Article

# Street Tree Extraction and Segmentation from Mobile LiDAR Point Clouds Based on Spatial Geometric Features of Object Primitives

**Zhenyang Hui [1], Zhuoxuan Li [1], Shuanggen Jin [2,3], Bo Liu [1,*] and Dajun Li [1]**

[1] Faculty of Geomatics, East China University of Technology, Nanchang 330013, China
[2] Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, China
[3] School of Surveying and Land Information Engineering, Henan Polytechnic University, Jiaozuo 454000, China
* Correspondence: liubo@ecut.edu.cn

**Abstract:** Extracting street trees from mobile Light Detection and Ranging (LiDAR) point clouds is still encountering challenges, such as low extraction accuracy and poor robustness in complex urban environment, and difficulty in the segmentation of overlapping trees. To solve these problems, this paper proposed a street tree extraction and segmentation method based on spatial geometric features of object primitives. In this paper, mobile LiDAR point clouds were first segmented into object primitives based on the proposed graph segmentation method, which can release the computation burden effectively. According to the spatial geometric features of the segmented object primitives, stem points were extracted. In doing so, the robustness and accuracy for stem detecting can be improved. Furthermore, voxel connectivity analysis and individual tree optimization were combined successively. In doing so, the neighboring trees could be separated successfully. Four datasets located in Henan Polytechnic University, China, were used for validating the performance of the proposed method. The four mobile LiDAR point clouds contained 106, 45, 76, and 46 trees, respectively. The experimental results showed that the proposed method can achieve the performance of individual tree separation in all the four testing plots. Compared to the other three methods, the proposed method can make a good balance between the commission and omission errors and achieved the highest average F1 scores.

**Keywords:** mobile LiDAR; object primitive; special geometric feature; street tree

**Citation:** Hui, Z.; Li, Z.; Jin, S.; Liu, B.; Li, D. Street Tree Extraction and Segmentation from Mobile LiDAR Point Clouds Based on Spatial Geometric Features of Object Primitives. *Forests* **2022**, *13*, 1245. https://doi.org/10.3390/f13081245

Academic Editors: Henning Buddenbaum

Received: 30 June 2022
Accepted: 5 August 2022
Published: 6 August 2022

## 1. Introduction

Constructing a high-precision tree model is important for improving environmental quality and urban resident service [1–3]. The traditional methods of obtaining street tree information mainly include manual, Real-Time Kinematic, and total station measurements. These methods require a great deal of manpower and material resources. Moreover, the low information collecting efficiency of these measurements cannot meet the needs of the rapid extraction of street tree information [4].

LiDAR technology is a rapidly developing active remote sensing technology that can quickly obtain accurate three-dimensional (3D) point clouds on the surface of objects [5]. Although commercial photogrammetry software can generate millions of points from images in a cheaper way, this technique cannot provide additional information, such as intensity, reflected echoes, etc., which will be also useful for post-application. Thus, LiDAR technology has been widely applied in road detection, power line extraction, forest parameter estimation, 3D model reconstruction, and other fields [6,7].

According to different platforms, LiDAR systems can be divided into airborne LiDAR, terrestrial LiDAR, and mobile LiDAR [8]. The airborne LiDAR has the advantages of wide measurement range and high efficiency [9]. Therefore, Airborne Laser Scanning (ALS) data have been widely used in a wide range of trees extraction [10–13]. Yao et al. [14] jointly analyzed spectral, geometric, and spatial context attributes from airborne LiDAR data and imagery. Experimental results showed that the trees in both residential and street areas could be extracted and segmented well. Zhang et al. [15] proposed an urban vegetation extraction and segmentation method by combining hyperspectral images and airborne LiDAR data. Due to canopy overlap, the individual trees are often under-segmented or over-segmented. To solve the difficulty in determining a segmentation stop criterion, Amiri et al. [16] proposed an adaptive stop criterion based on the visual appearance of trees. Although airborne LiDAR can be used to extract 3D urban information in a wide range, it is a top-down scanning system, whose signal is easily blocked by the canopy, making it difficult to obtain complete tree stem information [17].

Compared with the airborne LiDAR, terrestrial LiDAR generally can obtain more dense point clouds with higher accuracy, especially in local areas. Thus, terrestrial LiDAR is one of the important means in forest inventory, which can be used for individual tree measurement and reconstruction [18–20]. Kiraly and Brolly [21] proposed a method of individual tree detection based on statistical clustering. Lindberg et al. [22] adopted the Hough transform to identify and detect stems in terrestrial laser scanning (TLS) data. Pueschel et al. [23] firstly stratified the data according to the elevation value and then used the distance difference between adjacent points as the condition of stem detection to determine the stem position and extract the complete tree. Zhong et al. [24] proposed a tree detection and segmentation method based on octree structure. Experimental results showed that the stem detection method based on an octree node histogram could distinguish stems from other rod-shaped ground objects such as lamp poles and pedestrians. Although TLS can obtain dense point clouds, its poor mobility and low efficiency limit TLS to be applied in larger urban environments [25,26].

Compared with airborne LiDAR and terrestrial LiDAR, mobile LiDAR can obtain the 3D information of urban roads and surrounding objects more efficiently [27,28]. Mobile laser scanning (MLS) can not only obtain more complete tree stem points than ALS, but also can achieve a larger measurement range than TLS in a short time. Although TLS has a similar workflow for individual tree detection as MLS, the point clouds acquired by TLS must be registered to obtain complete point clouds. Obviously, the registration process is prone to error. More importantly, the data acquisition efficiency of TLS is much lower than that of MLS in urban environments. Consequently, the street tree extraction and segmentation method based on MLS data is still mainstream [29]. Lin et al. [30] adopted the three-layer frame strategy and the RD-schematic algorithm to detect street trees using the morphological characteristics of canopy surface model. Zhong et al. [31] combined RGB information for tree separation. Wu et al. [32] proposed a new voxel-based marked neighborhood searching method. The experimental results showed that the completeness and correctness of this method for street tree detection were both above 98%. Guan et al. [33] used Euclidean distance clustering and voxel-based normalized cut segmentation to extract individual trees. Huang et al. [34] extracted complete street trees, according to the eigenvalue and horizontal information. Husain et al. [35] proposed an analytic hierarchy process to classify tree objects. Yadav et al. [36] detected stems with a bottom-up strategy according to the linearity and data distribution homogeneity. The experimental results showed that this method could identify trees effectively from the MLS data of road scenes. Li et al. [37] proposed an individual tree extraction method that worked on points directed and adopted the supervised learning algorithm. Li et al. [38] proposed a branch-stem-constrained hierarchical clustering method to extract street trees from the MLS data.

Although MLS data can be used for extracting street tree information in urban areas efficiently, street tree extraction and segmentation based on MLS still involve the following challenges:

i. The robustness of the street tree extraction method is poor; when encountering complex urban environment extraction, accuracy will be low.
ii. The objects that are linear features are easily mistaken as tree stems, reducing the accuracy of street tree extraction.
iii. Neighboring clustered trees are difficult to separate, which will lead to a larger individual tree extraction error.

To solve the problems mentioned above, a street tree extraction and segmentation method based on spatial geometric features of object primitives was proposed in this paper. In this paper, constraint conditions were first set to construct a graph structure to obtain the object primitives from MLS data. Then, stem points were detected according to the spatial geometric features of the segmented object primitives. After that, the street tree points were extracted by voxelizing the point clouds and conducting voxel component analysis. Finally, by comparing the shortest path from each point to the root points, the individual trees were separated.

The remainder of this paper is organized as follows. Section 2 describes the principle of the proposed method. In Section 3, the experiments are described. Section 4 discusses and analyzes the experimental results. Section 5 concludes this paper.

## 2. Methodology

The flowchart of the proposed method is shown in Figure 1. First, an improved morphological filtering method proposed by Hui et al. [39] was used to separate the ground points and non-ground points. Then, the non-ground points with higher elevations were removed. In this paper, the elevation threshold was set to 2 m. In doing so, almost all the tree stems can be retained while crown points or high building points can be removed. Subsequently, the graph structure was constructed and the object primitives were obtained by graph segmentation under the proposed constraints in this paper. After that, the spatial characteristics of the object primitives were calculated to extract the tree stem points. Then, the non-ground points were voxelized, and the initial tree points were extracted by analyzing the connectivity of adjacent voxels. To separate neighboring clustered trees successfully, this paper segmented the initial tree points through the shortest path analysis to obtain the individual trees. Four main steps were included in this paper: (i) object primitive acquisition based on graph segmentation, (ii) stems detection using object primitive characteristics, (iii) initial tree identification based on the connectivity analysis, and (iv) individual tree optimization based on voxel shortest path analysis.

