

An Individual Tree Segmentation Method From Mobile Mapping Point Clouds Based on Improved 3-D Morphological Analysis

Weixi Wang^{1b}, Member, IEEE, Yuhang Fan, You Li, Xiaoming Li^{1b}, Member, IEEE, and Shengjun Tang^{1b}, Member, IEEE

Abstract—Street tree extraction based on the 3-D mobile mapping point cloud plays an important role in building smart cities and creating highly accurate urban street maps. Existing methods are often over- or under-segmented when segmenting overlapping street tree canopies and extracting geometrically complex trees. To address this problem, we propose a method based on improved 3-D morphological analysis for extracting street trees from mobile laser scanner (MLS) point clouds. First, the 3-D semantic point cloud segmentation framework based on deep learning is used for preclassification of the original point cloud to obtain the vegetation point cloud in the scene. Considering the influence of terrain unevenness, the vegetation point cloud is deterraformed and slice point cloud containing tree trunks is obtained through spatial filtering on height. On this basis, a voxel-based region growing method constrained with the changing rate of convex area is used to locate the street trees. Then we propose a progressive tree crown segmentation method, which first completed the preliminary individual segmentation of the tree crown point cloud based on the voxel-based region growth constrained by the minimum increment rule, and then optimizes the crown edges by “valley” structure-based clustering. In this article, the proposed method is validated and the accuracy is evaluated using three sets of MLS datasets collected from different scenarios. The experimental results show that the method can effectively identify and localize street trees with different geometries and has a good segmentation effect for street trees with large adhesion between canopies. The accuracy and recall of tree localization are higher than 96.08% and 95.83%, respectively, and the average precision and recall of instance segmentation in three datasets are higher than 93.23% and 95.41%, respectively.

Manuscript received 8 November 2022; revised 3 January 2023; accepted 30 January 2023. Date of publication 8 February 2023; date of current version 23 March 2023. This work was supported in part by the National Key Research and Development Program of China under Project 2022YFB3903700, in part by the Research Program of Shenzhen S and T Innovation Committee under Project JCYJ20210324093012033, in part by the Natural Science Foundation of Guangdong Province under Project 2121A1515012574, in part by the Open Fund of Key Laboratory of Urban Land Resources Monitoring and Simulation, MNR under Project KF-2021-06-125, in part by the National Natural Science Foundation of China under Project 71901147, Project 41901329, Project 41971354, and Project 41971341, and in part by Shenzhen Polytechnic Institute of IOT, Shenzhen 518055, China under Project 42001389. (Corresponding author: Shengjun Tang.)

Weixi Wang, Yuhang Fan, Xiaoming Li, and Shengjun Tang are with the Key Laboratory of Urban Land Resources Monitoring & Simulation, Ministry of Natural Resources, Shenzhen University, Shenzhen 518034, China (e-mail: wangwx@szu.edu.cn; 931367653@qq.com; lixming@szu.edu.cn; shengjun-tang@szu.edu.cn).

You Li is with the Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen University, Shenzhen 518034, China (e-mail: liyou@szu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3243283

Index Terms—Aquaculture ponds extraction, diffusion model, hyperspectral image, image superresolution, remote sensing, unsupervised classification.

I. INTRODUCTION

AS A vital component of the street environment, street trees play an important role in reducing noise [1] and improving air quality and other aspects of the urban environment [2]. Three-dimensional models of individual trees are used in road improvement design, 3-D modeling of street trees, urban climate studies, monitoring of road tree growth, and parameter extraction and estimation of road tree biomass. These applications rely on the accurate extraction of information such as the location, height, and canopy width of trees in the street environment [3]. For example, to create highly accurate navigation maps of cities, the obstruction of the field of view (FOV) by trees must be accurately calculated, which requires the accurate detection of tree contours as well as locations [4]. In the power industry, vegetation easily interferes with power lines due to its rapid growth, so obtaining accurate information regarding the height, location, and other characteristics of trees is crucial [5]. These and many other applications illustrate the practical value of accurately extracting individual trees in urban streets. With the rapid development of 3-D sensing, computer vision, and other technologies, laser scanning is widely used for 3-D data acquisition in cities. Compared with airborne laser scanning, the 3-D point clouds acquired by MLS on the ground produce complete structures with a higher level of detail and are, thus, widely used for high-quality urban data collection and road modeling. Subsequently, a large number of researchers are committed to extracting building facades [6], vegetation element segmentation [7], street facilities extraction and modeling [8], and semantic segmentation of 3-D scenes from mobile scanning laser point clouds [9]. Segmentation and extraction of tree elements based on mobile scanning laser data is an important process in the current high-precision map generation and 3-D scene modeling in smart cities, but some challenges remain.

- 1) In addition to vegetation elements, the vehicle’s 3-D point cloud also contains a large number of other urban targets, such as buildings, roads, and urban infrastructure. Removing such elements from the complex urban environment and accurately segmenting vegetation elements remains difficult.

- 2) Existing algorithms tend to under-segment or miss trees when 3-D point cloud data are incomplete due to occlusions.
- 3) When vegetation on urban streets is too dense, resulting in large adhesions between trees, existing algorithms are also prone to problems such as inaccurate canopy segmentation and large errors in tree parameter calculations.

Considering the above problems, this article proposes an improved 3-D morphology analysis method for accurate extraction and parameter calculation of street trees in complex street environments. This method compensates for the shortcomings of existing methods and has significant advantages, particularly in the extraction of trees with adhesion cases. The rest of this article is organized as follows. In Section II, previous methods of extracting individual street trees based on MLS point clouds are presented. In Section III, the street tree extraction method proposed in this article is introduced in detail. Three sets of data are used to verify the accuracy and robustness of the proposed method in Section IV. Finally, Section V concludes this article.

II. RELATED WORKS

In recent years, researchers have proposed many methods to extract individual row trees from MLS point clouds, which can be divided into three main categories: cluster-based methods, graph cut-based methods, and contextual information-based methods.

A. Cluster-Based methods

The cluster-based approach first preprocesses the original point cloud. Then, the objects are segmented by instances using various clustering algorithms, and individual trees are identified according to the size, height, location, and shape of the clusters. The K-means clustering algorithm and density-based spatial clustering of applications with noise (DBSCAN) clustering algorithm, as classical density-based point cloud clustering methods, are widely used in single tree extraction. For example, [10] and [11] first preprocessed original data using methods such as principal component analysis to remove the ground and buildings from the original point clouds, and then used the improved K-means clustering algorithm and the DBSCAN clustering algorithm to extract a single tree from the remaining point cloud. In the K-means clustering algorithm, the number of clusters must be determined artificially, and it is difficult to detect nonspherical tree crowns when the crowns overlap. Alternatively, the DBSCAN algorithm is more sensitive to the threshold of density and tends to reject some crown points as noisy point clouds, which results in incomplete crowns. To solve this issue, [12] and [13] segmented trees from LiDAR data using the mean drift method. However, the clustering method based on adaptive density is prone to over- and under-segmentation in tree crown extraction when the trees are mixed among other street components. In contrast to direct tree segmentation methods, some researchers have used progressive segmentation methods to improve the accuracy of tree crown segmentation. These methods first perform tree localization over the trunk and then perform further individual segmentation of trees. The authors

in [14] used the constraints of the morphological characteristics of trunks and crowns, including the horizontal positional relationship between trunk and crown, cylindrical morphology, and crown diameter, to distinguish the trunk from various road poles and finish tree localization. Then, a voxel-based region growth method could be used to segment tree crowns. This type of method can effectively extract tree crowns when trees are accurately located. However, it remains difficult to accurately segment trees when crowns are sagging. In contrast, [15] first utilized the super voxel-based clustering method to enhance the robustness of tree localization, in which super voxels are processed through principal component analysis (PCA) and trunks are identified by regional growth on the super voxels. Overall, the clustering method can typically segment individual trees, but these methods are mainly used for tree extraction in simple scenes due to difficulties in obtaining accurate extraction results for scenes with challenging situations or complex tree structures.

B. Graph Cut-Based Methods

Graph cut-based methods are effective and popular energy optimization algorithms widely used in the field of computer vision for image segmentation and stereo vision. In recent years, such approaches also have been used to segment point clouds by obtaining the topology relations between points using methods such as the kd-tree algorithm and then segmenting objects by the feature vector of the weight matrix. The authors in [16] and [17] individually segmented trees from laser point clouds using the graph cut method, in which the radius parameter was calculated based on the position of the tree crown and trunk. These methods can greatly improve the detection of crowns in the lower and middle heights of a tree, but the accuracy of trunk identification is sensitive to laser point cloud density. The authors in [18] located trees based on local maxima in a horizontal histogram of point cloud octree nodes and their shape features, and then used Voronoi diagrams and Ncut segmentation to achieve instance segmentation. Moreover, [19] combined machine learning methods and graph-cut methods for individual tree segmentation. They first constructed a global graph model and then used graph-cut-based clustering to achieve tree instance segmentation. The graph-cut-based approach can generally achieve individual tree segmentation in nonadhesive scenes, but this method relies on manually set parameters, such as the tree radius. It therefore struggles to obtain robust tree extraction results when tree features are highly variable or contain different tree species in the scenes.

