifications and missed segmentations still exist. Trochta et al. [34] proposed an algorithm that segments individual trees from bottom to top by clustering points in ground-parallel slices. The algorithm identifies tree bases as low clusters and merges adjacent clusters with stems to form complete tree structures. Burt et al. [35] proposed a segmentation technique called treeseg. Once tree stems were detected, they employed proportional relationships between stem diameter, tree height, and crown size to isolate tree crowns. Wu et al. [36] proposed a voxel-based labeled neighborhood search method that combines seed point selection and region growth algorithms for identifying street trees and obtaining their morphological features. This method achieves up to 98% tree detection accuracy in a test area where trees rarely overlap. Xing et al. [37] proposed a voxel-based layer-by-layer clustering method for segmenting single trees in TLS point cloud data. The method is based on sequential analysis of the z-values of the point cloud data in the vertical direction and layer-by-layer clustering using a fuzzy C-means algorithm to determine the specific location of each tree. This method utilizes the three-dimensional information of the point cloud data and achieves tree segmentation by layer-by-layer clustering. It is clear from these studies that the ITS method starting from the trunk portion of the tree is more appropriate for TLS and MLS point cloud data. In these ITDS methods, accurately recognizing the trunk and preventing the problems of missegmentation and misrecognition are critical to the accuracy of the final segmentation results.

In this study, we proposed a new framework for ITDS in TLS and MLS point clouds, aiming to improve overall accuracy and reduce the tendency to miss small trees in high-density forests. Our specific objectives are as follows: we aim (1) to solve the common problems of missegmentation and misrecognition when using DBSCAN to detect tree trunks, and to improve the accuracy of tree trunk detection; (2) to explore the best ITS strategy based on the seed points, and study the effect of seed points on the accuracy of ITS methods; and (3) to evaluate the effectiveness of the proposed methods in terms of trunk detection, overall segmentation, and small tree segmentation. We also analyze segmentation accuracy in high-density forests. The data used for this purpose are (1) MLS data for planted forests in China and (2) TLS data for natural forests in Germany.

2. Datasets

2.1. Planted Forest of MLS Point Clouds in China

As shown in Figure 1, the study area is located in a coal mine in the northwestern region of Shenmu City, Shaanxi Province, China. The primary landform types in this area include wind-sand plains, river-valley terraces, and loess hills and gullies. This is a planted forest with an area of about 8 ha, and the average tree density is 391 ha^{-1}. The main tree species in this forest are poplars. We selected three plots from the area as experimental data. For detailed information on these plots, please refer to Plot 1, Plot 2, and Plot 3 in Table 1.

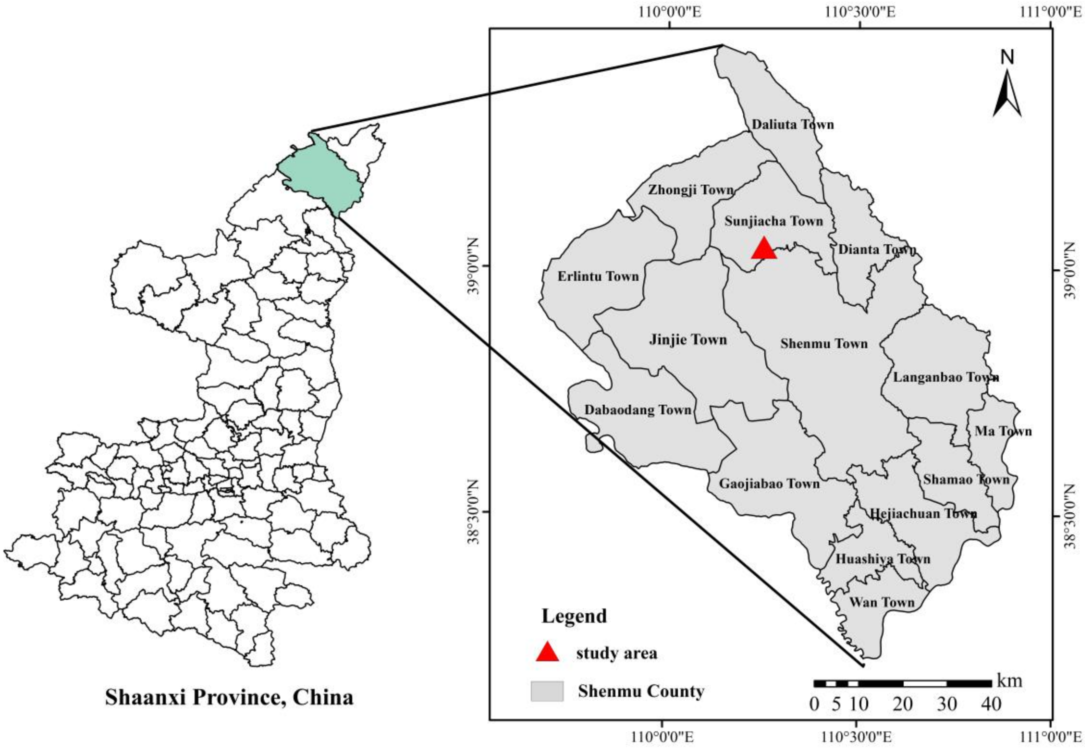


Figure 1. Location of the study area.

Table 1. Detailed information on the six experimental plots.

Forest Type	Plot ID	Tree Number	Area (m^2)	Stem Density (Plants/ha)	Tree Height (m)		
					Max	Min	Mean
Planted forest	Plot 1	49	1238.06	396	9.65	8.50	9.15
	Plot 2	40	1002.72	399	9.25	8.46	9.12
	Plot 3	50	1308.06	382	9.70	8.42	9.13
Natural forest	BR01	63	387.58	1625	19.78	7.36	17.92
	BR03	40	782.20	511	27.28	8.86	23.86
	BR05	74	1461.49	506	34.35	6.42	29.89

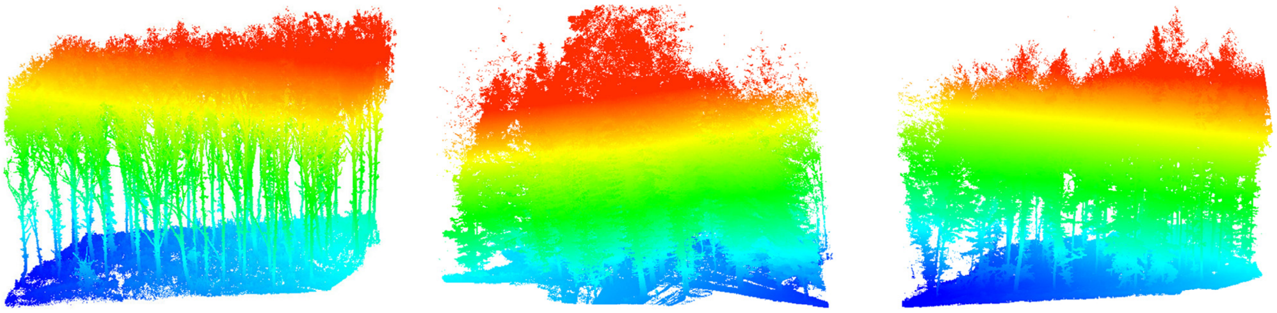
The MLS data were collected on 13 November 2019 using the RIEGL miniVUX-1UAV acquisition device. The data were collected on a clear day with minor cloud cover over higher elevations. To ensure the accuracy of MLS data collection, the collection routes were meticulously planned to minimize the effects of sharp turns and major obstacles in the data collection process. Given the sloping terrain of the study area, an eight-shaped path was chosen for data acquisition. Starting from the center of the experimental sample plots, the data collection proceeded along the eight-shaped route at a consistent pace, concluding at the center of the forest. The parameters of MLS data acquisition are detailed in Table 2.

Table 2. The parameters of MLS data acquisition.