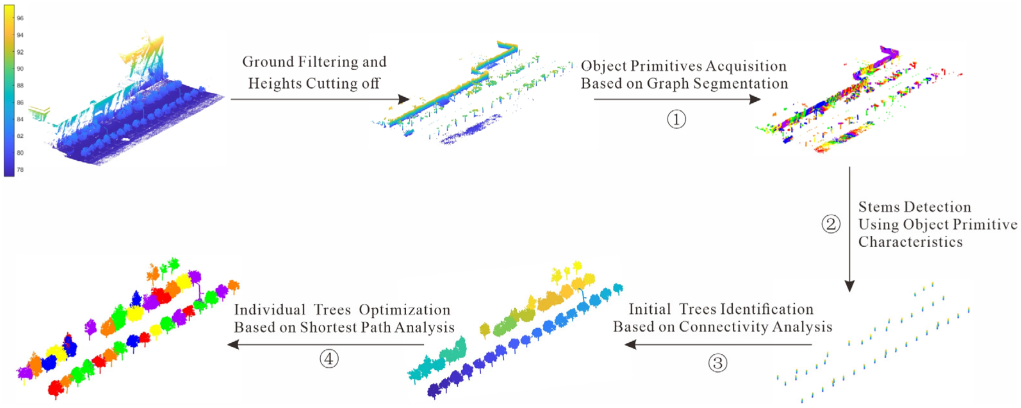

Ground Filtering and
Heights Cutting off
Object Primitives Acquisition
Based on Graph Segmentation
①
②
Stems Detection
Using Object Primitive
Characteristics
Initial Trees Identification
Based on Connectivity Analysis
③
Individual Trees Optimization
Based on Shortest Path Analysis
④

**Figure 1.** Flowchart of the proposed method.

### 2.1. *Object Primitives Acquisition Based on Graph Segmentation*

To extract tree stems effectively, this paper first acquired the initial object primitives by building a graph structure of the points. The graph is defined as Equation (1) [40]:

$$G = (V, E) \tag{1}$$

where $V$ represents nodes ($v_i$) and $E$ represents edges ($e_{i,j}$). $v_i$ is made up by all points ($p_i, i = 1, 2, \ldots, N$), while $e_{i,j}$ connects pairs of neighboring points ($p_i, p_j$). In general, edge construction determines the results of the graph segmentation. To extract tree stem objectives effectively, this paper constructed the graph based on the proposed three constraints, as shown in Figure 2.

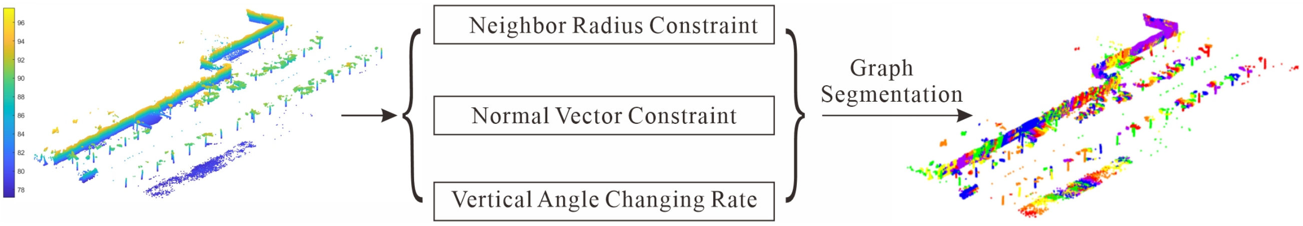


**Figure 2.** Graph segmentation. The left figure shows the points colored according to the elevation before segmentation. The right figure shows the result of segmentation; different object primitives have different colors.

#### 2.1.1. The Constraint of Neighboring Radius

To limit the complexity of the graph, the edges within the graph should be constrained. The effective way is to construct the edges only within the neighboring points. In general, there are two means that can be used for defining the neighboring points. One is to set a fixed number of neighboring points. However, the density of MLS data is varied; if a fixed number of neighboring points is selected, then there will be longer edges within the graph. This is because the distance constraint is lacked. The graph structure tends to have long edges when the point cloud is sparse or outliers exist. The other way of defining neighboring points is to set a fixed neighboring radius. However, the threshold of the fixed neighboring radius is hard to determine. To solve this problem, this paper combined these two neighboring determination methods together. First, *n* points were selected from tree points randomly, and then, the *k* nearest neighboring points for the *n* points were detected. For each set of *k* nearest neighbors, the longest distance within the *k* points can be calculated. The fixed neighboring radius is calculated as the average value of these *k* longest distances. As shown in Figure 3a, *R* is the calculated fixed neighboring radius, which can be defined as Equation (2):

$$R = \frac{\sum_{i=1}^{n} \max\left(dis_j^i, j = 1, 2, \cdots, k\right)}{n} \tag{2}$$

where *n* is the number of randomly selected points, $dis_j^i$ is the *j*-th neighboring distance, and *k* is the number of neighboring points. It must be noted that *k* influences the graph complexity and implementation efficiency. Considering the balance between them, *k* is set to 10 in this paper [41]. According to Equation (2), the neighborhood radius can be automatically determined according to the density and spatial distribution of different data.

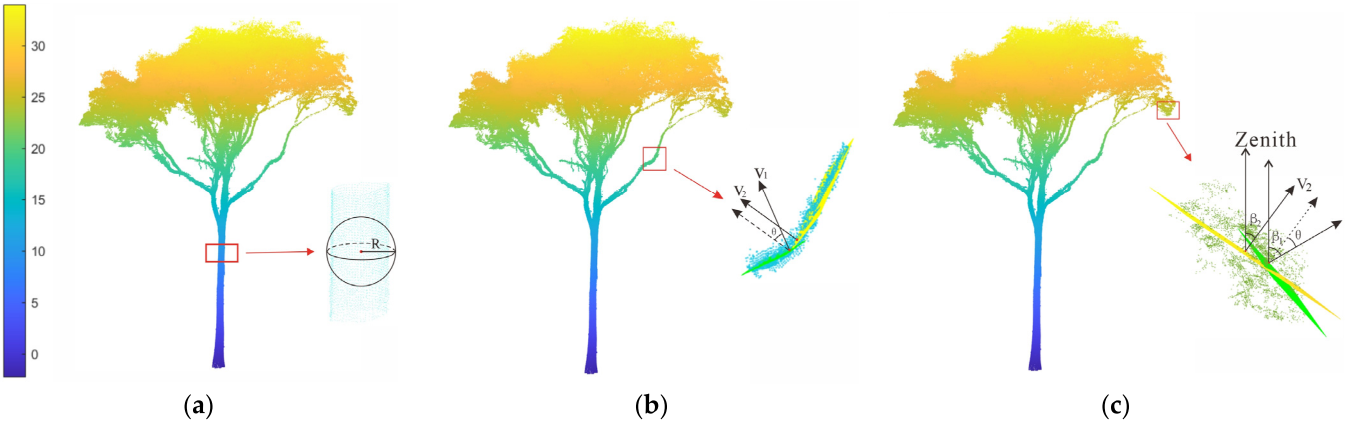


**Figure 3.** Graph segmentation constraints: (**a**) the constraint of neighborhood radius; (**b**) the constraint of the angle between normal vectors; (**c**) the constraint of the vertical angle gradient.

### 2.1.2. The Constraint of the Angle between Normal Vectors

In general, the object primitives of tree stems have a normal vector consistency. That is, the normal vector angle between adjacent points is small. Therefore, this feature can be used to separate these non-ground objects into the same object primitives. On the contrary, the points of crowns and low vegetation are scattered and do not have the consistency of the normal vector. As a result, these objects will be divided into small, scattered object primitives. Thus, to extract the tree stem correctly, the points within the same object primitive should have similar normal vectors. In this paper, the angle ($\theta$) of normal vectors between each point and its neighboring points is calculated to obtain the object primitives with similar spatial characteristics, as shown in Figure 3b. If $\theta$ is less than the threshold, the edge between the two points is preserved. In this paper, the threshold was set to 30° by trial and error. Based on this threshold, stem points can be better extracted. $\theta$ is defined as Equation (3).

$$\theta = \arccos \frac{v_1 \cdot v_2}{\|v_1\|\|v_2\|} \tag{3}$$

where $v_1$ and $v_2$ are the normal vectors of two adjacent nodes.

### 2.1.3. The Constraint of the Vertical angle Changing Rate

In general, stem points have regular vertical distribution, while other points, such as crown points, are usually in random scattered distribution. In other words, the angles between the normal vectors of crown points and the Z direction generally change greatly than that of stem points. Thus, the third constraint was set as the vertical angle changing rate in this paper, which is defined as Equation (4):

$$\beta_{rate} = \sqrt{\frac{\sum_{i=1}^{num} \Delta\beta_i^{\,2}}{num}} \tag{4}$$

where $\triangle\beta_i$ is the vertical angle difference between the point and its neighboring points; as shown in Figure 3c, $num$ is the number of neighboring points. As shown in Figure 2, the points with similar vertical angle changing rates can be divided into the same object primitives, such as tree stems and buildings. Meanwhile, the points with larger vertical angle changing rates, such as tree canopies, were divided into several smaller object primitives.