C. Contextual Information-Based Methods

Tree segmentation methods based on contextual information use the nearest neighbor feature of each point in a point cloud as the contextual feature of the corresponding point. As early as 2006, Lalonde et al. [20] attempted to use 3-D descriptors to characterize the local geometric features of point clouds and classify the geometric tensor features of the target point cloud into three categories: voluminous, which mainly represent volumetric objects (e.g., grass, treetops); faceted, which mainly

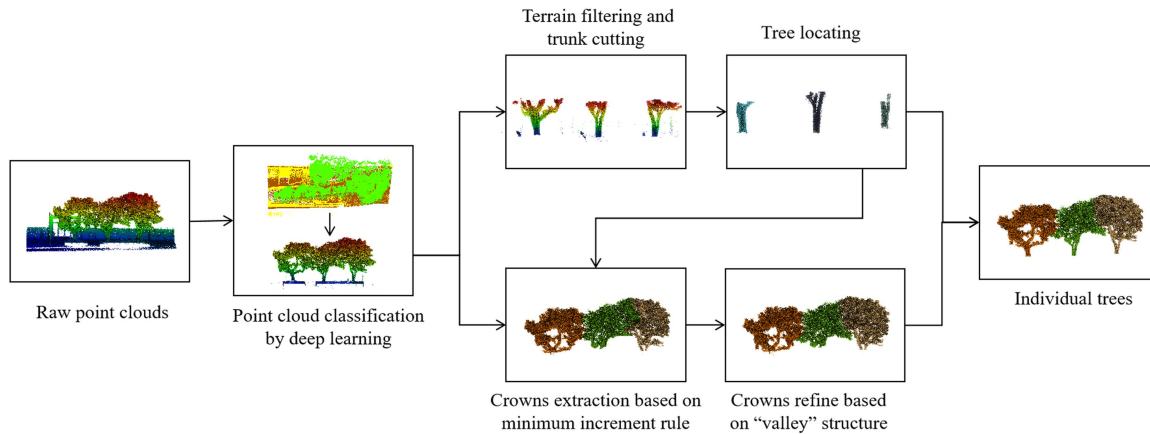


Fig. 1. Overview of the proposed method.

represent planar objects (e.g., ground, elevation); and linear, which mainly represent linear objects (e.g., tree trunks, light poles). Such methods can roughly classify scenes, but the classification results are often too coarse to be used. Subsequently, [7] proposed a cylindrical fitting model by modifying the existing probabilistic relaxation model, which identifies cylindrical objects as tree trunks (and voluminous features) and merges the adjacent tree to construct the street model. However, this method has poor extraction accuracy for tree crowns when there are large areas of adhesions. The authors in [21] also proposed a new 3-D segmentation algorithm for dominant tree detection by using the symmetric structure of the tree, but the search radius parameter in this method must be manually determined based on *a priori* knowledge, and its generalizability must be improved. Although context-based methods can roughly segment point clouds into several types, they have difficulty with accurately extracting individual trees and can over- or under-segment in complex scenes. The aforementioned state-of-the-art methods can successfully identify and extract individual street trees in cases with simple trunk shapes, large tree spacing, and small overlaps between trees. However, when the distance between trees is too small, the adhesion is severe, or the tree branching structure is complex, the existing algorithms struggle to achieve accurate individual tree segmentation and tend to poorly generalize parameter settings and over- or under-segment. To address these issues, we propose an improved 3-D morphological analysis-based method for individual tree segmentation in complex street environments. The main contributions of this work are as follows.

- 1) The data preprocessing phase involves 3-D deep learning to presegment point clouds and reduce the impact of terrain unevenness on individual tree segmentation through the terrain filtering process.
- 2) A voxel-based region growing algorithm based on the rate of change of the convex area is proposed to locate street trees, which enables accurate positioning of trees with complex geometry.
- 3) A coarse-to-fine canopy extraction method for adhesion scenes is proposed, which integrates the area increment rule and height change rule to achieve multiscale area growth of crown point clouds.

III. METHODOLOGY

Fig. 1 shows the overall technical flow of this method. First, the raw data are preprocessed to remove noise and perform a preliminary classification of the point clouds to obtain tree point clouds. The digital elevation model is then created by filtering, and the total elevation of the point cloud is unified to a datum before locating the trees. Based on these tree localization results, a coarse to fine tree canopy extraction method is used to accurately extract the individual tree canopies and merge them with the branches to obtain individual tree point clouds.

A. Data Preprocessing

Mobile mapping point cloud data includes vegetation, as well as information about the ground, buildings, road infrastructure, and other elements that require a large amount of memory [22], [23]. This information affects the accuracy of vegetation extraction and causes problems such as long data processing times. In this study, we preprocess original vehicle point cloud data using mainly point cloud denoising and preliminary semantic classification. First, the point cloud is denoised based on the statistical outlier removal filter (SOR) algorithm to remove the noise information. Furthermore, we use the RandLA-Net deep learning 3-D point cloud network for semantic classification based on [9] to classify the original point cloud into the ground, vegetation, and man-made buildings (including human-operated poles), and reserve the vegetation elements for individual tree segmentation. The results of mobile mapping point cloud data preprocessing are shown in Fig. 2.

B. Terrain Filtering and Trunk Filtering

Most studies [24], [25], [26] have used the spatial elevation section method to obtain the main stem or part of the branch point cloud of trees for use as the basic information for tree localization. However, if a uniform height threshold is used in a scene with hilly terrain, point cloud filtering tends to miss numerous tree trunks, which leads to failed localization. Our work therefore considers the influence of terrain ripple, and the

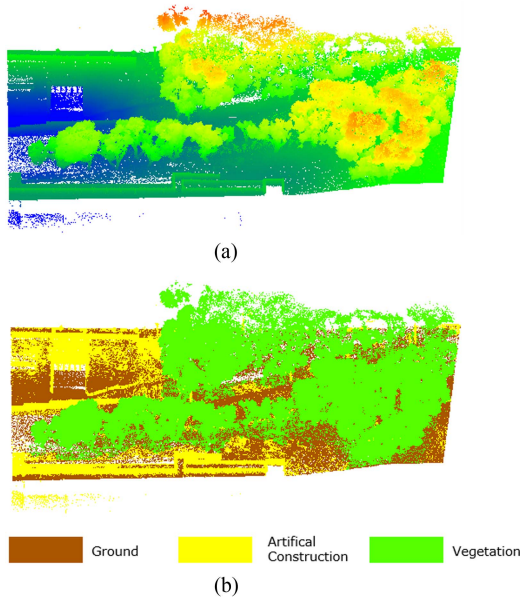


Fig. 2. Data preprocessing. (a) Raw point cloud. (b) Classification result by deep learning. (Brown: Ground points; Green: Vegetation points; Yellow: Artificial construction points).

point clouds of vegetation are unified to the same height level to eliminate this influence before height-based point cloud filtering. The details are as follows. 1) Ground points are interpolated according to the ground point cloud obtained by preprocessing to create a digital elevation model (DEM) using the inverse distance-weighted interpolation method (1), (2). The vegetation point cloud is then subtracted from the DEM elevation according to (3) to unify it to the same horizontal plane

$$Z = \frac{\sum_{i=1}^n \frac{z_i}{p_i}}{\sum_{i=0}^n \frac{1}{p_i}} \quad (1)$$

$$p_i = \left((X - x_i)^2 + (Y - y_i)^2 \right)^{\frac{q}{2}} \quad (2)$$

$$z_{i_deter} = z_i - z_{grid} \quad (3)$$

In (1) and (2), (X, Y, Z) are the coordinates of the interpolation points; $(x_i, y_i, \text{and } z_i)$ are the coordinates of the neighboring points; p is the weight; q is the power; i is the index of the neighbor points; and n is the number of neighboring points in the search area. In (3), Z_{i_deter} is the normalized height value of any vegetation point, Z_i is the original height value of the point, and Z_{grid} is the corresponding height value of the DEM grid in which the point is located. After eliminating the influence of terrain undulation, all vegetation point clouds are unified to the same horizontal plane. By iterating the vegetation point cloud data, the lowest value of elevation Z_b is determined and the elevation threshold T_h is set. As shown in Fig. 3, the vegetation point cloud is intersected based on the height filtering within $[Z_b, Z_b + T_h]$ to obtain the local trunk point cloud. Fig. 3 shows the point cloud after terrain filtering and trunk filtering.

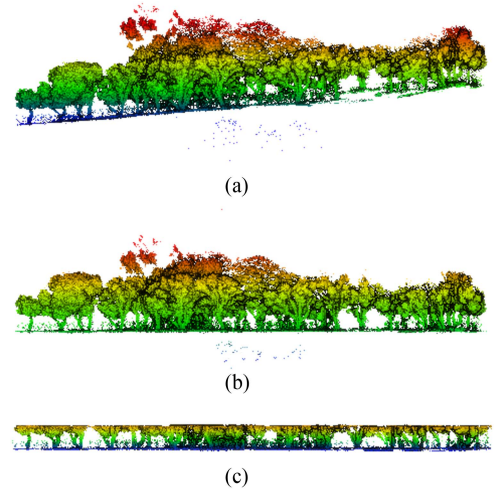


Fig. 3. Terrain filtering and trunk filtering. (a) Vegetation point cloud. (b) Vegetation point cloud after terrain filtering. (c) Vegetation point cloud after filtering with height.

C. Tree Localization

The trunk is an important feature to distinguish street trees from other objects. When investigating the individual segmentation of street trees, most studies [27], [28], [29] have tended to first identify and extract the trunk structure from the target point cloud as the basis for localization. However, the existing methods are prone to inaccurate localization of trees with complex structures, such as those with nonvertical trunks, multiple branches, and low branches. We thus propose a method to localize street trees with complex trunk geometry. The method first performs Euclidean distance clustering on the trunk point clouds obtained by filtering and then voxelizes each clustered point cloud cluster. The seed points in the voxels are subsequently selected for voxel area growth to obtain the stem candidate point clouds as constrained by the convex packet area change rate. Finally, possible low shrubs in the candidate trunks are removed by a point cloud filtering algorithm based on surface variability. Each step in this process is described in greater detail below.