Related Parameters	The Parameters Settings
Starting angle (°)	15
Straight angle (°)	345
Scanning frequency (kHz)	100
Density (pts/m^2)	564

2.2. Natural Forests of TLS Point Clouds in Germany

To effectively highlight the strengths of the proposed methods in small tree detection and high-density forest segmentation, as well as to assess the performance of ITS methods across various types of plots, we also used the TLS point cloud data provided by Weiser et al. [38]. The data are located in the mixed Central European forest lands of Bretten and Karlsruhe near Baden-Württemberg, Germany. The TLS point cloud data were collected by mounting a RIEGL VZ-400 unit on a tripod, which scanned selected tree areas using the multi-scan approach with five to eight scanning positions. The device offers a measurement accuracy of 5 mm and a point cloud density of 7000 points per cubic meter at a scanning range of 100 m. The sensor operates with a pulse repetition frequency of 300 kHz and an angular step width of 0.017 degrees in both vertical and horizontal directions. At some positions, an additional scan was performed using a tilt mount to capture the top of the trees at proximity. Figure 2a shows the TLS point cloud data of Plot BR01 collected on 3 July 2019. Figure 2b shows the TLS point cloud data of Plot BR03 collected on 4 June 2019. Figure 2c shows the TLS point cloud data of Plot BR05 collected on 17 July 2019. Detailed information on the three experimental plots is presented in Table 1.



(**a**) (**b**) (**c**)

Figure 2. Terrestrial laser scanning (TLS) data. (**a**) Plot BR01, (**b**) Plot BR03, (**c**) Plot BR05. (Different colors in this figure indicate different height levels, with the yellow line specifically representing the height-to-live crown).

3. Methodology

Figure 3 shows the flowchart of the proposed framework. It contains three main steps: data preprocessing, trunk detection, and ITS. First, the raw point cloud data were preprocessed to generate a normalized non-ground point cloud, and the spatial elevation section method was used to obtain the specific height layer slices of the trunk. Then, the KNN algorithm was used to solve the missing segmentation and misrecognition problems that occur when using DBSCAN to detect tree trunks, and the RANSAC cylinder fitting algorithm was utilized to correct the results of trunk detection. Finally, we computed the centroid of the detected trunk point cloud as the seed point. Three seed-based ITS methods (Seeds + Dalponte-2016, Seeds + Li-2012, Seeds + Tao-2015) were compared with the direct application of these three ITS methods [30,39,40]. These are described in more detail below.

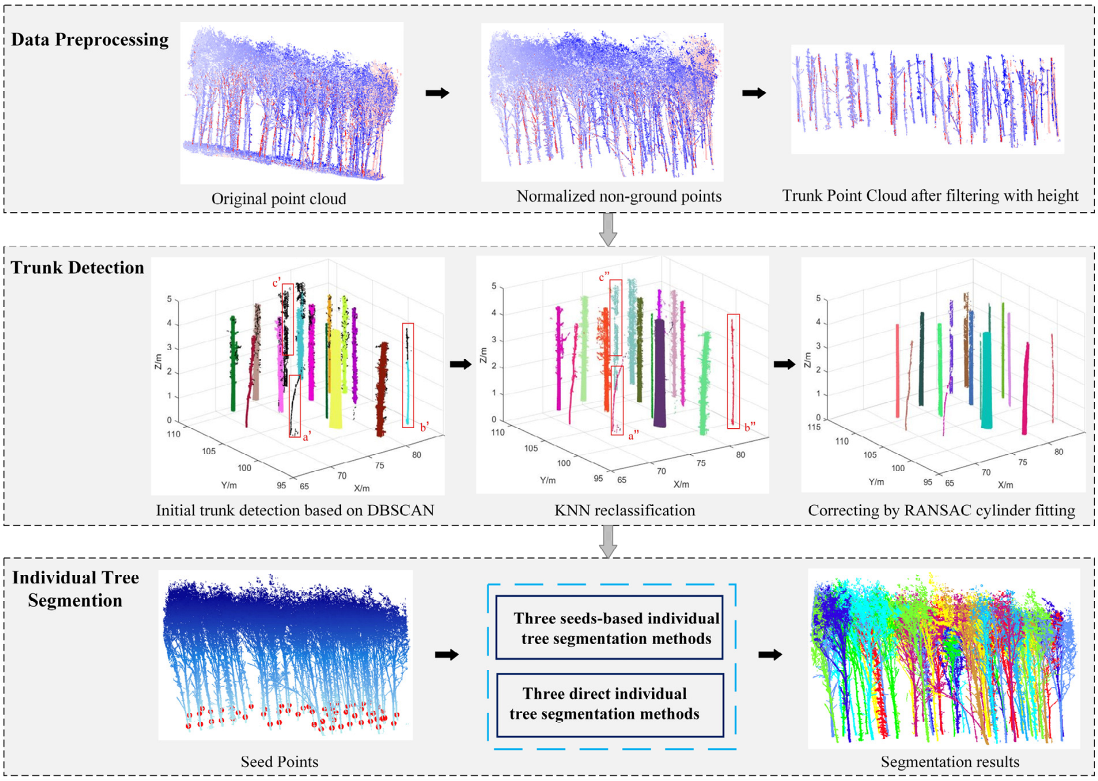

Data Preprocessing
Original point cloud
Normalized non-ground points
Trunk Point Cloud after filtering with height
Trunk Detection
c'
a'
b'
Z/m
Y/m
X/m
Initial trunk detection based on DBSCAN
c''
a''
b''
KNN reclassification
Correcting by RANSAC cylinder fitting
Individual Tree Segmention
Seed Points
Three seeds-based individual tree segmentation methods
Three direct individual tree segmentation methods
Segmentation results

Figure 3. Flowchart of the proposed framework. The a′, b′, and c′ represent the error cases encountered when using the DBSCAN algorithm for trunk detection, while the a″, b″, and c″ denote the corresponding error correction results obtained using the KNN algorithm.

3.1. Data Preprocessing

To enhance the accuracy and efficiency of the ITDS method, this study applied a series of preprocessing operations to the raw point cloud data. First, the random sampling algorithm was employed to downsample the data, reducing the number of point clouds to decrease the complexity of the segmentation algorithm. The algorithm maintains the initial spatial distribution of the point cloud and ensures each point has an equal probability of being sampled within a set threshold range. This achieves a uniform reduction in the data volume. Then, the point cloud was denoised using the statistical outlier removal (SOR) filter algorithm to remove noise. Next, the cloth simulation filtering (CSF) algorithm [41] was employed to segment ground and non-ground point clouds. The CSF method leverages the intrinsic properties of the textile material to enhance the point cloud filtering process by adjusting the simulated physical properties associated with the fabric. Finally, height normalization was performed on the point cloud data. The simulated fabric generated by the CSF algorithm was employed as the Digital Terrain Model (DTM), and the normalized point cloud height was derived by subtracting the DTM height from the original point cloud height (Figure 4). This transformation process ensures that the height values of the point cloud accurately reflect changes in height relative to the ground. It also maintains a consistent spatial distribution of trees and their branches, preserving their precise positional relationships within the point cloud data. The preprocessing of the aforementioned point cloud was conducted using the open-source software CloudCompare (https://www.cloudcompare.org/, accessed on 5 October 2022). In this process, random sampling was configured to retain either one-half of the original number of points. The SOR filtering parameters, including the mean distance estimation points and the standard deviation multiplier threshold, were set to their default values. For CSF filtering, the fabric resolution was specified as 0.4, with the maximum number of iterations set to 800.

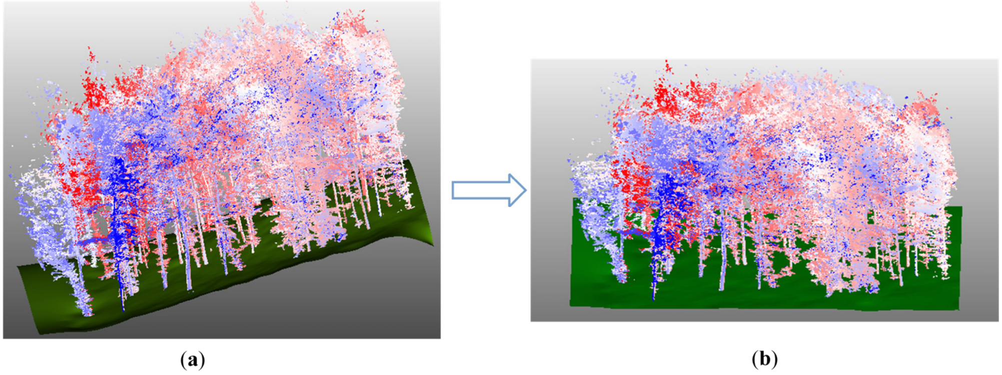


Figure 4. Normalization of the height. (**a**) The simulated fabric generated by the CSF algorithm as the Digital Terrain Model (DTM), (**b**) the normalized point cloud height was obtained by subtracting the DTM height from the original point cloud height.