### 2.2. Stems Detection Using Geometric Characteristics of Object Primitives

After the graph segmentation, the point clouds will be divided into multiple object primitives. The stem primitives usually present independent cylindrical shapes. To extract stem points accurately, this paper calculated the geometric features of each object primitive, as shown in Figure 4a. Figure 4b shows the calculation of linearity. It can be found that most stem primitives own large linearity, such as the yellow points shown in Figure 4c. The linearity of object primitives is defined as Equation (5):

$$linearity_{obj} = (\lambda_1 - \lambda_2) / \lambda_1 \tag{5}$$

where $\lambda$ is the eigenvalue of the covariance matrix of the primitive. Here, $\lambda_1 > \lambda_2 > \lambda_3 > 0$. From Figure 4b, it can be found that, compared with crown points, $\lambda_1$ of the stem and branch is significantly larger than $\lambda_2$. Thus, the linearities of stem and branch points are generally larger than the ones of crown points.

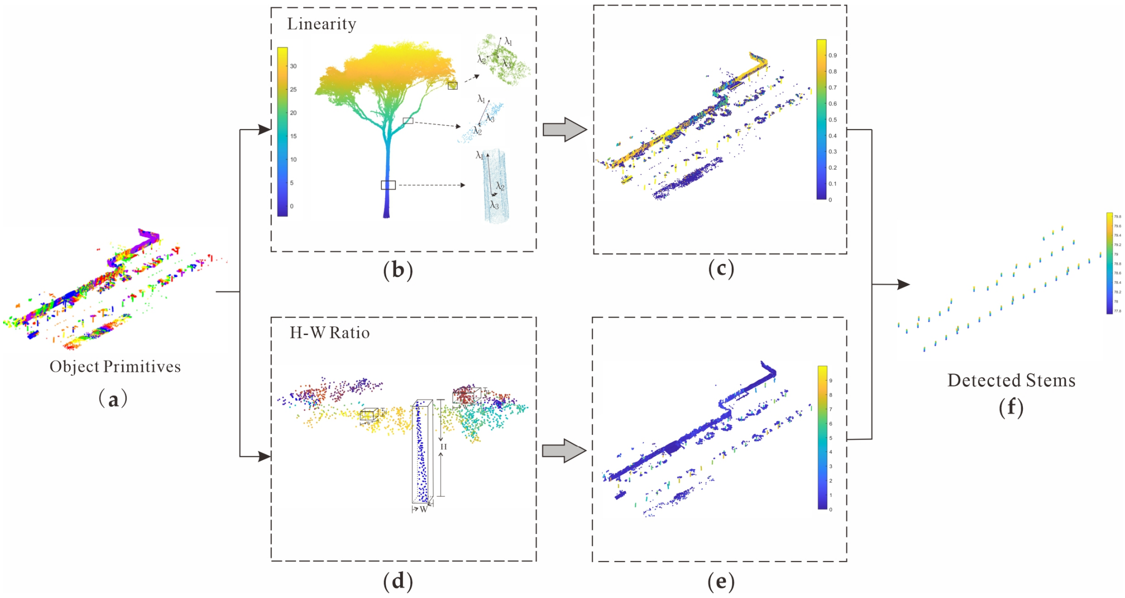


**Figure 4.** Stem detection using geometric characteristics of object primitives: (**a**) is the initial object primitives. The linearity and H-W ratio of each object primitive are calculated and shown in (**b**–**e**), respectively. The detected stems are shown in (**f**).

Since the objects, such as buildings and low vegetation, may also have high linearities, another spatial feature was adopted in this paper to detect tree stems. This paper defined it as the Height-Width (H-W) ratio of the object primitives. In this paper, the H-W ratio refers to the ratio of the height and width of the minimum bounding box of each object, as shown in Figure 4d. As the stems are in cylindrical shapes, the H-W ratio of stem primitives is larger than that of other primitives. Therefore, the stem points can be further optimized by detecting the primitives with a larger H-W ratio. The H-W ratio is defined as Equation (6):

$$ratio_{obj} = \frac{h_{obj}}{\sqrt{{\delta_x}^2 + {\delta_y}^2}} \tag{6}$$

Where $h_{obj}$ is calculated as the elevation difference between the highest and lowest points within the object. $\delta_x$ and $\delta_y$ are the ranges of points within the object primitive on the X and Y axes, respectively. Based on linearity and the H-W ratio, the stems can be detected as shown in Figure 4f.

### 2.3. Initial Tree Points Identification Based on Voxel Connectivity Analysis

After tree stem extraction, initial tree points were acquired based on voxel connectivity analysis. As shown in Figure 5a, stems (red points) were detected by the method described in Section 2.2. Subsequently, all the points were voxelized as a series of voxels. To obtain initial tree points, the voxel connectivity analysis was applied to the voxelization result. The process of voxel connectivity analysis is shown in Figure 6 [42]. Figure 6a shows the voxels before segmentation. The voxels with no point are labeled as 0, while the voxels containing points are labeled as 1. Figure 6b is the six-connected model used in this paper. This means only the voxels distributed as the six-connected model can be seen as connectivity. In the step of voxel connectivity analysis, the voxels of stems were selected as the seed voxels. Then, the neighboring voxels were searched, and the connectivity to the seed voxels was judged. Only the adjacent voxels with common faces are regarded as having connectivity, as mentioned above. Subsequently, the connected result can be obtained as shown in Figure 6c.

Using the proposed voxel connectivity analysis, the voxels connected to stems can be merged together, as shown in Figure 5c. According to the connected results, the whole tree points can be obtained as shown in Figure 5d. It should be noted that, in the step of stems detection, some elongated objects, such as lamp posts, telephone poles, etc., will also be misclassified as tree stems. This is because these object primitives also own a larger linearity and H-W ratio simultaneously. To avoid the influence of these wrongly detected stems, this paper removed the objects with a smaller number of voxels after the voxel connectivity analysis. This is because the size of the wrongly detected trees, such as lamp posts, are generally smaller. Thus, by limiting the number of voxels for each object, the wrongly detected trees can be removed.

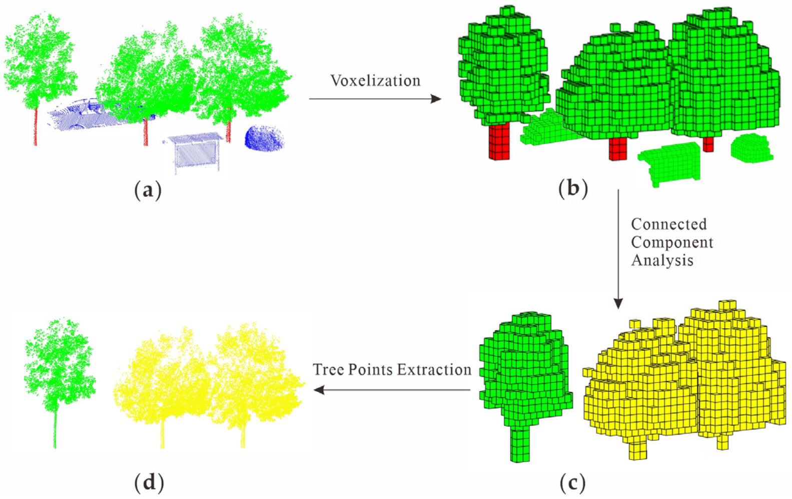


**Figure 5.** Tree point extraction based on the voxel connectivity analysis: (**a**) point clouds acquired by MLS; (**b**) the result of voxelization; (**c**) the outcome of the voxel connectivity analysis; (**d**) the extracted tree points.

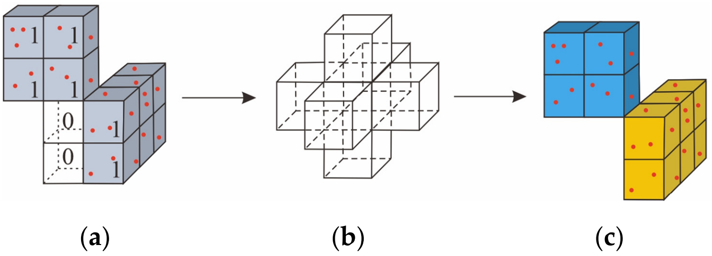


**Figure 6.** Process of the voxel connectivity analysis. (**a**) The voxels before segmentation. The voxels with no point are labeled as 0, while the voxels containing points are labeled as 1. (**b**) A

six-connected model; (**c**) the result after the voxel connectivity analysis. Different object primitives are colored with different colors.

### *2.4. Individual Trees Optimization Based on Voxel Shortest Path Analysis*

From Figure 5, it can be found that, after the voxel connectivity analysis, tree points can be discriminated from other points based on the detected tree stems. However, there are some neighboring clustered trees that are connected as one tree, as shown in Figure 5d. This is because the distance between the two neighboring trees is close; when conducting voxel connectivity analysis, it will be easy to connect the neighboring trees as one tree. Thus, further optimization should be conducted to obtain optimal individual tree separation results.