1) The trunk point cloud is clustered based on the adaptive Euclidean clustering method proposed by [24]. To obtain the optimal clustering effect, the three thresholds $d(p_1, p_2)$, $d_{NN}(p_1)$, and \bar{d}_{NN} are calculated using (4)–(6) to obtain $d(p_1, p_2)$, where $d(p_1, p_2)$ represents the 3-D spatial distance between points $(p_1$ and $p_2)$, $d_{NN}(p_1)$ represents the average spatial distance from the point (p_1) to its n nearest neighbors, and \bar{d}_{NN} represents the average distance from all points in point cloud P to their nearest neighbors, which is generally used to measure the point cloud density. To obtain the fine trunk point cloud, the single cluster point cloud Z obtained after clustering is filtered, and the point cloud data less than 0.5 m from the ground are retained (as shown in Fig. 4) to obtain the point cloud cluster $PC_{S_{trunk}}$

$$d(p_1, p_2) = \sqrt{(x_{p_1} - x_{p_2})^2 + (y_{p_1} - y_{p_2})^2 + (z_{p_1} - z_{p_2})^2} \quad (4)$$

$$d_{NN}(p_1) = \min_{\{p_i \in P: p_i \neq p_1\}} d(p_1, p_i) \quad (5)$$

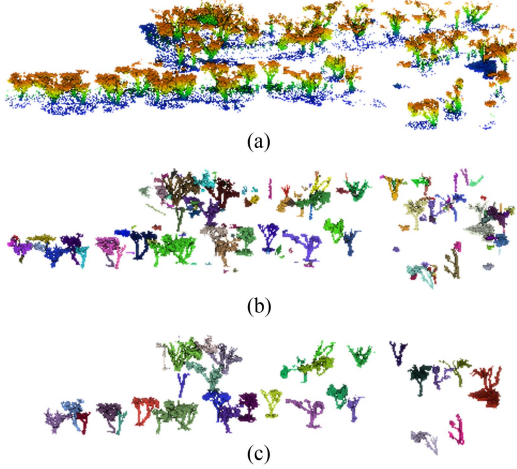


Fig. 4. Slice point cloud clustering processing. (a) Slice cloud. (b) Clustering point cloud clusters. (c) Clustered point cloud containing tree trunks.

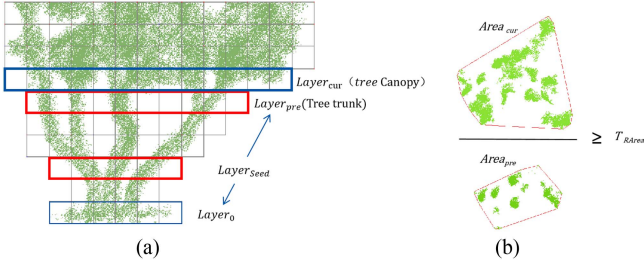


Fig. 5. Schematic diagram of the tree trunk growth process. (a) Voxelization of a candidate trunk cluster. (b) Horizontal convex hull of the trunk and crown layers.

$$\overline{d_{NN}} = \frac{1}{N(P)} \sum_{i=1}^{N(P)} d_{MN}(p_i). \quad (6)$$

2) The point cloud cluster $PC_{S_{trunk}}$ is voxelized, and all voxels are organized hierarchically from bottom to top as $Layer_0, Layer_1, \dots, Layer_n$. The layer with the lowest number of voxels is selected as the location of the stem seeds ($Layer_{seed}$). Considering that a single point cloud cluster may contain more than one stem, we cluster $Layer_{seed}$ to obtain multiple subclusters and compute the horizontal convex envelope area $Area_{CH}$ of each subcluster corresponding to the point cloud if it is smaller than the specified threshold T_{Area} , where if $T_{Area} = 1.5$, then the subcluster represents a trunk seed $seed_{Trunk}$.

3) As shown in Fig. 5(a), for each subcluster $seed_{Trunk}$, the upward and downward growth methods are used to obtain the overall trunk point cloud. The upward growth method obtains the overlapping voxels in the layer $Layer_{Seed} + 1$ with $Seed_{Trunk}$ in the X, Y plane and grows in the horizontal region of the layer such that the growth of the layer ends when no adjacent voxels are added. As seen in Fig. 5(b), the growth of the current layer is analogous to that of the previous layer. If the ratio of the horizontal convex packet area $Area_{cur}$ to the upper layer area $Area_{pre}$ is greater than T_{RArea} , then the current layer is the

canopy layer and the growth ends. The bottom of the tree trunk is reached by similar downward growth.

4) Based on the above results, the local surface variation in trunk growth P_{sv} is used to distinguish trees from shrubs. The parameters are first calculated by s (7) and (8), where $\lambda_1, \lambda_2, \lambda_3 (\lambda_1 \geq \lambda_2 \geq \lambda_3)$ are the point p_i neighborhood eigenvalues; $P_{sv}(p_i)$ is the degree of surface variation of the neighboring point cloud; and $P_{sv}(P)$ denotes the degree of local surface variation of the point cloud P . If the degree of local surface variation in this growth result exceeds the specified threshold T_{sv} , the result is considered a shrub and removed

$$P_{sv}(p_i) = \frac{\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} \quad (7)$$

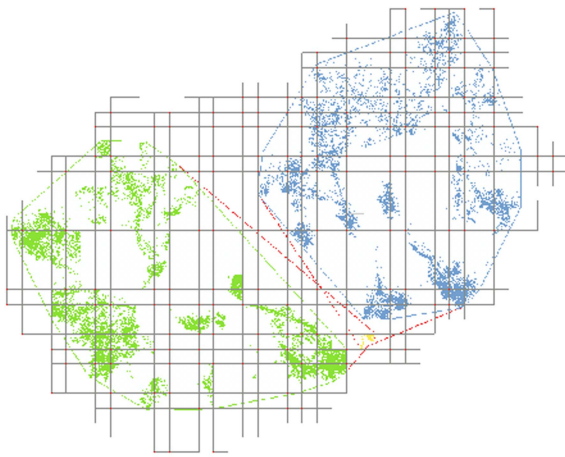
$$P_{sv}(P) = \frac{1}{N(P)} \sum_{i=1}^{N(P)} P_{sv}(p_i). \quad (8)$$

D. Coarse Extraction of Tree Crowns

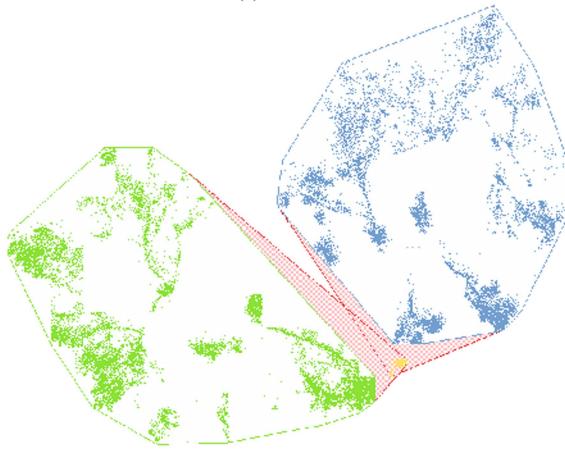
Based on the trunk information obtained in Section III-C, the tree crown is then extracted using the coarse-to-fine process. The point cloud for the tree crown is obtained by Euclidean clustering at the local region and the initial point cloud of the tree crown by region growth, after which the crown point cloud is optimized based on the ‘‘valley’’ structure as follows. 1) All trunk points are removed from the vegetation point cloud, and the trunk centroid T_{bc} is used as the seed point. The average distance D_{NeT} between this trunk and neighboring trunks is used as the initial radius to perform a Euclidean clustering operation in this region, and the resulting clustered point cloud PC_{roCd} is used as the initial candidate crown point cloud. To obtain a more complete point cloud of the tree crowns, a growth method similar to that of trunk extraction is used to further extract the tree canopies. The distance $d(P_{CroCd}^i, P_{CroCd}^j)$ between each point cloud of the candidate regions is calculated, and if $d(P_{CroCd}^i, P_{CroCd}^j) = 0$ it is assumed that the point clouds of the candidate regions P_{CroCd}^i, P_{CroCd}^j overlap with those of the candidate regions P_{CroCd}^i, P_{CroCd}^j . If there is an overlap, the overlapping region point clouds are obtained and voxelized. 2) Fig. 7(a) shows the voxelization results for two tree crowns and that the algorithm grows from the initial layer marked in blue. When two adjacent trees grow simultaneously in the same layer and a voxel is marked at the same time, the mapping is judged based on the minimum increment rule in which A_{before} and A_{after} are the horizontal convex envelope areas corresponding to the tree crown growth results before and after adding the voxel, respectively. As shown in (9) and Fig. 6, the increase in the horizontal convex parcel area of $Area_{grow}^1, Area_{grow}^2$ (10) is first calculated after adding $Tree_1, Tree_2$ to the point cloud of this voxel. If $Area_{grow}^1 < Area_{grow}^2$, the voxel goes to $Tree_1$, while if $Area_{grow}^1 > Area_{grow}^2$, it goes to $Tree_2$

$$Area_{grow} = Area_{grow}^1 - Area_{grow}^2. \quad (9)$$

Fig. 7 shows the crown growth process in two adjacent layers, where green voxels represent $Tree_1$, blue voxels represent $Tree_2$, red voxels represent the point clouds to be grown, and yellow



(a)



(b)

Fig. 6. Schematic diagram of the minimum increment rule (yellow voxels are repeat labeled voxels). (a) Schematic diagram of the change of the horizontal convex hull before and after adding repeated label voxels. (b) Schematic diagram of the increase in the area of the horizontal convex hull after adding repeated labeled voxels.