3.2. Trunk Detection

There are three main steps for trunk detection: (1) initial trunk detection based on DBSCAN and filtering of correctly clustered trunks with set thresholds; (2) the KNN algorithm is used to reclassify the non-core clustered point cloud after threshold filtering; and (3) the RANSAC cylinder fitting algorithm is applied to correct the trunk detection results by identifying and removing outliers or incorrect matches, thus improving the accuracy of the detection.

3.2.1. Initial Trunk Detection Based on DBSCAN

DBSCAN is a classic density-based spatial clustering algorithm that determines the clustering results based on the tightness of sample distribution [42]. The algorithm operates by initially selecting a random core point and then recursively grouping points that meet the density criterion into the same cluster, eventually forming a maximal region encompassing both core and boundary points. DBSCAN recognizes arbitrarily shaped classes, allows clustering of different sizes for each class, and does not require the number of clusters to be defined. Neighborhood radius (Eps) and the minimum number of points (MinPts) are the criteria used by the DBSCAN algorithm to describe density. Eps determines the extent of the neighborhood of sample points, while MinPts sets the minimum number of samples needed within that neighborhood, and these are used to determine whether a point is a core point. DBSCAN determines the type of sample points based on their density, and when the number of sample points within a sample point's neighborhood radius Eps is greater than or equal to MinPts, these points are classified as core points. The points that do not satisfy the core point condition but are in the neighborhood of a core point are classified as boundary points. The points that are neither core nor boundary points are considered as noise points.

Due to the potential for over-segmentation in DBSCAN during the detection of tree trunks, a single tree trunk may be erroneously identified as multiple distinct clusters. In addition, factors such as lower vegetation, leaves, and branches interfere and can easily be misidentified as trunks during trunk detection, leading to inaccurate clustering results. Therefore, we computed three features of the trunks—the number of points in a cluster (N), the height of the clusters (H), and the angle between the main direction of the clusters and the vertical direction (T)—to evaluate the right cluster for the individual trunk. Feature H is the difference between the maximum and minimum values of a single cluster height. Feature T describes the relationship of data variation in different dimensions by calculating the covariance matrix of each cluster. Then, eigenvalue decomposition is performed to obtain the eigenvalues and eigenvectors of the covariance matrix. The eigenvector with the largest eigenvalue is selected as the main direction of the trunk. Finally, the angle between the trunk direction and the vertical direction is calculated. We set the condition that the core cluster would be selected as the right trunk from the DBSCAN clusters (Equation (1)). These constraints effectively eliminate interference from lower-level vegetation, leaves, and branches, and filter out incorrectly clustered trunks, thereby improving the accuracy of trunk detection.

$$\mathrm{Trunk} = \begin{cases} \mathrm{Ni} > 0.001\ \mathrm{N} \\ \mathrm{Hi} > 2 \\ \mathrm{Ti} < 40 \end{cases} \tag{1}$$

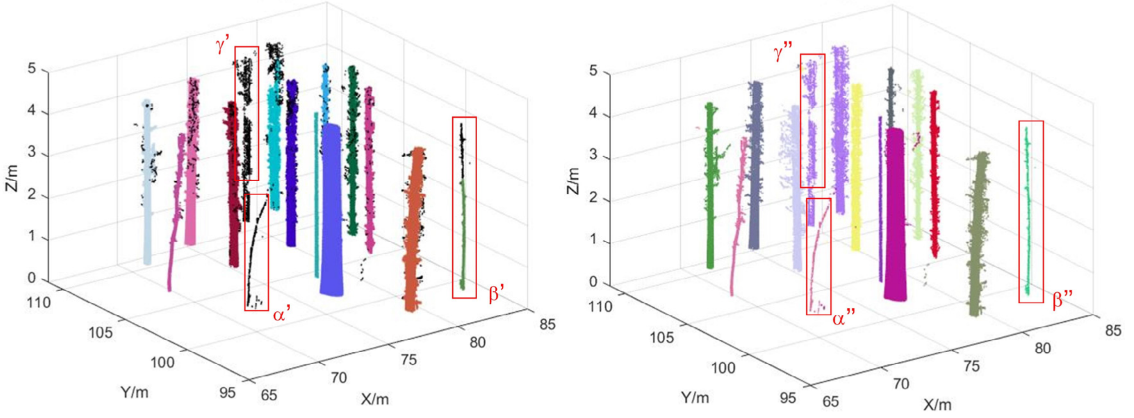
where Trunk is each core clustering in DBSCAN detection of trunks, Ni is the number of points of cluster i, Hi is the height of cluster i in meters, Ti represents the angle between the main direction of a single cluster and the vertical direction in degrees, N is the total number of points in all trunk point clouds.

3.2.2. Reclassification of Non-Core Cluster Point Clouds Using KNN Algorithm

The KNN algorithm, as a classical supervised learning algorithm, has demonstrated its simplicity, effectiveness, and intuition in classification tasks and has been widely used in several fields [43,44]. The core idea of the algorithm is to classify the training samples based on the k training samples with the closest distance to the sample to be tested by comparing the features in the training set and their corresponding labels. Ultimately, based on the labels of these neighbors, the principle of majority voting is used to determine the category affiliation of the object to be classified.

Due to the shading effect of understory shadows and the obstruction of lower-level vegetation, the point cloud data of small tree trunks typically exhibit low-density characteristics. At the same time, the point cloud of the trunk far from the center of the scanning device may be relatively sparse, and there may even be discontinuities in the point cloud of the trunk surface. Moreover, the clustering outcomes of DBSCAN are significantly influenced by the compactness of the sample distribution, posing a challenge in the selection of Eps and MinPts parameters. Especially in complex forest environments, determining the optimal parameters becomes more difficult, and inappropriate parameter settings can greatly reduce the accuracy of trunk detection. In Figure 5a, the small tree labeled as α'

was over-segmented into multiple stem segments due to insufficient point cloud density and was subsequently eliminated during the point-cluster filtering stage. The small tree labeled as β′ was over-segmented due to the low density of the top trunk point cloud. For certain tree trunks whose surface point clouds may exhibit discontinuities due to understory shading and occlusion, DBSCAN may erroneously segment them into multiple stem segments, leading to their removal during the point-cluster filtering stage and consequently resulting in the omission of these trunks (γ′). On the other hand, if these clusters of points segmented into multiple stem segments are not filtered out during the threshold screening stage, then this in turn increases the over-segmentation error when detecting trunks.



(**a**) (**b**)

Figure 5. Trunk detection results, colored for core trunks and black for other points. (**a**) DBSCAN detection results after threshold filtering and (**b**) results after reclassification of the non-core clustered point cloud using the KNN algorithm.

To solve the above mentioned problem, the KNN algorithm is used to reclassify the non-core clustered point cloud after threshold filtering. In this process, the core trunk point clouds identified from the DBSCAN clustering results, which were filtered using the threshold defined by Equation (1), are used as training data. Subsequently, a distance-weighted method is employed to reclassify the non-core trunk point clouds, thereby optimizing the classification results. The results of KNN algorithm reclassification are shown in Figure 5b, it can be seen that the small tree (α') that was missed in Figure 5a has been reclaimed and correctly identified into the core clusters (α''). Meanwhile, in the clustering filtering stage, the point cloud at the top of the trunk (β′), which was originally mistakenly excluded due to over-segmentation, is correctly reattributed to that trunk (β″). For the trunks in Figure 5a where the point cloud is discontinuous and removed in the point cluster filtering stage (γ'), the KNN algorithm reclassifies the whole trunks into the core clusters (γ''). Using this method, the issue of small and edge trunks being easily missed and over-segmented is effectively resolved, resulting in a significant improvement in the accuracy of trunk detection.