The flowchart of individual tree optimization is shown in Figure 7. In this method, each tree primitive was traversed one by one. The tree primitives are the tree point extraction results after the voxel connectivity analysis. Obviously, there will be one or more trees contained in each tree primitive. To determine this, the stems within each tree primitive are counted. During the traversing, if more than one stem is within the tree, then the tree primitive is under-segmented and should be further optimized. In the optimization, the under-segmented tree primitive is first voxelized, and then, the center of each voxel forms the built graph. In doing so, the calculation burden will be released, and the efficiency will be improved. Since the under-segmented tree primitive contains more than one stem, each point within the tree primitive will have a shortest path (SP) to a different base of stems, according to the Dijkstra algorithm shown in Figure 8. Obviously, the points should belong to the stems with shorter SPs. The Dijkstra algorithm is a famous SP calculation method [43]. In the Dijkstra algorithm, the direct path from the source node to the end nodes is first calculated. Then, the shortest path among them can be calculated, which will be used for adjusting the other paths. Within the adjusted paths, the shortest one can be found. This process is iterated until all the SPs from the source node to the other end nodes are detected. By comparing the SPs, the neighboring trees can be optimized as individual trees.

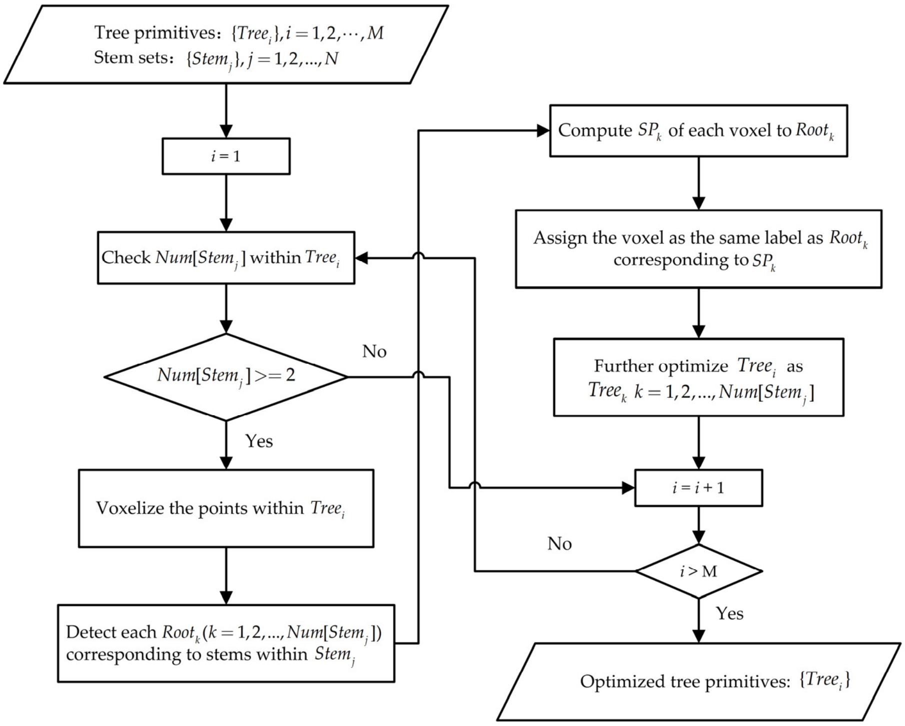

Tree primitives: $\{Tree_i\}, i=1,2,\cdots,M$
Stem sets: $\{Stem_j\}, j=1,2,...,N$
$i=1$
Check $Num[Stem_j]$ within $Tree_i$
$Num[Stem_j] >= 2$
No
Yes
Voxelize the points within $Tree_i$
Detect each $Root_k (k=1,2,...,Num[Stem_j])$ corresponding to stems within $Stem_j$
Compute $SP_k$ of each voxel to $Root_k$
Assign the voxel as the same label as $Root_k$ corresponding to $SP_k$
Further optimize $Tree_i$ as $Tree_k$ $k=1,2,...,Num[Stem_j]$
$i=i+1$
No
$i>M$
Yes
Optimized tree primitives: $\{Tree_i\}$

**Figure 7.** The flowchart of individual tree optimization based on the shortest path analysis.

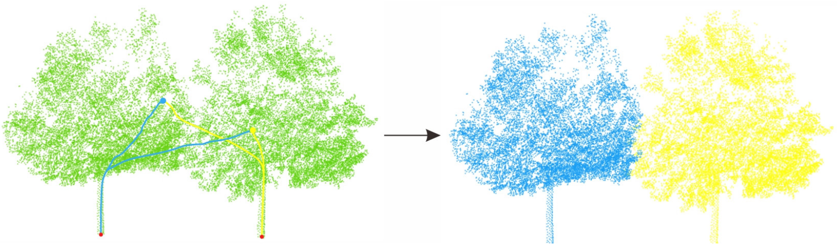


**Figure 8.** Individual tree optimization. The left figure shows two trees with overlapping crowns before optimization. The shortest path length from the blue point to the bottom of the left tree is obviously smaller than the shortest path length from it to the bottom of the right tree, so the blue point is classified as the same class with the bottom of the left tree. The right figure is the result of optimization.

## 3. Experiment

### 3.1. Experimental Datasets

To evaluate the performance of the proposed method, four datasets located in the Henan Polytechnic University, China, were used for testing [44]. The datasets were obtained by SSW-2 MLS along road scenes in the campus. The acquired point clouds include street trees, buildings, street lamps, cars, pedestrians, etc. As shown in Figure 9, the street trees in four areas have different distribution characteristics. In Sample 1 (Figure 9a) and Sample 4 (Figure 9d), there are a large number of street trees with different heights and overlapping crowns. In Sample 2 (Figure 9b) and Sample 3 (Figure 9c), most trees are regularly distributed and have similar heights. In addition, along the street trees of the four datasets there are many different objects, such as buildings, cars, and road lamps. The characteristics of the four datasets are tabulated in Table 1. From Table 1, it can be found that the datasets can test the extraction performance of the proposed method for street trees with different distribution characteristics to verify the effectiveness and robustness of the method.

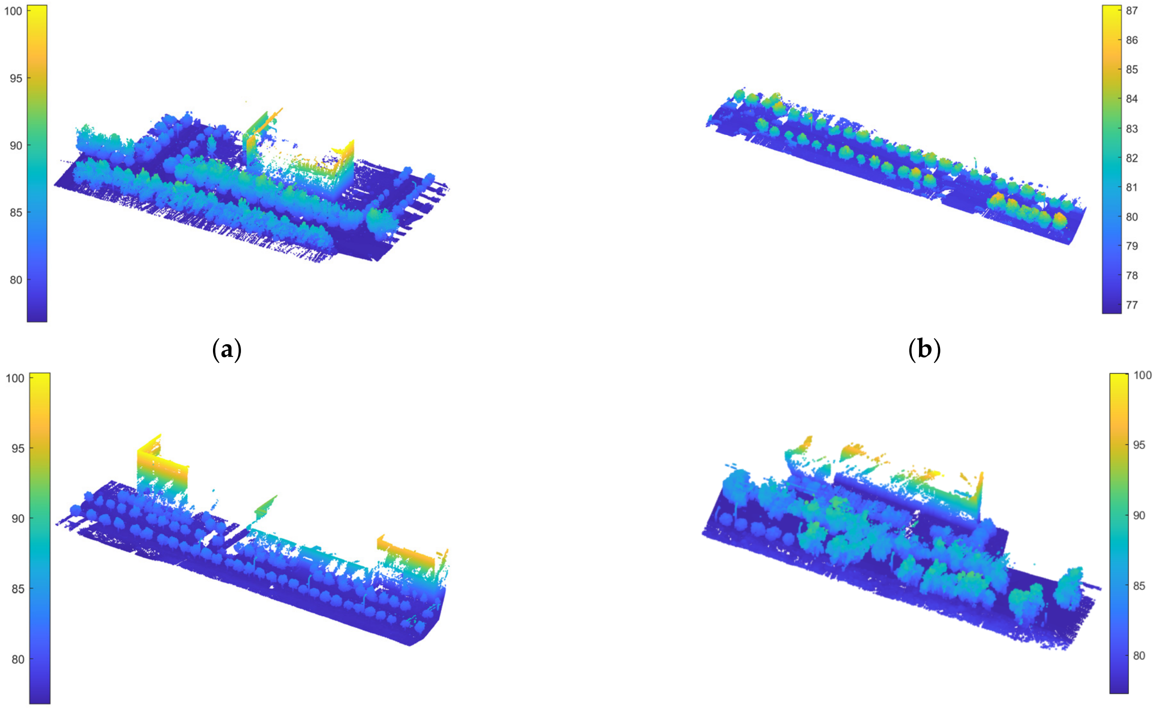


**Figure 9.** Four MLS plots with different characteristics: (**a**) Sample 1; (**b**) Sample 2; (**c**) Sample 3; (**d**) Sample 4. The points are colored based on elevation.