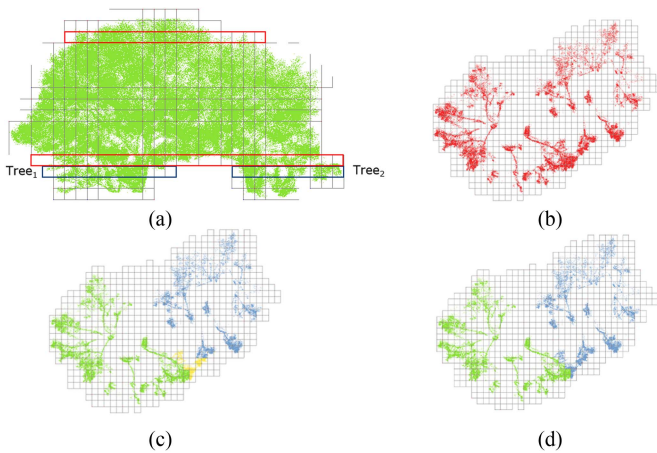


Fig. 7. Growth process of crowns in $Layer_{i+1}$. (a) Side view of voxelization results of $Tree_1$ and $Tree_2$ (b) Point cloud of the canopy to be grown in $Layer_{i+1}$ of $Tree_1$ and $Tree_2$ (c) Preliminary growth results and repeat labeling of voxels (d) Reassignment of duplicate marker voxels using the minimum increment rule.

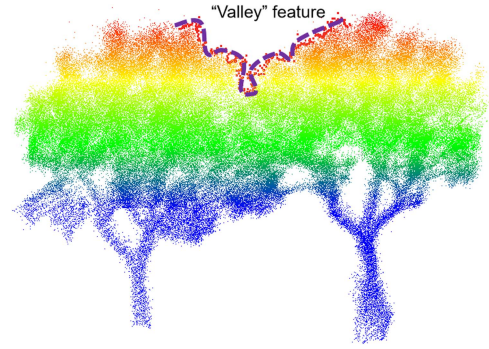


Fig. 8. Valley structure in overlapping regions.

voxels represent adherent point clouds. Fig. 7(a) shows the side view of the initial voxels of two trees, and Fig. 7(b) shows the top view of the canopy point cloud to be segmented after the upward regional growth of $Layer_{i+1}$, Fig. 7(c) shows the voxel adhesion between two trees in the current layer, and Fig. 7(d) shows the canopy point cloud of two trees obtained using the minimum increment rule.

E. Tree Crown Refinement

As shown in Fig. 10(a), the results of coarse segmentation have some degree of over- or under-segmentation due to possible size differences or excessive adhesion between adjacent trees. To further improve the tree crown segmentation results, this work optimizes adjacent tree crowns based on the “valley” structural feature such that the point cloud in the overlapping region of adjacent street trees has a height change from high to low and then low to high, as shown in Fig. 8. 1) Equation (10) calculates the deviation degree P_d of each point such that if $P_d < 0.2$, the point occurs in the point cloud of the middle region of the adherent canopy. Euclidean clustering is performed on the point cloud of this middle region, and the overlapping canopy point clouds that must be reassigned are clustered into two classes, CR_1 and CR_2 , as shown in Fig. 10(b) and (c)

$$P_d = \frac{|d_{xy}(p_i, p_{trunk_j}) - d_{xy}(p_i, p_{trunk_k})|}{d_{xy}(p_i, p_{trunk_j}) + d_{xy}(p_i, p_{trunk_k})} \quad (10)$$

where $d_{xy}(p_i, p_{trunk_j})$ denotes the 2-D distance between point p_i and the principal point p_{trunk_j} of trunk j . 2) The tree canopy point clouds CR_1 and CR_2 with a lower mean height CR_1 are used as initial points, while their edge points are used as the seed point set P_{seeds} for clustering, which is iteratively optimized by the following method.

- 1) First, the highest point p_{seed^h} of the seed point P_{seeds} in CR_x is obtained as the nearest neighbor (R-nearest neighbors, Rnn).
- 2) If the highest point $p_{already}$ of CR_f in Rnn is higher than the highest point p_{yet} in CR_x , and the lowest distance d_{min} from CR_1 to CR_2 in Rnn is small than double the average distance $\overline{d_{NN}}$ in Rnn, then the seed point is added to the clustered set CR_1 and the other points in Rnn are added as new seed points to P_{seeds} . If the above condition is not satisfied, the seed point is removed from the seed set.

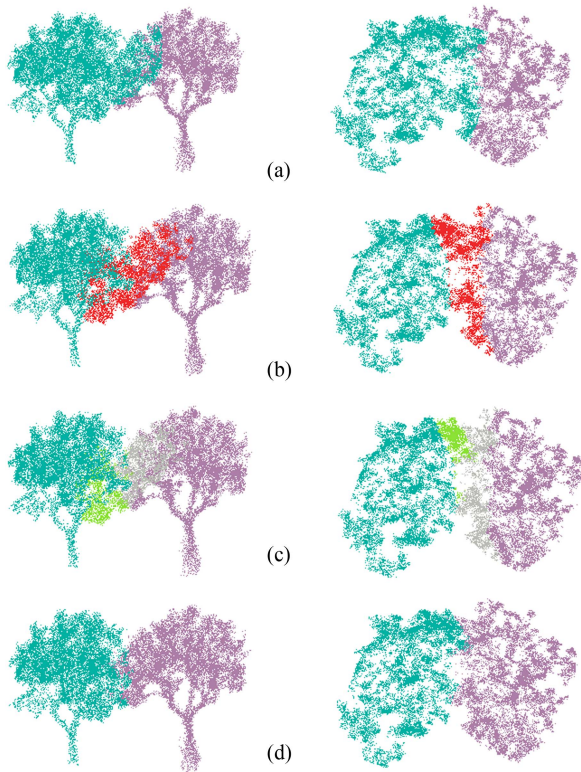


Fig. 9. Refine crown extraction pipeline. (a) Coarse crown extraction. (b) Crown point cloud to be reassigned in the middle area. (c) Clustering results (green: clustered points; grey: unclustered points). (d) Refined crown extraction.

- 3) Repeat the above steps until there are no more seed points. As shown in Fig. 9(d), the remaining unfinished labeled point clouds are merged into CR_2 to complete the tree crown optimization

When multiple street trees adhere to each other in multiple directions, we first arbitrarily select two adherent street trees for processing and then iteratively traverse each pair of adherent street trees and use the same method to complete the single canopy extraction.

IV. EXPERIMENTATION AND ANALYSIS

A. Experimental Data and Evaluation Criterion

To verify the feasibility and effectiveness of the proposed monomerization extraction method for street trees, experiments were conducted using three MLS point cloud datasets, followed by qualitative and quantitative analyses. Datasets I [see Fig. 10(a)] and II are mobile laser scanning point clouds of campus streets. The street trees in Dataset I are mostly lychee trees with complex trunk geometry, and the topographic elevation of the area is hilly. The trees in Dataset II [see Fig. 10(b)] have serious problems with canopy adhesion, and clearly identifying boundaries between canopies is difficult. These two datasets were used to test the effectiveness of the proposed method for individual segmentation of trees with complex geometric structures or strong adhesion. Dataset III [see Fig. 10(c)] is a public point cloud dataset (part of the Oakland 3-D point cloud dataset), which has a lower point cloud density than the first two

TABLE I
DETAILS OF THE EXPERIMENTAL DATASETS

Dataset	Number of Points	Density	Length (m)	Number of Trees
Dataset I	15 696 165	1323	173	69
Dataset II	47 401 751	4661	181	197
Dataset III	5 611 906	293	171	51

TABLE II
RECALL AND PRECISION OF TREE LOCALIZATION IN THREE DATASETS

Datasets	Recall(%)	Precision(%)
Dataset I	100	95.83
Dataset II	96.44	98.44
Dataset III	96.08	100

datasets. For accuracy validation, the ground truth of individual trees was obtained by manual segmentation and labeling. Table I shows a basic overview of the three experimental datasets. For quantitative analysis, we used *Recall* and *Precision* as detection and evaluation metrics for the results of the segmentation of individual trees (11). Recall represents the completeness or quantity of tree segmentation, while precision is a metric of precision or quality. The experiment was divided into two parts: tree localization and individual tree segmentation, the results of which were analyzed in comparison with those of existing tree segmentation methods

$$\begin{cases} \text{Recall} = \frac{TP}{TP+FN} \times 100\% \\ \text{precision} = \frac{TP}{TP+FP} \times 100\%. \end{cases} \quad (11)$$