For edge trunks located far from the scanning center, when only half of the trunk point cloud is captured during scanning (with the point cloud on the other side of the trunk having a lower density), this can directly result in some of the trunk points being incorrectly identified as noise by DBSCAN (Figure 6a). The KNN algorithm can effectively reclassify the trunk points that were incorrectly recognized as noise to their correct trunks, with the results after reclassification shown in Figure 6b. This step not only enhances the completeness of trunk detection but also establishes a more comprehensive and accurate data foundation for subsequent RANSAC cylinder fitting. The accuracy of RANSAC cylinder fitting is highly dependent on the completeness and precision of the input data, and the processing by the KNN algorithm can effectively prevent the issue of difficulty in fitting RANSAC cylinders and the degradation of accuracy due to insufficient trunk points.

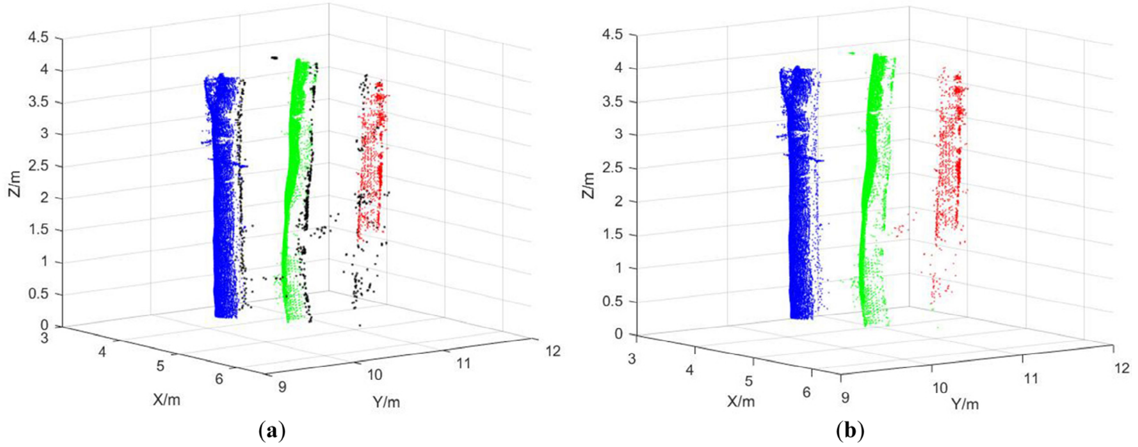


Figure 6. Trunk detection results under insufficient point cloud scanning, colored for core trunks, black for other points. (**a**) Detecting results using DBSCAN and (**b**) results after reclassification of the non-core clustered point cloud using the KNN algorithm.

3.2.3. Correcting Trunk Detection Results Based on RANSAC Cylinder Fitting

Due to the presence of low-lying vegetation in the understory and numerous noise points around the tree trunks, this study utilized the RANSAC cylinder fitting algorithm to refine trunk detection outcomes, excluding non-trunk point clouds [45]. The RANSAC algorithm is an iterative algorithm that is primarily used to estimate geometric models in a dataset. The basic principle of the algorithm is to randomly select a set of points from the dataset and calculate the parameters of the geometric model based on this set of points. The algorithm then repeats this process several times, with the goal of finding the best model with the largest number of interior points. In each iteration, the RANSAC algorithm uses the selected point set to compute a score indicating how many points in the entire dataset match this point set. Points that match the selected point set are considered interior points. To determine whether each point is an interior point, the algorithm calculates the distance from each point to the model and compares it to a preset threshold. The RANSAC algorithm chooses the point set with the most supported interior points as the most likely model, and the final estimate is obtained based on a least-squares adjustment using all these interior points. The algorithm exhibits strong robustness in estimating geometric models, obtaining good results even in the presence of noise and outliers. Its iterative nature allows for multiple attempts, which enhances its resistance to outliers in the data.

3.3. ITS Methods

To demonstrate the advantages of ITS based on seed points obtained through the proposed method, we explore the effects of different seed point selection techniques on the accuracy of ITS methods and evaluate the performance of commonly used ITS algorithms in different forest environments. In this study, the three seed-point-based ITS methods (i.e., the seed-point-based Dalponte-2016 method: Seeds + Dalponte-2016, the seed-point-based Li-2012 method: Seeds + Li-2012, and the seed-point-based Tao-2015 method: Seeds + Tao-2015) are utilized for segmenting trees. The Tao-2015 and Li-2012 methods were implemented using LiDAR360 software (https://www.lidar360.com/, accessed on 13 February 2022). The Dalponte-2016 method was implemented using the lidR and lidRplugins packages in R software (https://cran.r-project.org/, accessed on 31 October 2023) [46].

The Dalponte-2016 method is a top–bottom segmentation strategy, which is based on the Canopy Height Model (CHM) and applies a region-growth algorithm for accurate tree segmentation [39]. Specifically, the method first searches for local maxima in the rasterized CHM, which represent the locations of the tree tops. Subsequently, from each of the identified local maxima, a complete canopy structure grows around these local maxima.

The Li-2012 method is a regional growth method, combined with the threshold judgment, based on point cloud data rather than CHM raster [40]. The core idea of the method is to identify the highest point in the point cloud as the tree top, and then use the interval threshold rule to determine whether the points below the tree top belong to the tree starting from the tree top. After completing the classification of all points below the top of the tree, the algorithm removes the segmented tree and searches for a new highest point in the remaining point cloud, continuing the iterative segmentation until all trees are successfully recognized. The key to this algorithm is the selection of the interval threshold, which directly affects the accuracy of the segmentation. In sparse forests, larger thresholds can be used to separate trees; while in dense forests, smaller thresholds need to be chosen to avoid under-segmentation.

The Tao-2015 method is the comparative shortest path (CSP) algorithm developed by Tao et al. [30]. The method is based on the 3D spatial structure of the point cloud data and builds a continuous topology starting from the bottom of the tree point cloud data upwards until reaching the top of the tree. First, this method uses DBSCAN to automatically identify tree trunks within the sample plot. Next, it selects a 10 cm thick trunk point cloud slice, extracted from the vertical ground level at 1.25 m to 1.35 m from the tree base, to measure the diameter at breast height, which serves as the seed point for the tree. Finally, the CSP algorithm, based on tree seed points and metabolic ecology theory, is applied to identify the dendritic portion of the tree and complete the segmentation of the canopy point cloud data.

3.4. Accuracy Assessment

To evaluate the performance and sensitivity of the proposed ITDS strategy in this study, the ITDS results were analyzed in comparison with real trees, with an emphasis on counting the number of true positives (TP), false negatives (FN), and false positives (FP). The accuracy of the ITDS method is evaluated using recall (r), precision (p), and F-score (F) indices on the basis of the TP, FN, and FP. The calculation formula is as follows:

$$\mathrm{r} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

$$\mathrm{p} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{3}$$

$$\mathrm{F} = 2 \times \frac{\mathrm{r} \times \mathrm{p}}{\mathrm{r} + \mathrm{p}} \tag{4}$$

where TP is the number of trees that were correctly segmented, FN is the number of neighboring trees that were assigned incorrect segmentation, and FP is the trees that were segmented but did not actually exist. "r" represents the proportion of ground reference trees that were successfully detected, reflecting the recall of the results. "p" indicates the proportion of detected trees that accurately correspond to ground reference trees, representing the precision of the results. The F, serving as an index of overall accuracy, is derived from the values of r and p.

4. Results and Analysis

4.1. Trunk Detection Results

Figure 7 demonstrates the results of trunk detection and its accuracy for the six plots of the proposed method. TP indicates that trunks were correctly detected; FN indicates that multiple trunks were actually present, but only one was identified; FP indicates that they were not actually present, but incorrectly identified as trunks. "r" is the trunk detection rate, which is the ratio of detected to real trunks; "p" is the correct rate of detected trunks; and F is the weighted mean calculated by combining r and p. In the trunk detection across six plots, r ranged from 94.6% to 98.0%, with a total r of 95.9%; p ranged from 95.0% to 100%, with a total p of 97.4%; and F ranged from 0.95 to 0.99, with a total F of 0.97. The r, p, and F values for the three plots selected from Chinese planted forests (Plot1, Plot2, and Plot3) were higher than those for the three sample plots selected from German natural forests (BR01, BR03, and BR05), indicating that planted forests have better r, p, and F values than natural forests. Among the six plots, the total FN for the three sample plots selected from the German natural forest (BR01, BR03, and BR05) was nine, while the total FN for the three sample plots selected from the Chinese planted forests (Plot1, Plot2, and Plot3) was three. In addition, it is noteworthy that the total number of FP is lower than the number of FN.