**Table 1.** The characteristics of the four testing datasets.

| Area | Total Points | Tree Points | Non-tree Points | Street Trees | Tree Species | Non-Tree Objects |
|---|---|---|---|---|---|---|
| Sample 1 | 4,463,551 | 2,185,245 | 2,278,306 | 106 | firmiana, populus | buildings, cars, lamps, bicycles, pedestrians, low vegetation |
| Sample 2 | 2,018,179 | 792,204 | 1,225,975 | 45 | populus | cars, lamps, bicycles, pedestrians, low vegetation, advertising boards |
| Sample 3 | 2,109,819 | 709,638 | 1,400,181 | 76 | populus | buildings, cars, lamps, low vegetation |
| Sample 4 | 2,281,056 | 1,170,290 | 1,110,766 | 46 | willow, populus | buildings, cars, lamps, bicycles, pedestrians, low vegetation |

### *3.2. Experimental Results*

The proposed method was implemented in a MATLAB 2020a environment on a laptop computer with an Intel® Core™ i7-9750H CPU, 16.0 GB of RAM, and a Windows 10 64-bit operating system. The final outcome of the proposed method is the separated individual trees. Thus, to evaluate the performance of the proposed method, the referenced individual trees should be obtained. In this paper, the referenced individual trees were acquired manually using a visual software named CloudCompare [45]. Specially, tree tops of the segmented individual trees were detected manually, which serve as the reference results. In this paper, the performance of individual tree separation was accessed based on treetops matching, which was developed by Eysn et al. [46]

The process of treetop matching algorithm is shown in Figure 10. This matching method firstly detected the candidates from the highest tree tops. The tree tops within a radius of the reference tree tops were judged as the candidates. In this paper, the radius was set to 5 m. After that, the candidates were ranked depending on their height difference and the two-dimensional (2D) distance between the candidates and the reference tree tops. Then, the candidate tree tops with a large height difference were removed. If a candidate tree top showed a better height difference and its 2D distance difference from the nearest candidate was less than 2.5 m, it was determined as the matching point of the reference tree top.

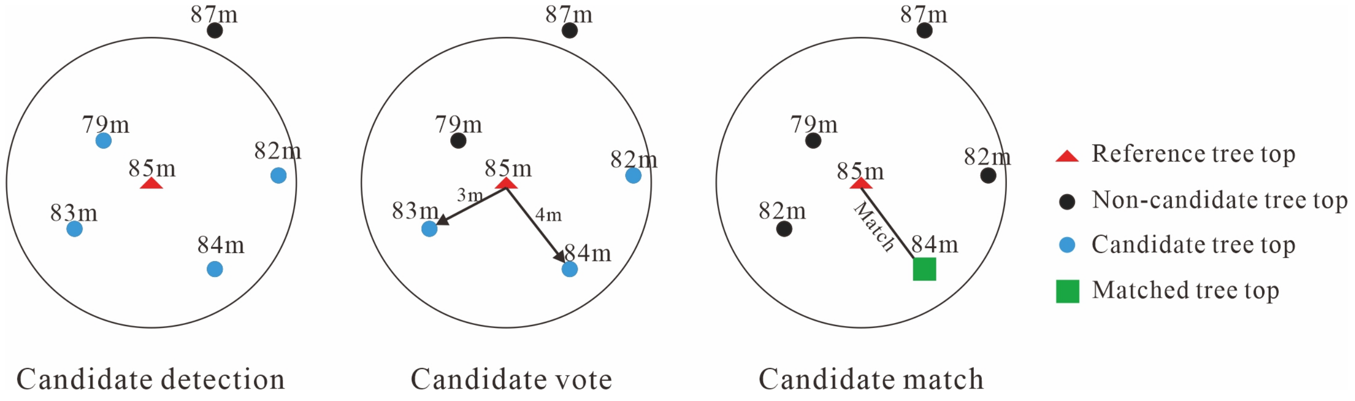


**Figure 10.** The treetop matching method. The treetops within a radius of the reference tree tops were judged as the candidates. Voting candidates with the height difference and 2D distance between the candidates and the reference tree tops. If a candidate tree top showed a better height difference and its 2D distance difference from the nearest candidate was less than 2.5 m, it was determined as the matching tree top. The numbers above the points indicate the elevation values.

Five indicators were adopted to evaluate the precision of the individual tree segmentation results. They are extraction rate, matching rate, commission error, omission error, and F1 score. The above five indicators are calculated as Equations (7)–(11):

$$Extraction\ rate = \frac{Num\_test}{Num\_ref} \tag{7}$$

$$Matching\ rate = \frac{Num\_match}{Num\_ref} \tag{8}$$

$$Commission\ error = \frac{Num\_test - Num\_match}{Num\_test} \tag{9}$$

$$Omission\ error = \frac{Num\_ref - Num\_match}{Num\_ref} \tag{10}$$

$$F_1 = 2 \times \frac{Num\_match}{Num\_ref + Num\_test} \tag{11}$$

where $Num\_test$ is the number of detected tree tops in the segmentation result, i.e., the total number of the detected individual trees, $Num\_ref$ represents the tree tops in the reference data, i.e., the accurate number of individual trees, $Num\_match$ is the number of detected tree tops matching the reference data, i.e., the number of correctly detected trees. The extraction rate represents the percentage of the detected trees to the total number of reference trees. The matching rate indicates how well the detected trees match the reference. Omission error is the percentage of trees that are not correctly segmented. Commission error tends to represent the over-segmentation appearance of the method. F1 score is a comprehensive indicator that reflect the effectiveness of the method.

Table 2 tabulates the accuracy calculation results of the proposed method. From Table 2, it can be found that the extraction rates in the four datasets were all close to 1. This means that the proposed method did not tend to over-segment more trees or wrongfully detect less trees. The average matching rate is larger than 0.85. Thus, it can be concluded that the proposed method can achieve a good extraction result compared with referenced individual trees. The commission and omission errors in Sample 4 are a little larger. This is because there are many weeping willows in Sample 4. Some branches of the weeping willows are attached to the ground, which will easily be misclassified as stems. Thus, the detection errors in this area are larger, while the F1 score is a little smaller. However, the F1 scores of the other three areas are also larger than 0.8. As a result, the average F1 score for the four areas is still larger than 0.8. This indicates that the proposed method can achieve good individual tree detection performance.

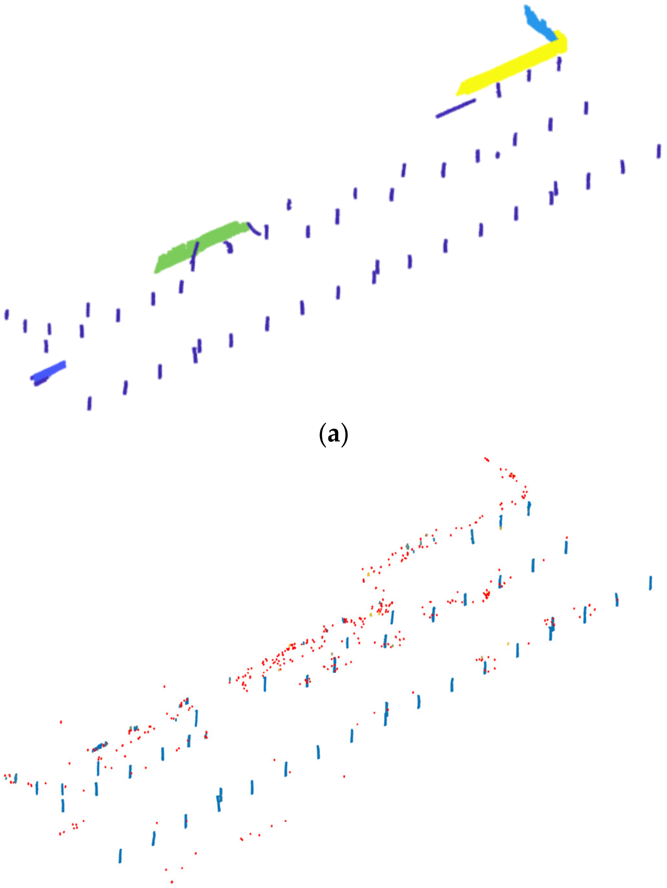
**Table 2.** The accuracy indicators of street tree segmentation results.

| Area | Extraction Rate | Matching Rate | Commission Error | Omission Error | $F_1$ |
|---|---|---|---|---|---|
| Sample 1 | 0.9823 | 0.7965 | 0.1892 | 0.2035 | 0.8036 |
| Sample 2 | 1.2955 | 0.9773 | 0.2456 | 0.0227 | 0.8515 |
| Sample 3 | 0.9868 | 0.8816 | 0.1067 | 0.1184 | 0.8874 |
| Sample 4 | 1.2381 | 0.7619 | 0.3846 | 0.2381 | 0.6809 |
| Average | 1.1257 | 0.8543 | 0.2315 | 0.1457 | 0.8059 |

## 4. Discussion and Analysis

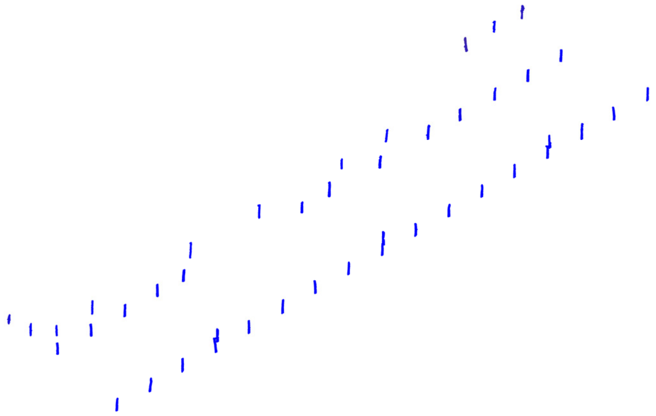
In this paper, three main steps were involved, including stem detection, tree point identification, and individual tree optimization. To further analyze each step in detail, this paper will discuss these three steps, respectively.