B. Performance of Tree Localization

An initial classification of the entire point cloud based on the RandLA-Net point cloud classification network was performed in this work before individual tree segmentation. To achieve better classification results, we used a 3 km road point cloud obtained from Shenzhen University for RandLA-Net model training. We performed manual segmentation of the point cloud and labeled the different point clouds with vegetation, roads, buildings, and light poles. Fig. 11 shows the tree localization results for Dataset I. From Fig. 11(a), we see that deep learning semantic classification removed most of the elements that were unrelated to vegetation and maintained the vegetation information. We use a resolution of 0.05 m for the DEM. Fig. 11(b) shows the point cloud of vegetation obtained after removing the terrain and cutting the tree trunks. All the vegetation is on one plane after terrain removal, and the tree trunk information can be obtained by height filtering. Based on the filtered trunk point cloud, the proposed method performed accurate tree localization as shown in Fig. 11(c). As shown in Table II, 72 tree trunk structures were identified in Dataset I, with 69 correctly identified and 3 incorrectly identified, while $Recall = 100.00\%$ and $Precision = 95.83\%$. Fig. 12 shows the tree location results for Dataset II. Similar to Dataset I, this method yielded excellent tree localization results: 193 trunk structures were identified, with 190 correctly identified, 3 incorrectly identified, and 7 not identified. Therefore, $Recall = 96.44.00\%$ and $Precision = 98.44\%$ in Dataset II. As shown in Fig. 13, in Dataset III, 48 trunk structures were identified, all of which were done

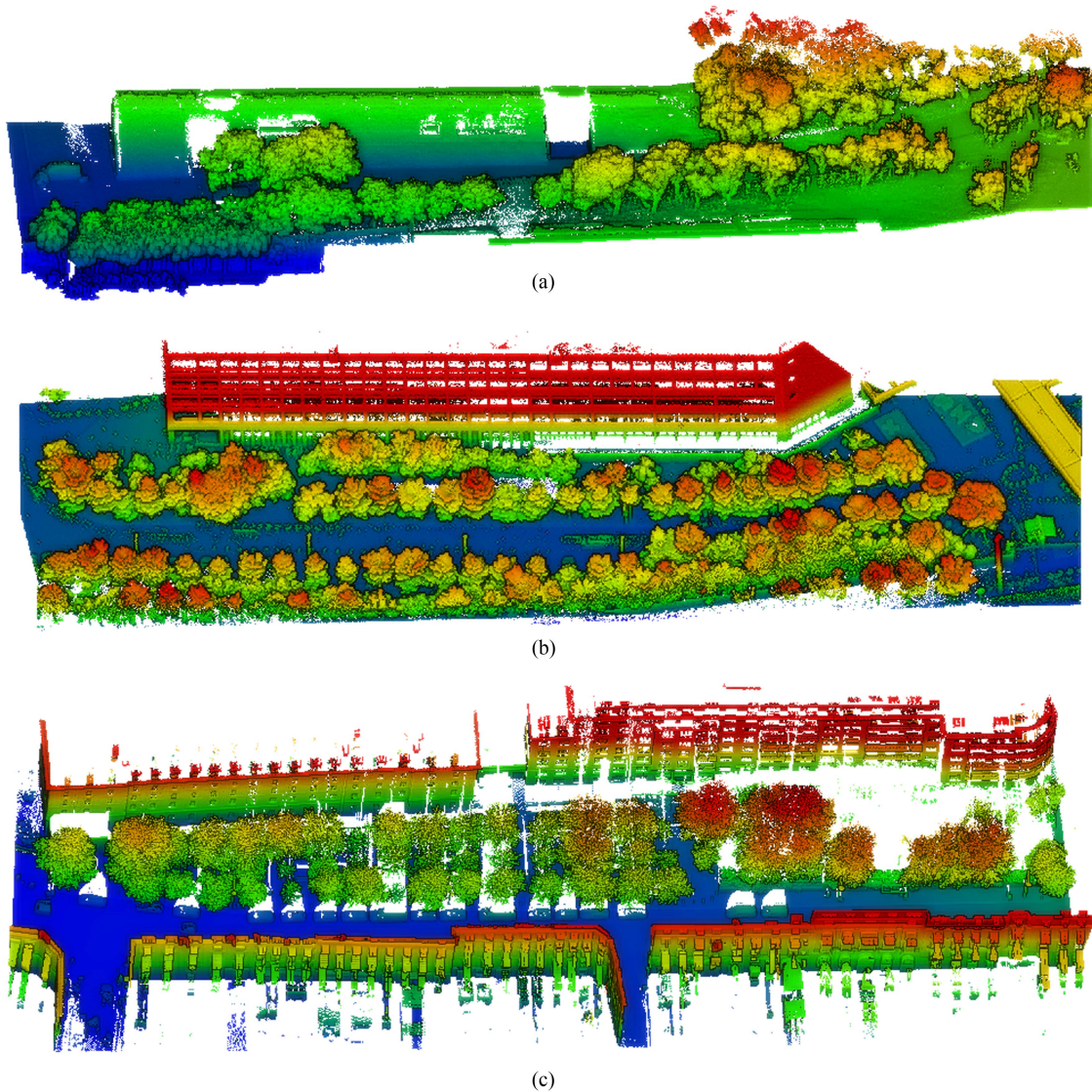


Fig. 10. Experimental datasets. (a) Dataset I. (b) Dataset II. (c) Dataset III.

correctly, but three were missed such that $Recall = 96.08\%$ and $Precision = 100\%$.

Fig. 14 shows two incorrect identification results of tree localization. Mislocalization mainly occurred because some of the point clouds of the street lights near the street trees were semantically labeled as vegetation point clouds during deep learning classification, which led to confusion between street lights and tree trunks. These street lights have a high degree of similarity with tree trunks, it is not considered in our current method. We will further optimize in our further research to improve the accuracy of trunk detection. In addition, some trees were missed during localization mainly due to the trunks being obscured by shrubs or elements such as cars, which resulted in a substantial lack of trunk point clouds. Nonetheless, this method performed well for tree localization in three datasets of different complexity, as its $Recall$ accuracy exceeded than 96% and $Precision$ exceeded 95% , thus, providing a good basis for subsequent tree crown segmentation.

C. Performance of Individual Tree Segmentation

The complete tree crown was obtained using the tree localization results and the proposed segmentation algorithm. Individual trees are obtained by merging them with the trunk point cloud. Notably, all parameters in the crown segmentation phase are computed adaptively and do not require human adjustment. We compared the accuracy of the proposed method with the existing tree monomerization algorithms TreeSeparation [30] and TreeSeg [24]. To be fair, all three methods are implemented in Windows OS and the experiments were performed on a computer with 3.6 GHz CPU and 64 GB RAM. Figs. 15–17 compare the tree crown segmentation results for the three datasets across methods.

Fig. 15 illustrates that the proposed method extracted the individual tree model well according to the tree localization results. Fig. 15(b)–(e) shows the ground truth and the segmentation results of the proposed method, TreeSeparation, and TreeSeg, respectively. The proposed tree canopy segmentation method

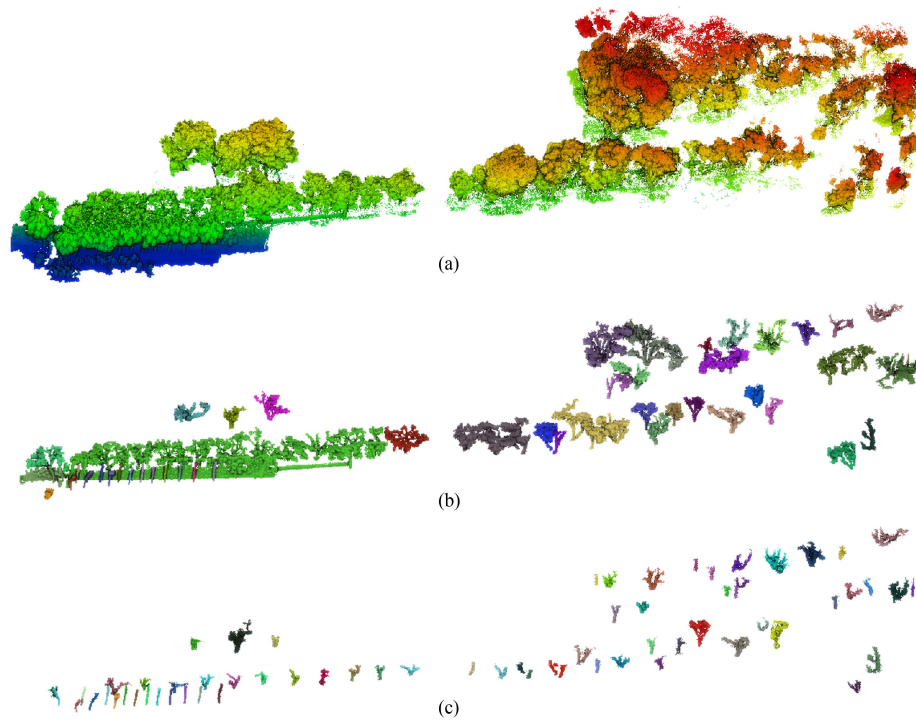


Fig. 11. Localization results of Dataset I. (a) Vegetation point cloud after deep learning classification. (b) Vegetation point cloud after terrain filtering and trunk cutting. (c) Locations of individual trees.