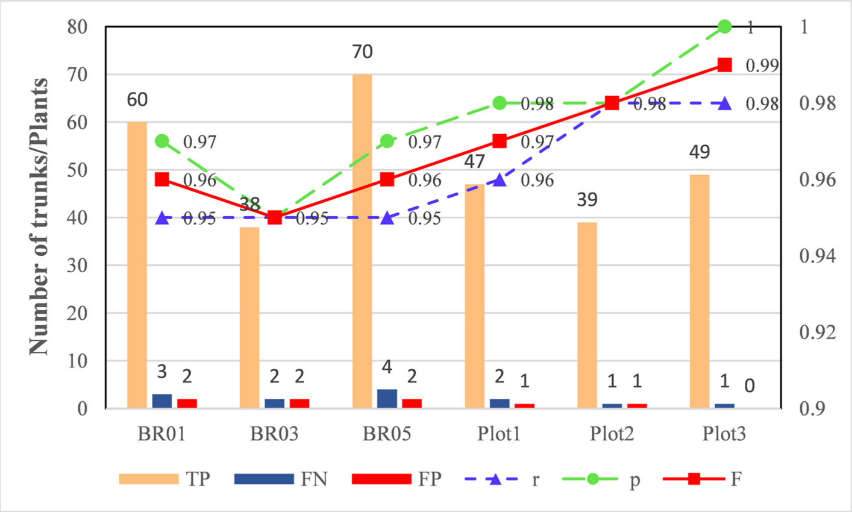


Figure 7. Trunk detection results and accuracy in six plots.

To better demonstrate the advantages of the trunk detection method presented in this paper, a comparison was made with the direct use of DBSCAN for trunk detection (Table 3). When directly using DBSCAN for trunk detection, the total TP was 280, indicating that 280 out of 316 trunks were correctly detected. The r, p, and F were 89.9%, 90.9%, and 0.90, respectively. The trunk detection method proposed in this paper correctly identified a total of 303 trunks, with an overall r, p, and F of 95.9%, 97.4%, and 0.97, respectively. The proposed trunk detection method improved the total r, p, and F by 6.0%, 6.5%, and 0.07, respectively, compared to the direct use of DBSCAN for trunk detection.

Table 3. Comparison with DBSCAN in trunk detection results and accuracy.

Plot	Actual Number	DBSCAN				Proposed			
		TP	r	p	F	TP	r	p	F
BR01	63	56	88.9%	90.3%	0.90	60	95.2%	96.8%	0.96
BR03	40	34	89.5%	87.2%	0.88	38	95.0%	95.0%	0.95
BR05	74	64	88.9%	90.1%	0.89	70	94.6%	97.2%	0.96
Total natural forest	177	154	89.0%	89.2%	0.89	168	94.9%	96.6%	0.96
Plot1	49	45	91.8%	91.8%	0.92	47	95.9%	97.9%	0.97
Plot2	40	36	90.0%	92.3%	0.91	39	97.5%	97.5%	0.98
Plot3	50	45	90.0%	93.8%	0.92	49	98.0%	100%	0.99
Total planted forest	139	126	90.6%	92.6%	0.92	135	97.1%	98.5%	0.98
Total	316	280	89.9%	90.9%	0.90	303	95.9%	97.4%	0.97

4.2. ITS Results

The overall segmentation accuracy of the three seed-point-based ITS methods (Seeds + Dalponte-2016, Seeds + Li-2012, Seeds + Tao-2015) was significantly better than that of directly using these three methods (Figure 8 and Table 4). All three seed-point-based ITS methods had stable F scores on the six plots (Figure 8), with the highest segmentation accuracy being achieved by the method of Seeds + Tao-2015, which correctly segmented

298 out of 316 trees, with total r, p, and F of 95.2%, 97.4% and 0.96, respectively (Table 4). In contrast, the three ITDS methods without seed points had lower overall accuracy (Figure 8 and Table 4). Compared to Dalponte-2016 and Li-2012, the Tao-2015 method performed slightly better, correctly segmenting 261 out of 316 trees, with an overall r, p, and F of 86.7%, 78.8%, and 0.83, respectively (Table 4).

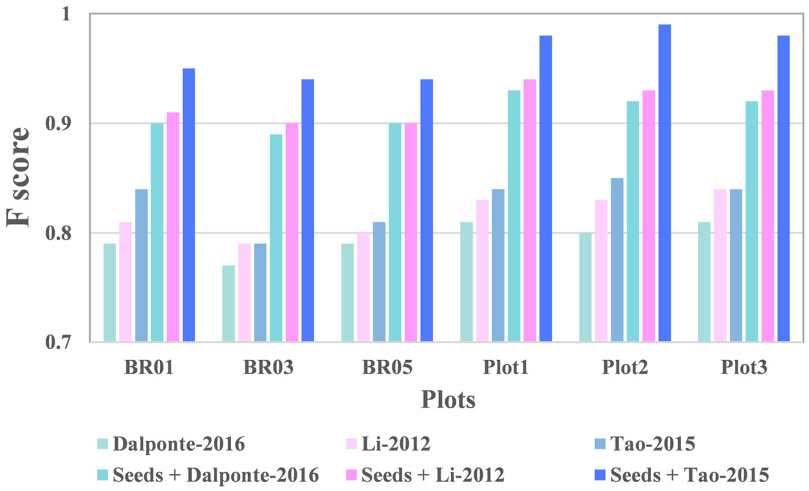


Figure 8. The accuracy tested by F score for all six individual tree segmentation (ITS) methods [30,39,40].

Table 4. The summarized accuracy for all six individual tree segmentation (ITS) methods.

Methods	Actual Number	Segmentation Number	TP	FN	FP	r	p	F
Dalponte2016 [39]	316	335	243	62	57	79.7%	81.0%	0.80
Li2012 [40]	316	350	249	52	63	82.7%	79.8%	0.81
Tao2015 [30]	316	364	261	40	70	86.7%	78.8%	0.83
Seeds + Dalponte2016	316	303	278	28	25	90.8%	91.7%	0.91
Seeds + Li2012	316	303	282	25	22	91.9%	92.8%	0.92
Seeds + Tao2015	316	303	298	15	8	95.2%	97.4%	0.96

4.3. Small Tree Detection Results

We define small trees as those that cannot be detected from the top-view point cloud data due to shading from the canopy layer. Small trees are usually in the lower vertical layers of the forest structure and are covered by the upper canopy layer. Among the total of 316 trees in the six plots, there are 46 small trees. The methods Seeds + Dalponte-2016, Seeds + Li-2012, and Seeds + Tao-2015 correctly identified and segmented 37, 39, and 42 small trees, respectively, with the correct segmentation rates for small trees being 0.80, 0.85, and 0.91. However, the methods Dalponte-2016, Li-2012, and Tao-2015 correctly recognized and segmented 28, 33, and 36 small trees, respectively, with correct small tree segmentation rates of 0.61, 0.72, and 0.78, respectively. Among the six ITDS methods, the Seeds + Tao-2015 method achieved the highest correct segmentation rate for small trees, reaching 0.91, while the Dalponte-2016 method had the lowest correct segmentation rate for small trees, at 0.61. Figure 9 shows the results of segmentation using the Seeds + Tao-2015 method for 23 trees in the BR05 plot, with the green small trees detected within the red box.

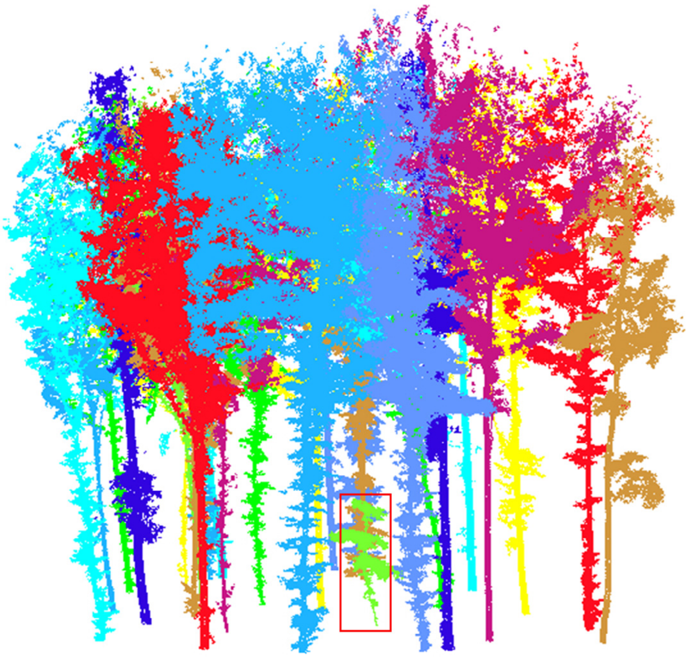


Figure 9. Segmentation results of small trees under tall trees, the red box marks the small green tree that was detected.