In the process of stem detection, two spatial geometric features of object primitives were involved, namely linearity ($linearity_{obj}$) and the H-W ratio ($ratio_{obj}$). The linearity reflects the similarity between the overall shape of each object primitive and the stems. The higher the value, the more likely the object primitive is to be a stem. Figure 11a shows the object primitives extracted only based on linearity. The linearity values of reserved object primitives are all higher than 0.9. It can be seen that most of the stems were extracted correctly, but there were still some other objects (green and yellow points) wrongly reserved. These are the facades of buildings whose linearity values meet the threshold. Therefore, when the linearity of the object primitives is only constrained, the vertical distributed stems and some other artificial objects extending horizontally or slanting are retained at the same time. Thus, the correctness of street tree extraction is low.



(**a**)

(**b**)



(**c**)

**Figure 11.** Stem extraction results obtained by different spatial geometric features: (**a**) only linearity constraint is used for stem extraction; (**b**) only H-W radio constraint is used for stem extraction; (**c**) both linearity and H-W radio constraints are used for stem extraction.

The H-W radio reflects the vertical distribution characteristics of object primitivity. For example, the H-W radio of the stem object primitive is usually higher than that of other object primitives. This is because the stem is a slender cylinder that grows vertically, and the height of the cylinder is significantly greater than its bottom diameter. Figure 11b is the object primitives extracted only based on the H-W ratio. From Figure 11b, some object primitives with fewer points were wrongly extracted (red points in Figure 11b). This is because these object primitives contain a few points, resulting in the widths of these object primitives being much smaller than the corresponding heights. As shown in Figure 11a,b, it can be concluded that the extracted stem results cannot be good under the constraint of linearity or H-W ratio separately. The object primitives wrongly extracted in Figure 11a have a smaller H-W ratio, while the object primitives wrongly extracted in Figure 11b have smaller linearity. Consequently, when the above two spatial geometric feature constraints are adopted simultaneously, the satisfying stem extraction results can be obtained, as shown in Figure 11c.

The second main step of the proposed method is tree point identification. The result of tree point identification directly affects the final individual tree separation outcome. In this paper, tree points were identified based on a voxel connectivity analysis. After tree point identification, point clouds in the scene can be classified as two categories, namely tree points and non-tree points. Thus, Type I, Type II, and total errors can be calculated to evaluate the performance of the proposed method. Here, Type I error means the percent of tree points misclassified as non-tree points, while Type II error represents the percent of non-tree points misclassified as tree points. Total error is the percent of points wrongly classified. The accuracy calculation results are tabulated in Table 3. From Table 3, it can be found that the errors of these three types are all smaller, except for the type II error of Sample 3. This is because there are several trees that are attached with adjacent walls in Sample 3. When applying voxel connectivity analysis to the detected stems, these wall points will be misclassified as tree points. As a result, its type II error and total error are a little larger. When calculating the average values of these three types of errors, it can be found that all the average errors are smaller than 5%. Thus, it can be concluded that the proposed method can achieve a good tree point identification result.

**Table 3.** Three types of errors of tree point identification.

| | **Type I Error (%)** | **Type II Error (%)** | **Total Error (%)** |
|---|---|---|---|
| Sample 1 | 3.07 | 0.59 | 1.74 |
| Sample 2 | 1.12 | 2.56 | 2.09 |
| Sample 3 | 1.87 | 11.01 | 8.41 |
| Sample 4 | 3.23 | 4.68 | 3.94 |
| Average | 2.32 | 4.71 | 4.04 |

After the tree points are detected, individual tree segmentation can be carried out. To evaluate the performance of the proposed segmentation method objectively, three other individual tree extraction methods were selected for comparative analysis. Chen et al. [47] proposed a marker-controlled watershed segmentation method (Watershed). In their method, the canopy maxima model was first established, and the variable size windows were used to detect treetops in it. After that, the marker-controlled watershed segmentation method was adopted to obtain individual trees. Wang [48] proposed an unsupervised method based on superpoint graph structure (SSSC). The point cloud was recursively segmented to obtain the object primitives, and the superpoint of each object primitive was extracted. Then, an undirected superpoint graph was constructed to calculate the eigenvectors of the superpoints. Finally, the object primitives represented by the superpoint were clustered based on the eigenvectors and shortest path analysis to achieve individual tree extraction. Latella et al. [49] proposed a density-based algorithm (ITDM) to detect individual trees. They first removed the points with low elevation, such as shrub and grassland points, to reduce the calculation cost. The point cloud was then projected onto a horizontal plane, and its projection density was calculated to extract locally dense points as stem points. Finally, the height maximum points around the stem location were extracted as the treetops.

Figures 12–15 show the accuracy comparison in four plots of the proposed method with the three methods mentioned above. It can be seen that the satisfying results in four plots were obtained by the proposed method. The F1 scores of the proposed method in four plots are all higher than those of other three methods, and the F1 scores of three plots are greater than 0.8. This shows that the proposed method has higher precision and stronger robustness compared to the other three methods. In the four plots, the proposed method achieves the best on three of five indicators. This indicates that the overall performance of the proposed method is the best compared with the other three methods. As shown in Figure 12, the proposed method can achieve the lowest commission rate and the best extraction rate. This shows that the proposed method performs well in the street trees with regular distribution and different heights. As shown in Figures 13 and 15, the proposed method has a low matching rate; however, its extraction rate is the best. It can be concluded that the proposed method can reduce the over-segmentation as much as possible, while ensuring the best segmentation. As shown in Figure 15, all methods have a higher commission error, which is related to the low detection correctness of the street trees. This is because there are lots of weeping willows in Sample 4. These elongated willow branches attached to the ground are often mistaken as stems. As a result, the number of wrongly extracted street trees is increased, leading to larger commission errors. Compared with the other three plots, the proposed method performed best in Sample 3, as shown Figure 14. All indicators of Sample 3 are the best out of the other three methods. This shows that the proposed method has a good detection and segmentation ability on the street trees with regular distribution.