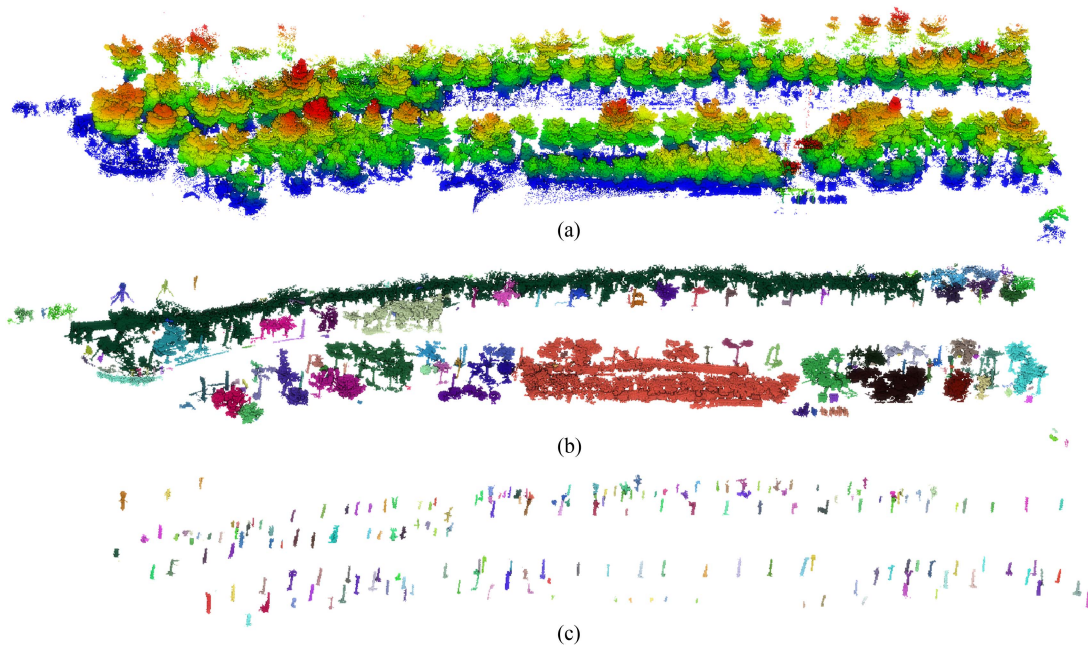


Fig. 12. Localization results of Dataset II. (a) Vegetation point cloud after deep learning classification. (b) Vegetation point cloud after terrain filtering and trunk cutting. (c) Locations of individual trees.

achieved highly precise individual tree segmentation. Compared with the results of the other two methods, the segmentation error (red point cloud of segmentation errors in (c)–(e)) is the lowest and optimal tree crown segmentation results were achieved. The TreeSeparation method achieved better individual extraction for scenes with small adhesion areas but tended to under- and over-segment when trees strongly adhered, mainly because the

method tends to discard the point cloud of the region as it obtains the individual tree by growing downward. In contrast, TreeSeg effectively extracted the main parts of street tree crowns in complex scenes with strong adhesions, but the extracted tree crowns had missing point cloud edges. To further quantify the results of individual tree extraction, we selected individual trees from different regions and calculated the precision and recall

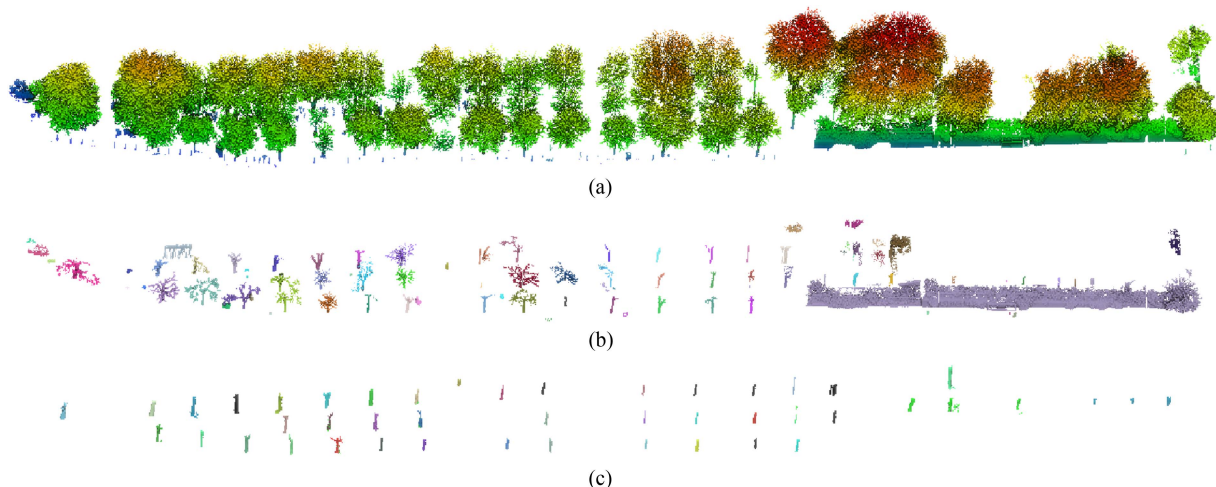


Fig. 13. Localization results of Dataset III. (a) Vegetation point cloud after deep learning classification. (b) Vegetation point cloud after terrain filtering and trunk cutting. (c) Locations of individual trees.

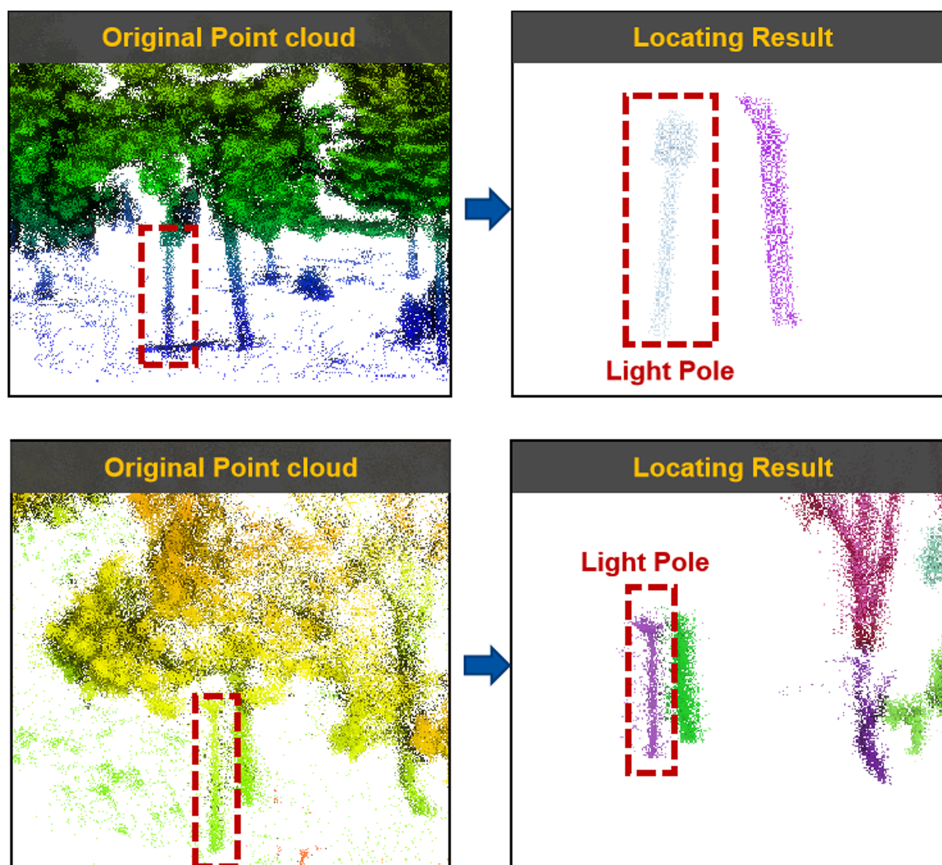


Fig. 14. Tree localization errors.

after segmenting the trees based on the ground truth. In this case, TP, FP, and FN no longer represent the number of street trees but rather the number of correct, incorrect, and missing points from individual trees, respectively. As shown in Table III, this analysis produced an average precision of 97.80% and an average recall of 97.83%. While the average precision and recall of individual

tree segmentation by TreeSeparation were 92.02% and 87.37%, respectively, TreeSeg achieved 95.24% precision and 64.85% recall. The proposed method achieved the best average recall and precision, with all trees attaining more than 93% precision and more than 94% recall. It also demonstrated good robustness for individual tree segmentation in different scenes, while the

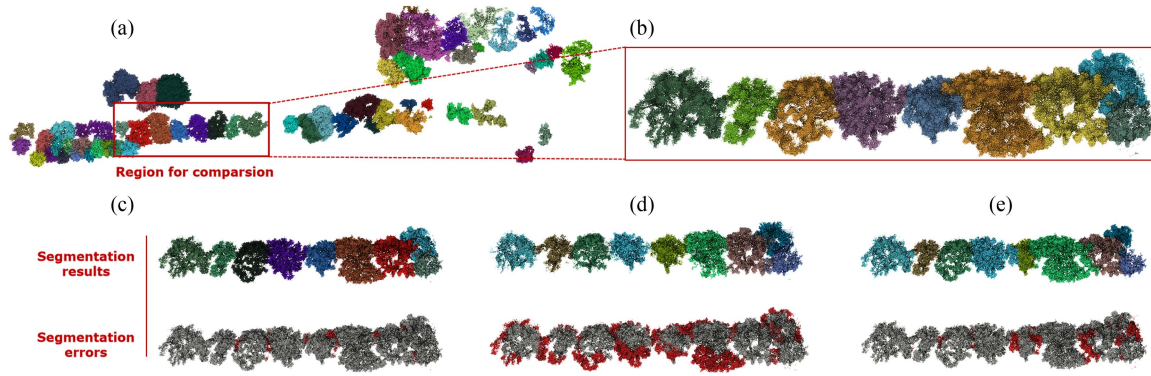


Fig. 15. Individual tree segmentation results from Dataset I. (Best accuracy in red). (a) Overall crowns segmentation results. (b) Ground truth. (c) Results from the proposed method. (d) Results from treeseg method. (e) Result from tree separation method.