5. Discussion

5.1. Parameter Sensitivity Analysis

It is known that the performance of the DBSCAN clustering process is affected by the selection of appropriate input parameters (i.e., Eps and MinPts). However, the framework proposed in this paper greatly reduces the sensitivity to parameters when using DBSCAN to detect tree trunks. When Eps is set too small or Minpts is set too large, it can easily lead to over-segmentation of the trunk. However, the over-segmented trunk point cloud can be reclassified to the trunk by the KNN classification algorithm, which reduces the over-segmentation error of the trunk. When Eps is set too large or Minpts is set too small, it can easily lead to under-segmentation of the trunk. For under-segmented trunks, the RANSAC cylinder fitting algorithm can further segment the trunks, thus reducing the under-segmentation error. In order to improve the ability of the proposed framework to detect tree trunks, multiple experiments were conducted on different plots, and the appropriate range of values for Eps was determined to be between 0.1 and 0.7. Considering the segmentation accuracy of tree trunks in different plots, the value of Eps was chosen to be 0.2. Although other values can also satisfy the detection of tree trunks, it is generally recommended to choose a smaller coefficient, because even if part of the tree trunk is over-segmented, KNN can reclassify it onto that tree trunk. Larger coefficients may reduce the initial detection rate of tree trunks, thereby increasing the probability of under-segmentation errors. When KNN is used to reclassify the non-core clusters after threshold screening, k values chosen from 1 to 300 can achieve good results, but when k values are obtained too large, the possibility of misclassification is increased. We observed that when the k value is small (less than 20), the value of k seems to have little effect on the final classification results. Therefore, for consistency and practicality, the default value of k was chosen to be 5.

5.2. ITS Methods with or without Seed Points

Among the three direct ITS methods (Dalponte-2016, Li-2012, Tao-2015), Dalponte-2016 has the lowest accuracy. This is mainly because this method searches for local maxima in the rasterized CHM as tree tops, and then grows complete crown structures from the tree tops. However, accurately identifying tree tops in dense forests is quite difficult, and it is also challenging to segment small trees. Moreover, the spatial resolution of the CHM and the degree of smoothing of the CHM affect the tree apex extraction and ITS results [47]. Li-2012 directly takes the highest point in the point cloud as the tree top and combines an interval threshold rule to complete ITS. Although the accuracy of ITS with this method is not affected by the CHM as much as the Dalponte-2016 method, the accuracy of identifying tree tops as the starting point for segmentation in dense forests is very low. The method sets too large an interval threshold that may lead to under-segmentation errors, while too low a threshold tends to trigger over-segmentation errors; moreover, the method tends to have difficulty in recognizing small trees underneath tall trees. Tao-2015 uses the DBSCAN algorithm to automatically identify tree trunks in sample plots and selects 10 cm-thick slices of the trunk point cloud to extract the tree diameter at breast height, which serves as the seed point of the trees. However, the accuracy of selecting seed points significantly decreases due to interference from understory vegetation, lower branches, and leaves, making the method prone to over-segmentation errors.

Among the three seed-point-based ITDS methods, Seeds + Tao-2015 has the highest accuracy. The error mainly comes from the accuracy of obtaining seed points according to the proposed method. Therefore, the ITS accuracy of Seeds + Tao-2015 will be further improved when there are new methods for individual tree detection with higher accuracy than the proposed method. Seeds + Dalpnote-2016 may have errors during CHM interpolation and may have inaccurate crown edge segmentation. Although the Seeds + Li-2012 method found more accurate seed points using the proposed approach, due to its top–down strategy and classification of point cloud data through interval thresholds, in high-canopy-density forest environments, small trees and the edges of tree crowns may be erroneously segmented into other trees.

Compared to directly using these three ITS methods, the segmentation accuracy of the three ITS methods based on seed points detected by the proposed methods was significantly improved (Figure 8, Table 4). As the number of selected seed points is equal to the number of ITS results when the number of seed points selected is greater than the actual number of trees, it is easy to cause over-segmentation errors; when the number of seed points selected is less than the actual number of trees, it is likely to lead to under-segmentation errors. The proposed method achieves high accuracy in obtaining seed points through trunk detection, significantly reduces the over-segmentation and under-segmentation errors of ITDS methods, and improves the overall segmentation accuracy.

5.3. Analysis of Small Tree Detection Results

In the detection of a total of 46 small trees in six plots, the correct segmentation rates of the Dalponte-2016, Li-2012, and Tao-2015 methods for small trees were 0.61, 0.72, and 0.78, respectively. However, after combining the seed points obtained by the proposed methods, the correct segmentation rate of small trees was improved to 0.80, 0.85, and 0.91 for Seeds + Dalponte-2016, Seeds + Li-2012, and Seeds + Tao-2015, respectively. It is clear that the three seed-point-based ITDS methods are significantly better than the direct use of these three ITS methods in terms of the correct segmentation rate of small trees. Dalponte-2016 and Li-2012 identify the tree top as the starting point. However, due to the high density of the tree crown, these methods often have difficulty capturing the tops of small trees, making them particularly prone to missing small trees. Tao-2015 uses the DBSCAN algorithm to automatically identify a 10-centimeter-thick trunk point cloud slice to extract seed points. However, due to the interference of understory vegetation and non-trunk point clouds, this method is prone to missing or over-segmenting small trees. The proposed trunk detection method effectively recognizes the trunks of small trees and accurately finds the seed points of small trees. Therefore, the proposed seed-point-based ITS method greatly improves the detection accuracy of small trees under tall trees.

5.4. Analysis of ITS Results in High-Canopy-Density Forests

Figure 10 illustrates the ITS results using the Seeds + Tao-2015 method in the BR03 plot. In the BR03 plot, the F-scores of ITS using the Dalponte-2016, Li-2012, and Tao-2015 methods were 0.77, 0.79, and 0.79, respectively, while the F-scores using the Seeds + Dalponte-2016, Seeds + Li-2012, and Seeds + Tao-2015 methods were 0.89, 0.90, and 0.94, respectively (Figure 8). This indicates that in high-canopy-density forest environments, the performance of the three direct ITS methods is suboptimal, with numerous instances of over-segmentation and under-segmentation errors, while the ITS methods based on seed points obtained by the proposed method exhibit superior segmentation accuracy. The seed-point-based ITS method has a great potential for development in high-canopy-density forests. Since the number of segmentation results is equal to the number of seed points, when the accuracy of seed points is high enough, the accuracy of single-wood segmentation will be increased accordingly, which will significantly reduce the over-segmentation and under-segmentation errors that often occur in high-density forests. Therefore, exploring methods to obtain high-precision seed points in complex forest environments to achieve ITS will become an important direction for future research.