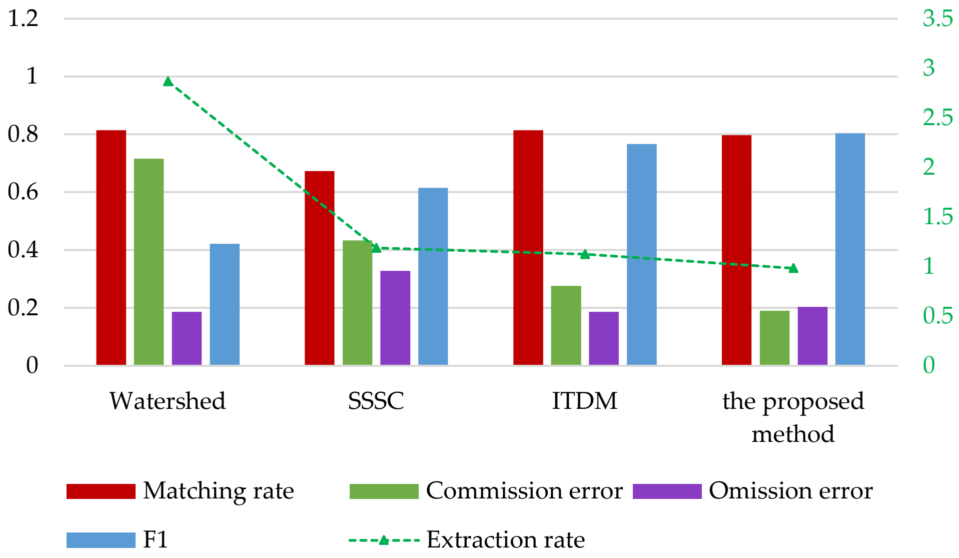


**Figure 12.** Accuracy comparison among four different methods in Sample 1.

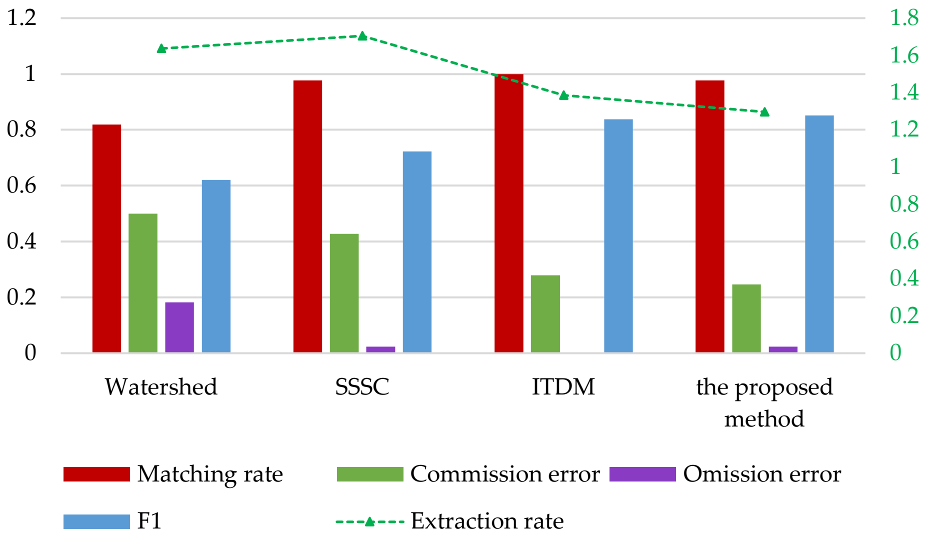


**Figure 13.** Accuracy comparison among four different methods in Sample 2.

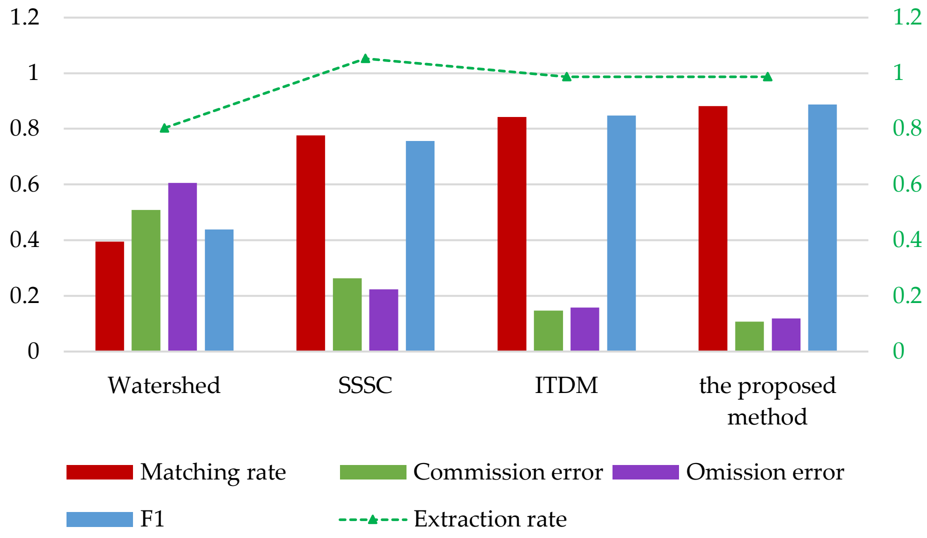


**Figure 14.** Accuracy comparison among four different methods in Sample 3.

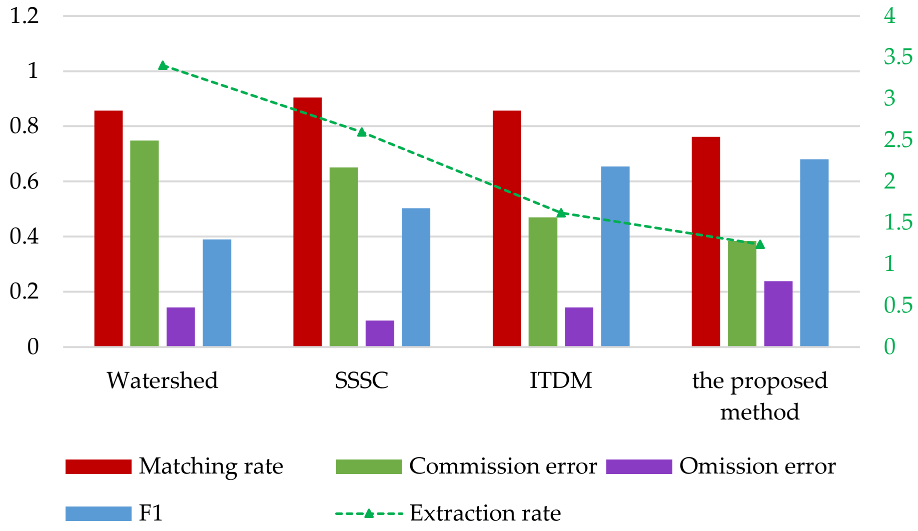


**Figure 15.** Accuracy comparison among four different methods in Sample 4.

**Table 4 shows the comparison of the average accuracy of the four methods.** From Table 4, it can be found that the extraction rate of the proposed method is the smallest one compared to the other three methods. The extraction rate of the Watershed method is about two times of that of the proposed method. This indicates that the proposed method will not try to over-segment trees. Although the average matching rate of the proposed method is slightly lower than that of the ITDM method, the average extraction rate and commission error of the proposed method are the best compared to the other three methods. As a result, the average F1 score of the proposed method is much higher than the other three methods. This shows that the proposed method can achieve good individual tree segmentation results in four plots.

**Table 4.** Comparison of average accuracy of the four methods. The bold represents the best value among the comparison results.

| Method | Extraction Rate | Matching Rate | Commission Error | Omission Error | $F_1$ |
|---|---|---|---|---|---|
| Watershed | 2.1778 | 0.7211 | 0.6181 | 0.2789 | 0.4672 |
| SSSC | 1.6346 | 0.8327 | 0.4433 | 0.1673 | 0.6494 |
| ITDM | 1.2790 | **0.8784** | 0.2929 | **0.1216** | 0.7767 |
| the proposed method | **1.1257** | 0.8543 | **0.2315** | 0.1457 | **0.8059** |

## 5. Conclusions

Street tree extraction is a significant step in the construction of a digital twin city. In this paper, mobile LiDAR point clouds were first segmented into object primitives based on the proposed graph segmentation under certain constraints. In doing so, the computation burden can be released. To improve the accuracy and robustness of stem detection, both geometric features of linearity and H-W ratio were combined. To most traditional individual tree detection methods, separating the neighboring trees is still a challenge. In this paper, voxel shortest path analysis was proposed to achieve optimized individual tree separation results. The proposed method was validated using four datasets. The experimental results showed that the proposed method can achieve good street tree extraction and segmentation performance in all four testing plots. Compared with the other three methods, the tree segmentation method in this paper achieved the highest average F1 score. This reveals that the proposed method can balance the commission rate and omission rate, effectively avoiding over-segmentation. However, the implementation of the proposed method requires setting some parameters, such as the thresholds of linearity and H-W ratio of the object primitives. How to improve the automation of the method, making the parameters determined automatically according to the particularity of each point cloud, will be focused on in our further research.

**Author Contributions:** Z.H. conceived the original idea of the study and drafted the manuscript. Z.L. and S.J. performed the experiments and made the experimental analysis. B.L. and D.L. contributed to the revision of the manuscript. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the National Natural Science Foundation of China (NSF) (42161060, 41801325, 42161064), the China Post-Doctoral Science Foundation (2019M661858), the Natural Science Foundation of Jiangxi Province (20192BAB217010, 20212BAB204003), and East China University of Technology Ph. D. Project (DHBK2017155).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Acknowledgments:** We thank the anonymous reviewers whose comments and constructive criticism helped improve and clarify this manuscript.

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

## References

1. Liu, Y.; Guo, J.; Benes, B.; Deussen, O.; Zhang, X.; Huang, H. Treepartnet: Neural decomposition of point clouds for 3d tree reconstruction. *ACM Trans. Graph.* **2021**, *40*, 232.
2. Safaie, A.H.; Rastiveis, H.; Shams, A.; Sarasua, W.A.; Li, J. Automated street tree inventory using mobile LiDAR point clouds based on Hough transform and active contours. *ISPRS J. Photogramm. Remote Sens.* **2021**, *174*, 19–34.
3. Hui, Z.; Jin, S.; Xia, Y.; Nie, Y.; Xie, X.; Li, N. A mean shift segmentation morphological filter for airborne LiDAR DTM extraction under forest canopy. *Opt. Laser Technol.* **2021**, *136*, 106728–106742.
4. Toth, C.; Jozkow, G. Remote sensing platforms and sensors: A survey. *ISPRS J. Photogramm. Remote Sens.* **2016**, *115*, 22–36.