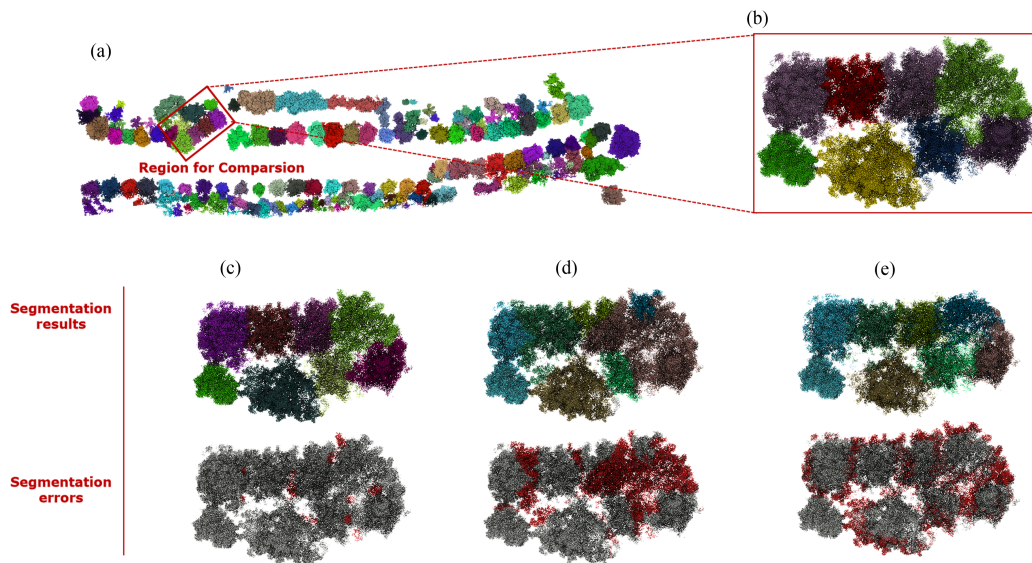


Fig. 16. Individual tree segmentation results from Dataset II. (a) Overall crowns segmentation results. (b) Ground truth. (c) Results from the proposed method. (d) Results from treeseg method. (e) Result from tree separation method.

TABLE III
QUANTITATIVE COMPARISON OF THE EXTRACTED INDIVIDUAL TREES IN DATASET I (BEST ACCURACY IN RED)

Metrics	Method	Tree index									Average
		1	2	3	4	5	6	7	8	9	
Precision	Proposed	100%	100%	93.92%	98.53%	94.76%	98.97%	97.47%	99.41%	96.24%	97.70%
	TreeSeg	99.99%	92.98%	94.85%	99.98%	91.03%	99.03%	94.26%	95.65%	89.40%	95.24%
	TreeSeparation	99.94%	99.37%	95.01%	86.93%	98.97%	80.34%	99.89%	67.18%	99.56%	92.02%
Recall	Proposed	100%	94.16%	98.35%	96.69%	99.00%	97.75%	100%	94.58%	100%	97.83%
	TreeSeg	65.55%	71.37%	66.24%	55.14%	60.72%	49.64%	59.64%	66.41%	99.02%	64.85%
	TreeSeparation	99.71%	93.19%	98.78%	98.57%	60.90%	99.50%	56.37%	81.30%	97.56%	87.37%

TreeSeparation and TreeSeg algorithms were less robust. For example, TreeSeparation achieved a segmentation precision of only 67.18% for the eighth tree, and TreeSeg achieved a recall of less than 70% for most trees. The low recall of the TreeSeg algorithm is mainly due to inconsistency between the search radius of the cylindrical filter and the real crown width of the street trees, which leads to substantial missing data in tree crown edges. Figs. 16 and 17 shows the segmentation results of individual trees

provide quantitative evaluations of the segmentation accuracy of a single tree. The proposed tree segmentation algorithm achieved optimal results across all of the experimental datasets, especially Dataset II in which optimal segmentation accuracy was achieved for all eight trees, with an average precision of 96.37% and an average recall of 97.46%. In the results of Dataset III, an average precision of 93.23% and an average recognition of 95.41% were achieved. Tables VI also shows a comparison of the time cost by the three methods for different data sets. It can be seen that the

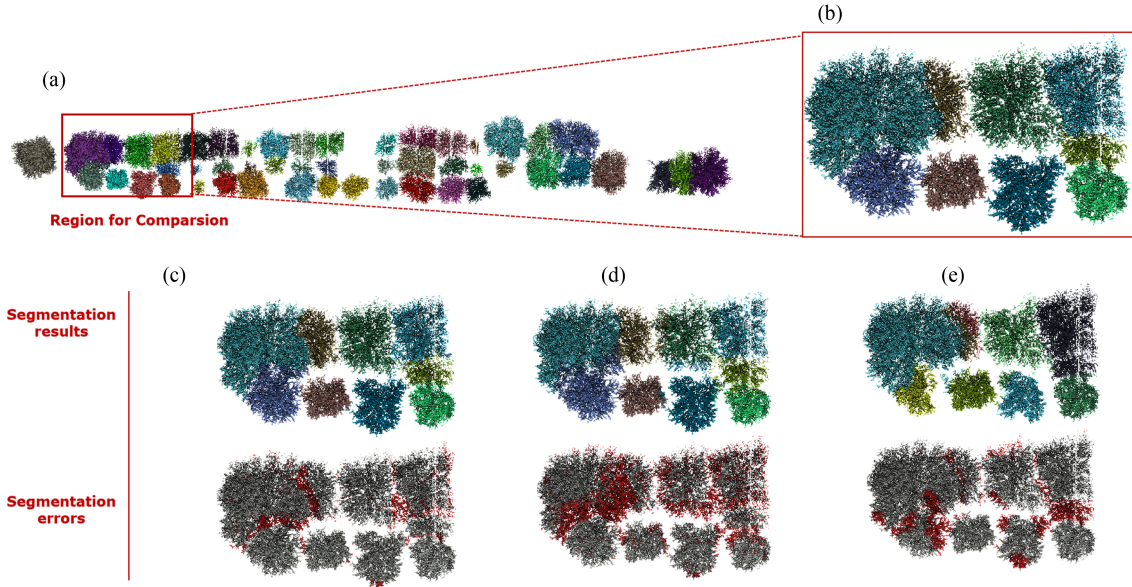


Fig. 17. Individual tree segmentation results from Dataset III. (a) Overall crowns segmentation results. (b) Ground truth. (c) Results from the proposed method. (d) Results from treeSeg method. (e) Result from tree separation method.

TABLE IV
QUANTITATIVE COMPARISON OF THE EXTRACTED INDIVIDUAL TREES IN DATASET II (BEST ACCURACY IN RED)

Metrics	Method	Tree index								
		1	2	3	4	5	6	7	8	
Precision	Proposed	98.13%	99.75%	94.31%	99.82%	98.90%	94.62%	98.79%	86.60%	96.37%
	TreeSeg	96.44%	90.83%	78.81%	96.96%	90.59%	75.85%	74.42%	74.64%	85.82%
	TreeSeparation	99.97%	97.15%	53.86%	96.47%	68.36%	91.11%	98.44%	28.28%	77.96%
Recall	Proposed	100%	99.01%	99.71%	98.31%	94.70%	98.67%	89.33%	100%	97.46%
	TreeSeg	99.69%	77.28%	82.83%	83.77%	64.38%	81.62%	60.26%	86.49%	79.54%
	TreeSeparation	95.35%	95.43%	82.67%	61.09%	29.29%	63.54%	15.47	99.77%	67.82%

TABLE V
QUANTITATIVE COMPARISON OF THE EXTRACTED INDIVIDUAL TREES IN DATASET III (BEST ACCURACY IN RED)

Metrics	Method	Tree index									Average
		1	2	3	4	5	6	7	8	9	
Precision	Proposed	99.31%	80.21%	99.44%	90.97%	83.61%	98.81%	99.36%	98.60%	90.78%	93.23%
	TreeSeg	99.30%	57.63%	96.03%	93.40%	63.33%	96.87%	92.96%	99.53%	70.18%	85.47%
	TreeSeparation	88.98%	97.38%	90.65%	70.41%	0.00%	99.09%	100.00%	99.15%	99.72%	82.81%
Recall	Proposed	89.73%	97.31%	91.68%	87.97%	97.25%	98.69%	98.82%	99.36%	97.93%	95.41%
	TreeSeg	87.89%	91.96%	86.82%	86.61%	96.57%	96.45%	98.77%	98.37%	98.26%	93.52%
	TreeSeparation	95.87%	82.75%	85.92%	99.87%	0.00%	98.33%	73.96%	99.00%	57.12%	76.99%

TABLE VI
QUANTITATIVE COMPARISON OF TIME CONSUMPTION OF THREE METHODS IN DATASETS (BEST ACCURACY IN RED)

Dataset	Time Consuming (s)		
	Proposed Method	TreeSeg	TreeSeparation
Dataset	2073	3105	1863
Dataset	6698	9850	7092
Dataset	1020	1785	1122

V. CONCLUSION

method proposed in this work achieves the highest processing efficiency on Datasets II and III. The above results illustrate that the proposed method achieves better results in terms of segmentation accuracy and robustness than the other methods.