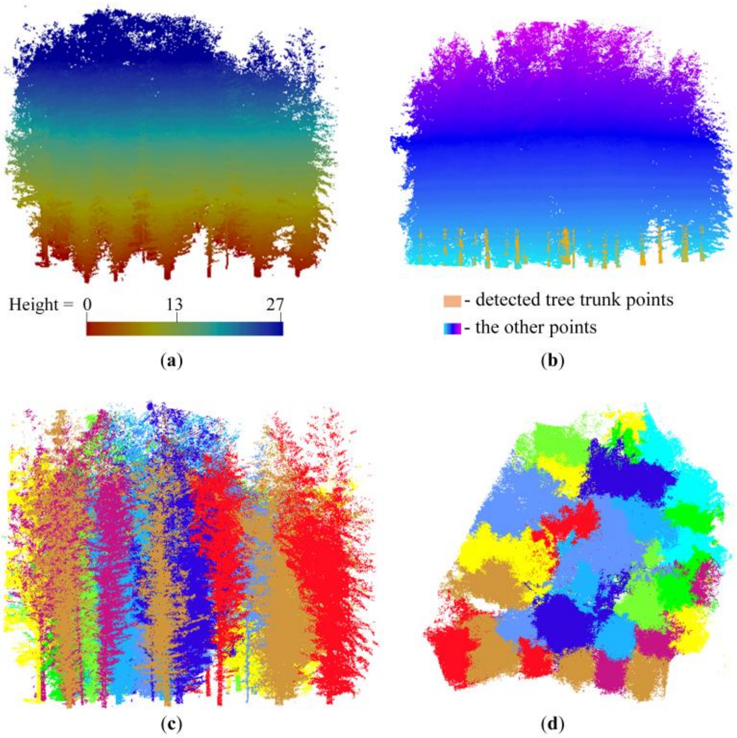


Figure 10. Individual tree segmentation (ITS) results in high-canopy-density forests. (**a**) The Original point clouds of BR03 plot, (**b**) trunk detection results, (**c**) side view of ITS results, where detected trees are randomly colored and the same color represents one tree, (**d**) top view of ITS results.

6. Conclusions

This study proposed a new framework for bottom–up ITDS based on seed points. The proposed framework used the KNN algorithm to solve the missing segmentation and misrecognition problems that occur when using DBSCAN to detect tree trunks, and the RANSAC cylinder fitting algorithm was utilized to correct the results of trunk detection. It improves the accuracy of trunk detection and consequently the accuracy of seed points. We used Chinese planted forest data and German natural forest data to investigate the effect of seed points on the accuracy of ITS methods and to evaluate the effectiveness of the proposed method. The results showed the following.

(1) This study addressed the issues of missing segmentation and misrecognition encountered when using DBSCAN for trunk detection, reducing the sensitivity of DBSCAN parameters and enhancing the accuracy of trunk detection. Compared to the direct use of DBSCAN, the r, p, and F for trunk detection were improved by 6.0%, 6.5%, and 0.07, respectively. (2) Compared to the direct application of these three ITS methods (Dalponte-2016: F = 0.80, Li-2012: F = 0.81, Tao-2015: F = 0.83), the three ITS methods based on the seed

points obtained from the proposed method (Seeds + Dalponte-2016: F = 0.91, Seeds + Li-2012: F = 0.92, Seeds + Tao-2015: F = 0.96) showed more stable segmentation results and better performance in all plots. (3) Among the six ITS methods, Seeds + Tao-2015 achieved the highest overall segmentation accuracy. This method can be considered as the preferred option for ITDS from TLS and MLS data, with excellent segmentation performance in both artifical and mixed forest experimental plots. (4) In terms of segmentation in high-canopy-density forests and detection of small trees, the ITS methods based on the seed points obtained from this study significantly outperformed other ITS algorithms.

The proposed method achieves high precision in seed point detection, significantly reducing issues such as under-segmentation, over-segmentation, and the omission of small trees that are common in high-canopy-density forests. This lays a solid foundation for precise calculation of tree structural parameters, offering robust datasets for biomass estimation and forest inventory.

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

Funding: This research was funded by the Shaanxi Geology and Mining Group Co. research special fund program, grant number KY202306.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to the consideration of data security.

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

References

1. Hakula, A.; Ruoppa, L.; Lehtomäki, M.; Yu, X.; Kukko, A.; Kaartinen, H.; Taher, J.; Matikainen, L.; Hyyppä, E.; Luoma, V.; et al. Individual tree segmentation and species classification using high-density close-range multispectral laser scanning data. *ISPRS-J. Photogramm. Remote Sens.* **2023**, *9*, 100039. [CrossRef]
2. Hyyppä, E.; Kukko, A.; Kaartinen, H.; Yu, X.; Muhojoki, J.; Hakala, T.; Hyyppä, J. Direct and automatic measurements of stem curve and volume using a high-resolution airborne laser scanning system. *Sci. Remote Sens.* **2022**, *5*, 100050. [CrossRef]
3. Hyyppä, E.; Kukko, A.; Kaijaluoto, R.; White, J.C.; Wulder, M.A.; Pyörälä, J.; Liang, X.; Yu, X.; Wang, Y.; Kaartinen, H.; et al. Accurate derivation of stem curve and volume using backpack mobile laser scanning. *ISPRS-J. Photogramm. Remote Sens.* **2020**, *161*, 246–262. [CrossRef]
4. Hu, T.; Sun, X.; Su, Y.; Guan, H.; Sun, Q.; Kelly, M.; Guo, Q. Development and Performance Evaluation of a Very Low-Cost UAV-Lidar System for Forestry Applications. *Remote Sens.* **2021**, *13*, 21. [CrossRef]
5. Rudge, M.L.M.; Levick, S.R.; Bartolo, R.E.; Erskine, P.D. Modelling the Diameter Distribution of Savanna Trees with Drone-Based LiDAR. *Remote Sens.* **2021**, *13*, 18. [CrossRef]
6. Torre-Tojal, L.; Bastarrika, A.; Boyano, A.; Lopez-Guede, J.M.; Graña, M. Above-ground biomass estimation from LiDAR data using random forest algorithms. *J. Comput. Sci.* **2022**, *58*, 14. [CrossRef]
7. Chen, S.; Feng, Z.; Chen, P.; Khan, T.; Lian, Y. Nondestructive Estimation of the Above-Ground Biomass of Multiple Tree Species in Boreal Forests of China Using Terrestrial Laser Scanning. *Forests* **2019**, *10*, 26. [CrossRef]
8. 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 Observ. Remote Sens.* **2020**, *13*, 1055–1067. [CrossRef]
9. Panagiotidis, D.; Abdollahnejad, A.; Slavik, M. 3D point cloud fusion from UAV and TLS to assess temperate managed forest structures. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102917. [CrossRef]
10. Disney, M.I.; Vicari, M.B.; Burt, A.; Calders, K.; Lewis, S.L.; Raumonen, P.; Wilkes, P. Weighing trees with lasers: Advances, challenges and opportunities. *Interface Focus.* **2018**, *8*, 14. [CrossRef]
11. Xu, D.; Wang, H.; Xu, W.; Luan, Z.; Xu, X. LiDAR Applications to Estimate Forest Biomass at Individual Tree Scale: Opportunities, Challenges and Future Perspectives. *Forests* **2021**, *12*, 19. [CrossRef]
12. Liang, X.; Hyyppä, J.; Kaartinen, H.; Lehtomäki, M.; Pyörälä, J.; Pfeifer, N.; Holopainen, M.; Brolly, G.; Pirotti, F.; Hackenberg, J.; et al. International benchmarking of terrestrial laser scanning approaches for forest inventories. *ISPRS-J. Photogramm. Remote Sens.* **2018**, *144*, 137–179. [CrossRef]