5. Zheng, G.; Moskal, L.M.; Kim, S. Retrieval of effective leaf area index in heterogeneous forests with terrestrial laser scanning. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 777–786.
6. Luo, S.; Wang, C.; Xi, X.; Pan, F.; Peng, D.; Zou, J.; Nie, S.; Qin, H. Fusion of airborne LiDAR data and hyperspectral imagery for aboveground and belowground forest biomass estimation. *Ecol. Indic.* **2017**, *73*, 378–387.
7. Tarsha Kurdi, F.; Awrangjeb, M.; Munir, N. Automatic filtering and 2D modeling of airborne laser scanning building point cloud. *Trans. GIS.* **2021**, *25*, 164–188.
8. Xu, S.; Sun, X.; Yun, J.; Wang, H. A new clustering-based framework to the stem estimation and growth fitting of street trees from mobile laser scanning data. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2020**, *13*, 3240–3250.
9. Shao, J.; Zhang, W.; Shen, A.; Mellado, N.; Cai, S.; Luo, L.; Wang, N.; Yan, G.; Zhou, G. Seed point set-based building roof extraction from airborne LiDAR point clouds using a top-down strategy. *Autom. Constr.* **2021**, *126*, 103660–103672.
10. Höfle, B.; Hollaus, M.; Hagenauer, J. Urban vegetation detection using radiometrically calibrated small-footprint full-waveform airborne LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2012**, *67*, 134–147.
11. Heinzel, J.; Koch, B. Exploring full-waveform LiDAR parameters for tree species classification. *Int. J. Appl. Earth Observ. Geoinf.* **2011**, *13*, 152–160.
12. Dai, W.; Yang, B.; Dong, Z.; Shaker, A. A new method for 3D individual tree extraction using multispectral airborne LiDAR point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *144*, 400–411.
13. You, H.; Li, S.; Xu, Y.; He, Z.; Wang, D. Tree extraction from airborne laser scanning data in urban areas. *Remote Sens.* **2021**, *13*, 3428–3445.
14. Yao, W.; Wei, Y. Detection of 3-d individual trees in urban areas by combining airborne LiDAR data and imagery. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1355–1359.
15. Zhang, C.; Zhou, Y.; Qiu, F. Individual tree segmentation from LiDAR Point clouds for urban forest inventory. *Remote Sens.* **2015**, *7*, 7892–7913.
16. Amiri, N.; Polewski, P.; Heurich, M.; Krzystek, P.; Skidmore, A.K. Adaptive stopping criterion for top-down segmentation of ALS point clouds in temperate coniferous forests. *ISPRS J. Photogramm. Remote Sens.* **2018**, *141*, 265–274.
17. Puttonen, E.; Jaakkola, A.; Litkey, P.; Hyyppä, J. Tree classification with fused mobile laser scanning and hyperspectral data. *Sensors.* **2011**, *11*, 5158–5182.
18. McHale, M.R. Volume estimates of trees with complex architecture from terrestrial laser scanning. *J. Appl. Remote Sens.* **2008**, *2*, 023521–023539.
19. Zande, D.V.D.; Jonckheere, I.; Stuckens, J.; Verstraeten, W.W.; Coppin, P. Sampling design of ground-based lidar measurements of forest canopy structure and its effect on shadowing. *Can. J. Remote Sens.* **2008**, *34*, 526–538.
20. Holopainen, M.; Kankare, V.; Vastaranta, M.; Liang, X.; Lin, Y.; Vaaja, M.; Yu, X.; Hyyppä, J.; Hyyppä, H.; Kaartinen, H.; et al. Tree mapping using airborne, terrestrial and mobile laser scanning—A case study in a heterogeneous urban forest. *Urban For. Urban Gree.* **2013**, *12*, 546–553.
21. Király, G.; Brolly, G. Tree height estimation methods for terrestrial laser scanning in a forest reserve. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2007**, *3*, 211–215.
22. Lindberg, E.; Holmgren, J.; Olofsson, K.; Olsson, H. Estimation of stem attributes using a combination of terrestrial and airborne laser scanning. *Eur. J. Forest Res.* **2012**, *131*, 1917–1931.
23. Pueschel, P.; Newnham, G.; Rock, G.; Udelhoven, T.; Werner, W.; Hill, J. The influence of scan mode and circle fitting on tree stem detection, stem diameter and volume extraction from terrestrial laser scans. *ISPRS J. Photogramm. Remote Sens.* **2013**, *77*, 44–56.
24. Zhong, L.; Cheng, L.; Xu, H.; Wu, Y.; Chen, Y.; Li, M. Segmentation of individual trees from TLS and MLS data. *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.* **2017**, *10*, 774–787.
25. Jones, T.; Marzen, L.; Chappelka, A. Mapping, modeling, and estimating tree measurements of urban tree canopy structure using terrestrial LiDAR scanning. *Appl. Geogr.* **2016**, *2*, 236–242.
26. Xu, S.; Xu, S.; Ye, N.; Zhu, F. Automatic extraction of street trees' nonphotosynthetic components from MLS data. *Int. J. Appl. Earth Observ. Geoinf.* **2018**, *69*, 64–77.
27. Bienert, A.; Georgi, L.; Kunz, M.; von Oheimb, G.; Maas, H. Automatic extraction and measurement of individual trees from mobile laser scanning point clouds of forests. *Ann. Bot.* **2021**, *128*, 787–804.
28. Cabo, C.; Ordoñez, C.; García-Cortés, S.; Martínez, J. An algorithm for automatic detection of pole-like street furniture objects from mobile laser scanner point clouds. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 47–56.
29. Donager, J.J.; Sánchez Meador, A.J.; Blackburn, R.C. Adjudicating perspectives on forest structure: How do airborne, terrestrial, and mobile Lidar-derived estimates compare? *Remote Sens.* **2021**, *13*, 2297.
30. Lin, Y.; Hyyppä, J.; Jaakkola, A.; Yu, X. Three-level frame and RD-schematic algorithm for automatic detection of individual trees from MLS point clouds *Int. J. Remote Sens.* **2011**, *33*, 1701–1716.
31. Zhong, R.; Wei, J.; Su, W.; Chen, Y.F. A method for extracting trees from vehicle-borne laser scanning data. *Math. Comput. Model.* **2013**, *58*, 733–742.
32. 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.
33. Guan, H.; Yu, Y.; Ji, Z.; Li, J.; Zhang, Q. Deep learning-based tree classification using mobile LiDAR data. *Remote Sens. Lett.*

**2015**, *6*, 864–873.

34. Huang, P.; Chen, Y.; Li, J.; Yu, Y.; Wang, C.; Nie, H.; Abe, T. Extraction of street trees from mobile laser scanning point clouds based on subdivided dimensional features. In Proceedings of the 2015 IEEE International Geoscience and Remote Sensing Symposium IGARSS, Milan, Italy, 26–31 July 2015.
35. Husain, A.; Vaishya, R.C. Detection and thinning of street trees for calculation of morphological parameters using mobile laser scanner data. *Remote Sens. Appl. Society Environ.* **2018**, *13*, 375–388.
36. Yadav, M.; Lohani, B. Identification of trees and their trunks from mobile laser scanning data of roadway scenes. *Int. J. Remote Sens.* **2020**, *41*, 1233–1258.
37. Li, Q.; Yuan, P.; Liu, X.; Zhou, H. Street tree segmentation from mobile laser scanning data. *Int. J. Remote Sens.* **2020**, *41*, 7145–7162.
38. 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*, 110440–110454.
39. Hui, Z.; Hu, Y.; Yevenyo, Y.Z.; Yu, X. An improved morphological algorithm for filtering airborne LiDAR point cloud based on multi-level kriging interpolation. *Remote Sens.* **2016**, *8*, 35–50.
40. 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.
41. Becker, C.; Häni, N.; Rosinskaya, E.; Angelo, E.D.; Strecha, C. Classification of aerial photogrammetric 3d point clouds. *ISPRS Ann. Photogramm.* **2017**, *IV-1/W1*, 3–10.
42. Wan, P.; Zhang, W.; Jin, S.; Wang, T.; Yan, G. Plot-level wood-leaf separation of trees using terrestrial LiDAR data based on a segmentwise geometric feature classification method. 2020. Available online: https://www.researchgate.net/publication/343205371 (accessed on 6 Aug 2022).
43. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numer. Math.* **1959**, *1*, 269–271.
44. The 7th National Lidar Conference. Available online: http://chxy.hpu.edu.cn/lidar2021/data.htm (accessed on 1 October 2021).
45. CloudCompare. Available online: http://www.cloudcompare.org/main.html (accessed on 25 July 2022).
46. Eysn, L.; Hollaus, M.; Lindberg, E.; Berger, F.; Monnet, J.; Dalponte, M.; Kobal, M.; Pellegrini, M.; Lingua, E.; Mongus, D.; et al. A benchmark of Lidar-based single tree detection methods using heterogeneous forest data from the alpine space. *Forests* **2015**, *6*, 1721–1747.
47. Chen, Q.; Baldocchi, D.; Gong, P.; Kelly, M. Isolating individual trees in a savanna woodland using small footprint LiDAR data. *Photogramm. Eng. Remote Sens.* **2006**, *8*, 923–932.
48. Wang, D. Unsupervised semantic and instance segmentation of forest point clouds. *ISPRS J. Photogramm. Remote Sens.* **2020**, *165*, 86–97.
49. Latella, M.; Sola, F.; Camporeale, C. A density-based algorithm for the detection of individual trees from LiDAR data. *Remote Sens.* **2021**, *13*, 322–343.