To achieve highly precise individual segmentation of street trees, an effective individual tree extraction method with improved 3-D morphological analysis is proposed in this article. The method was optimized and improved based on existing tree localization and tree crown segmentation algorithms to achieve individual tree segmentation of a large number of street trees and effectively solve the problem of inaccurate segmentation in regions with strong tree adhesion. Three sets of MLS point clouds from different areas, which contain point clouds of trees with different geometric features, were used for experimental validation and accuracy analysis. The experimental results show that this method achieved superior segmentation accuracy for

street trees in all three datasets compared to other existing methods. The average precision and recall accuracy of the proposed method exceeded 93.23% and 95.41%, respectively. In terms of algorithm robustness, the proposed method displayed good performance on most single tree extractions, which indicates that the algorithm functions in scenes of varying complexity, achieving good robustness in cases of tree adhesion. In future research, we will optimize and improve this work in two aspects. First, the color features of the point cloud will be added to help segment different tree species. Second, the shrub rejection and clutter rejection (e.g., light poles) will be optimized and improved with the deep learning method, and the 3-D semantic point cloud segmentation method with deep learning will be optimized to further improve the effect of scene classification.

REFERENCES

- [1] L. F. Ow and S. Ghosh, "Urban cities and road traffic noise: Reduction through vegetation," *Appl. Acoust.*, vol. 120, pp. 15–20, 2017.
- [2] Z. Tan, K.K.-L. Lau, and E. Ng, "Urban tree design approaches for mitigating daytime urban heat island effects in a high-density urban environment," *Energy Buildings*, vol. 114, pp. 265–274, 2016.
- [3] J. Li, X. Cheng, Z. Wu, and W. Guo, "An over-segmentation-based uphill clustering method for individual trees extraction in urban street areas from MLS data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, no. 28, pp. 2206–2221, Jan. 2021.
- [4] C. V. Tao, "Mobile mapping technology for road network data acquisition," *J. Geospatial Eng.*, vol. 2, no. 2, pp. 1–14, 2000.
- [5] S. J. Mills et al., "Evaluation of aerial remote sensing techniques for vegetation management in power-line corridors," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 9, pp. 3379–3390, Sep. 2010.
- [6] B. Yang, Z. Wei, Q. Li, and J. Li, "Semiautomated building facade footprint extraction from mobile LiDAR point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 4, pp. 766–770, Jul. 2013.
- [7] F. Monnier, B. Vallet, and B. Soheilian, "Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 3, pp. 245–250, 2012.
- [8] Y. Li et al., "Pole-like street furniture segmentation and classification in mobile LiDAR data by integrating multiple shape-descriptor constraints," *Remote Sens.*, vol. 11, no. 24, 2019, Art. no. 2920.
- [9] Q. Hu et al., "RandLA-Net: Efficient semantic segmentation of large-scale point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11108–11117.
- [10] T.-A. Teo and C.-M. Chiu, "Pole-like road object detection from mobile LiDAR system using a coarse-to-fine approach," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 10, pp. 4805–4818, Oct. 2015.
- [11] P. Huang, Y. Chen, J. Li, Y. Yu, C. Wang, and H. Nie, "Extraction of street trees from mobile laser scanning point clouds based on subdivided dimensional features," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2015, pp. 557–560.
- [12] X. Hu and Y. Xie, "Segmentation and clustering of 3d forest point cloud using mean shift algorithms," in *Proc. 4th Int. Conf. Machinery, Mater. Comput. Technol.*, 2016, pp. 1274–1278, doi: 10.2991/icmct-16.2016.250.
- [13] W. Yan, H. Guan, L. Cao, Y. Yu, C. Li, and J. Lu, "A self-adaptive mean shift tree-segmentation method using UAV LiDAR data," *Remote Sens.*, vol. 12, no. 3, 2020, Art. no. 515. [Online]. Available: <https://www.mdpi.com/2072-4292/12/3/515>
- [14] Y. Qin, A. Ferraz, C. Mallet, and C. Iovan, "Individual tree segmentation over large areas using airborne LiDAR point cloud and very high resolution optical imagery," in *Proc. IEEE Geosci. Remote Sens. Symp.*, 2014, pp. 800–803.
- [15] J. Li, X. Cheng, and Z. Xiao, "A branch-trunk-constrained hierarchical clustering method for street trees individual extraction from mobile laser scanning point clouds," *Measurement*, vol. 189, 2022, Art. no. 110440. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0263224121013282>
- [16] J. Reitberger, C. Schnörr, P. Krzystek, and U. Stilla, "3D segmentation of single trees exploiting full waveform LiDAR data," *ISPRS J. Photogrammetry Remote Sens.*, vol. 64, no. 6, pp. 561–574, 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271609000495>
- [17] W. Yao, P. Krzystek, and M. Heurich, "Enhanced detection of 3D individual trees in forested areas using airborne full-waveform LiDAR data by combining normalized cuts with spatial density clustering," *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 1, pp. 349–354, 2013.
- [18] L. Zhong, L. Cheng, H. Xu, Y. Wu, Y. Chen, and M. Li, "Segmentation of individual trees from TLS and MLS data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 774–787, Feb. 2017.
- [19] Y. Xu, Z. Sun, L. Hoegner, U. Stilla, and W. Yao, "Instance segmentation of trees in urban areas from MLS point clouds using supervoxel contexts and graph-based optimization," in *Proc. 10th IAPR Workshop Pattern Recognit. Remote Sens.*, 2018, pp. 1–5.
- [20] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional LiDAR data for ground robot mobility," *J. Field Robot.*, vol. 23, no. 10, pp. 839–861, 2006.
- [21] L. Huo, E. Lindberg, and J. Holmgren, "Towards low vegetation identification: A new method for tree crown segmentation from LiDAR data based on a symmetrical structure detection algorithm (SSD)," *Remote Sens. Environ.*, vol. 270, 2022, Art. no. 112857. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425721005770>
- [22] Y. Yu, J. Li, H. Guan, C. Wang, and J. Yu, "Semiautomated extraction of street light poles from mobile LiDAR point-clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1374–1386, Mar. 2015.
- [23] J. Yang, Z. Kang, and P. H. Akwensi, "A skeleton-based hierarchical method for detecting 3-D pole-like objects from mobile LiDAR point clouds," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 5, pp. 801–805, May 2019.
- [24] A. Burt, M. Disney, and K. Calders, "Extracting individual trees from LiDAR point clouds using treeseg," *Methods Ecol. Evol.*, vol. 10, no. 3, pp. 438–445, 2019.
- [25] A. Jaakkola et al., "A low-cost multi-sensoral mobile mapping system and its feasibility for tree measurements," *ISPRS J. Photogrammetry Remote Sens.*, vol. 65, no. 6, pp. 514–522, 2010. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S092427161000064X>
- [26] A. Husain and R. Chandra Vaishya, "An automated approach for street trees detection using mobile laser scanner data," *Remote Sens. Appl.: Soc. Environ.*, vol. 20, 2020, Art. no. 100371. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352938519301582>
- [27] B. Yang and Z. Dong, "A shape-based segmentation method for mobile laser scanning point clouds," *ISPRS J. Photogrammetry Remote Sens.*, vol. 81, pp. 19–30, 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271613001020>
- [28] R. Zhong, J. Wei, W. Su, and Y. F. Chen, "A method for extracting trees from vehicle-borne laser scanning data," *Math. Comput. Model.*, vol. 58, no. 3, pp. 733–742, 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0895717713000034>
- [29] Z. Kang, J. Yang, R. Zhong, Y. Wu, Z. Shi, and R. Lindenbergh, "Voxel-based extraction and classification of 3-D pole-like objects from mobile LiDAR point cloud data," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 11, pp. 4287–4298, Nov. 2018.
- [30] J. Wang, R. Lindenbergh, and M. Menenti, "Voxel based scalable registration of laser scanned point cloud data by neighbourhood voting," in *Proc. Iqmulus 1st Workshop Process. Large Geospatial Data*, 2014, pp. 39–54.



Weixi Wang (Member, IEEE) received the Ph.D. degree in geodesy and surveying engineering from Liaoning Technical University, Fuxin, China, in 2007.

He was a Postdoctoral Fellow with the State Key Laboratory of Information Engineering in Surveying, Mapping, and Remote Sensing, Wuhan University, Wuhan, China, from 2007 to 2014. He is currently an Associate Professor with Shenzhen University, Shenzhen, China. His research interests include urban 3-D modeling, virtual urban environment, smart city, etc.



Yuhang Fan received the B.E. degree in transportation (highway logistics) from Henan Agricultural University, Zhengzhou, China, in 2019. He is currently working toward the M.E. degree in transportation with Shenzhen University, Shenzhen, China.

His research interests include LiDAR data processing, 3-D deep learning, urban street information extraction, etc.



Xiaoming Li (Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan, China, in 2011.

He is currently an Associate Professor with Shenzhen University, Shenzhen, China. His research interests include urban 3-D modeling and virtual urban environment construction, urban spatiotemporal big data organization management, and calculation analysis, etc.



You Li received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2017.

He was a Postdoctoral Fellow with the Research Institute for Smart Cities, Shenzhen University, Shenzhen, China, from 2017 to 2020. He is currently an Associate Researcher with Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen University. His research interests include light point cloud data target recognition, multisource laser point cloud data fusion, indoor and outdoor 3-D

reconstruction, deep learning, etc.



Shengjun Tang (Member, IEEE) received the Ph.D. degree in cartography and GIS from the State Key Laboratory of Mapping and Remote Sensing Information Engineering, Wuhan University, Wuhan, China, in 2017.

He was a Research Assistant with Hong Kong Polytechnic University, Hong Kong, from February 2014 to July 2017. He is currently an Associate Researcher with Shenzhen University, Shenzhen, China. His research interests include theory and application of visual SLAM, indoor 3-D mapping, and multisensor fusion modeling.

reconstruction, deep learning, etc.