13. Zhong, L.; Cheng, L.; Xu, H.; Wu, Y.; Chen, Y.; Li, M. Segmentation of Individual Trees from TLS and MLS Data. *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **2017**, *10*, 774–787. [CrossRef]
14. Hillman, S.; Wallace, L.; Reinke, K.; Jones, S. A comparison between TLS and UAS LiDAR to represent eucalypt crown fuel characteristics. *ISPRS-J. Photogramm. Remote Sens.* **2021**, *181*, 295–307. [CrossRef]
15. Ayrey, E.; Fraver, S.; Kershaw, J.A.; 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]
16. Silva, C.A.; Hudak, A.T.; Vierling, L.A.; Valbuena, R.; Cardil, A.; Mohan, M.; Almeida, D.R.A.; Broadbent, E.N.; Zambrano, A.M.A.; Wilkinson, B.; et al. Treetop: A Shiny-based application and R package for extracting forest information from LiDAR data for ecologists and conservationists. *Methods Ecol. Evol.* **2022**, *13*, 1164–1176. [CrossRef]
17. Wallace, L.; Lucieer, A.; Watson, C.S. Evaluating Tree Detection and Segmentation Routines on Very High Resolution UAV LiDAR Data. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7619–7628. [CrossRef]
18. Ma, K.; Chen, Z.; Fu, L.; Tian, W.; Jiang, F.; Yi, J.; Du, Z.; Sun, H. Performance and Sensitivity of Individual Tree Segmentation Methods for UAV-LiDAR in Multiple Forest Types. *Remote Sens.* **2022**, *14*, 20. [CrossRef]
19. Yan, W.; Guan, H.; Cao, L.; Yu, Y.; Gao, S.; Lu, J. An Automated Hierarchical Approach for Three-Dimensional Segmentation of Single Trees Using UAV LiDAR Data. *Remote Sens.* **2018**, *10*, 16. [CrossRef]
20. Balsi, M.; Esposito, S.; Fallavollita, P.; Nardinocchi, C. Single-tree detection in high-density LiDAR data from UAV-based survey. *Eur. J. Remote Sens.* **2018**, *51*, 679–692. [CrossRef]
21. Yang, Q.; Su, Y.; Jin, S.; Kelly, M.; Hu, T.; Ma, Q.; Li, Y.; Song, S.; Zhang, J.; Xu, G.C.; et al. The Influence of Vegetation Characteristics on Individual Tree Segmentation Methods with Airborne LiDAR Data. *Remote Sens.* **2019**, *11*, 18. [CrossRef]
22. Dalla Corte, A.P.; Souza, D.V.; Rex, F.E.; Sanquetta, C.R.; Mohan, M.; Silva, C.A.; Zambrano, A.M.A.; Prata, G.; de Almeida, D.R.A.; Trautenmüller, J.W.; et al. Forest inventory with high-density UAV-Lidar: Machine learning approaches for predicting individual tree attributes. *Comput. Electron. Agric.* **2020**, *179*, 14.
23. Wu, X.; Shen, X.; Cao, L.; Wang, G.; Cao, F. Assessment of Individual Tree Detection and Canopy Cover Estimation using Unmanned Aerial Vehicle based Light Detection and Ranging (UAV-LiDAR) Data in Planted Forests. *Remote Sens.* **2019**, *11*, 21. [CrossRef]
24. Dong, T.; Zhou, Q.; Gao, S.; Shen, Y. Automatic Detection of Single Trees in Airborne Laser Scanning Data through Gradient Orientation Clustering. *Forests* **2018**, *9*, 19. [CrossRef]
25. Brede, B.; Lau, A.; Bartholomeus, H.M.; Kooistra, L. Comparing RIEGL RiCOPTER UAV LiDAR Derived Canopy Height and DBH with Terrestrial LiDAR. *Sensors* **2017**, *17*, 16. [CrossRef] [PubMed]
26. Picos, J.; Bastos, G.; Míguez, D.; Alonso, L.; Armesto, J. Individual Tree Detection in a Eucalyptus Plantation Using Unmanned Aerial Vehicle (UAV)-LiDAR. *Remote Sens.* **2020**, *12*, 17. [CrossRef]
27. Yin, D.; Wang, L. Individual mangrove tree measurement using UAV-based LiDAR data: Possibilities and challenges. *Remote Sens. Environ.* **2019**, *223*, 34–49. [CrossRef]
28. Panagiotidis, D.; Abdollahnejad, A.; Surovy, P.; Chiteculo, V. Determining tree height and crown diameter from high-resolution UAV imagery. *Int. J. Remote Sens.* **2017**, *38*, 2392–2410. [CrossRef]
29. Lu, X.; Guo, Q.; Li, W.; Flanagan, J. A bottom-up approach to segment individual deciduous trees using leaf-off lidar point cloud data. *ISPRS-J. Photogramm. Remote Sens.* **2014**, *94*, 1–12. [CrossRef]
30. Tao, S.; Wu, F.; Guo, Q.; Wang, Y.; Li, W.; Xue, B.; Hu, X.; Li, P.; Tian, D.; Li, C.; et al. Segmenting tree crowns from terrestrial and mobile LiDAR data by exploring ecological theories. *ISPRS-J. Photogramm. Remote Sens.* **2015**, *110*, 66–76. [CrossRef]
31. Lee, H.; Slatton, K.C.; Roth, B.E.; Cropper, W.P. Adaptive clustering of airborne LiDAR data to segment individual tree crowns in managed pine forests. *Int. J. Remote Sens.* **2010**, *31*, 117–139. [CrossRef]
32. Xia, S.; Wang, C.; Pan, F.; Xi, X.; Zeng, H.; Liu, H. Detecting Stems in Dense and Homogeneous Forest Using Single-Scan TLS. *Forests* **2015**, *6*, 3923–3945. [CrossRef]
33. Comesaña-Cebral, L.; Martínez-Sánchez, J.; Lorenzo, H.; Arias, P. Individual Tree Segmentation Method Based on Mobile Backpack LiDAR Point Clouds. *Sensors* **2021**, *21*, 17. [CrossRef] [PubMed]
34. Trochta, J.; Krucek, M.; Vrska, T.; Král, K. 3D Forest: An application for descriptions of three-dimensional forest structures using terrestrial LiDAR. *PLoS ONE* **2017**, *12*, 17. [CrossRef] [PubMed]
35. Burt, A.; Disney, M.; Calders, K. Extracting individual trees from lidar point clouds using treeseg. *Methods Ecol. Evol.* **2019**, *10*, 438–445. [CrossRef]
36. Wu, B.; Yu, B.; Yue, W.; Shu, S.; Tan, W.; Hu, C.; Huang, Y.; Wu, J.; Liu, H. A Voxel-Based Method for Automated Identification and Morphological Parameters Estimation of Individual Street Trees from Mobile Laser Scanning Data. *Remote Sens.* **2013**, *5*, 584–611. [CrossRef]
37. Xing, W.; Xing, Y.; Huang, Y.; Qu, L.; You, H. Individual tree segmentation of TLS point cloud data based on clustering of voxels layer by layer. *J. Cent. South. Univ. For. Technol.* **2017**, *37*, 58–64.
38. Weiser, H.; Schäfer, J.; Winiwarter, L.; Krašovec, N.; Seitz, C.; Schimka, M.; Anders, K.; Baete, D.; Braz, A.S.; Brand, J.; et al. Terrestrial, UAV-Borne, and Airborne Laser Scanning Point Clouds of Central European Forest Plots, Germany, with Extracted Individual Trees and Manual Forest Inventory Measurements. Available online: https://doi.pangaea.de/10.1594/PANGAEA.942856 (accessed on 31 March 2022). [CrossRef]

39. Dalponte, M.; Coomes, D.A. Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data. *Methods Ecol. Evol.* **2016**, *7*, 1236–1245. [CrossRef] [PubMed]
40. 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]
41. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An Easy-to-Use Airborne LiDAR Data Filtering Method Based on Cloth Simulation. *Remote Sens.* **2016**, *8*, 22. [CrossRef]
42. Fu, H.; Li, H.; Dong, Y.; Xu, F.; Chen, F. Segmenting Individual Tree from TLS Point Clouds Using Improved DBSCAN. *Forests* **2022**, *13*, 18. [CrossRef]
43. Wei, C.; Huang, J.; Mansaray, L.; Li, Z.; Liu, W.; Han, J. Estimation and Mapping of Winter Oilseed Rape LAI from High Spatial Resolution Satellite Data Based on a Hybrid Method. *Remote Sens.* **2017**, *9*, 16. [CrossRef]
44. Akbulut, Y.; Sengur, A.; Guo, Y.; Smarandache, F. NS-k-NN: Neutrosophic Set-Based k-Nearest Neighbors Classifier. *Symmetry* **2017**, *9*, 10. [CrossRef]
45. Olofsson, K.; Holmgren, J.; Olsson, H. Tree Stem and Height Measurements using Terrestrial Laser Scanning and the RANSAC Algorithm. *Remote Sens.* **2014**, *6*, 4323–4344. [CrossRef]
46. Roussel, J.-R.; Auty, D.; Coops, N.C.; Tompalski, P.; Goodbody, T.R.H.; Meador, A.S.; Bourdon, J.-F.; de Boissieu, F.; Achim, A. lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. *Remote Sens. Environ.* **2020**, *251*, 112061. [CrossRef]
47. Lisiewicz, M.; Kaminska, A.; Kraszewski, B.; Sterenczak, K. Correcting the Results of CHM-Based Individual Tree Detection Algorithms to Improve Their Accuracy and Reliability. *Remote Sens.* **2022**, *14*, 22. [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.