



Original article

Coarse-to-fine segmentation of individual street trees from side-view point clouds

Qiujie Li^{*}, Yu Yan, Weizheng Li

Nanjing Forestry University, China

ARTICLE INFO

Handling Editor: Raffaele Laforteza

Keywords:

Individual tree segmentation
Side-view point cloud
Coarse-to-fine segmentation
Overlapping tree segmentation
Point cloud instance segmentation

ABSTRACT

Segmenting individual street trees from a street side-view point cloud is the first and key step of obtaining a street tree inventory. Using the classification-segmentation framework for individual tree segmentation makes tree detection simple and accurate, but segmenting overlapping trees is still challenging. To more accurately segment overlapping trees, a coarse-to-fine method for segmenting individual street trees from a side-view point cloud is proposed in this paper. Following the classification-segmentation framework, the tree points are first detected from the side-view street point cloud by a pointwise classifier fused from 13 local geometric features and then trained using random forest (RF). Second, the tree proposals are obtained by density-based spatial clustering of applications with noise (DBSCAN) clustering and detection error filtering. Third, the overlapping tree proposals are recognized by trunk identification, and the single tree proposals are directly output as individual trees. Fourth, the overlapping trees are roughly divided into individual tree proposals through vertical planes. Finally, individual trees with optimized contours are obtained by iteratively using DBSCAN clustering and k-nearest neighbor (k-NN) classification. The side-view point cloud of a 290 m-long urban street containing 77 street trees is captured by a hand-held mobile ZEB Horizon laser scanner. The tree detection attained an F_1 score of 0.9916 with a precision of 0.9989 and a recall of 0.9864. For individual tree segmentation, the F_1 score was 0.9745 with a precision of 0.9672 and a recall of 0.9819. Compared to two current classification-segmentation methods, the overlapping tree segmentation F_1 scores were increased by 0.0914 and 0.0617, respectively. The proposed method can be applied to tree parameter extraction, which is an important urban forest inventory task and is crucial for urban forest management. In our experiment, the root mean squared error (RMSE) of the trunk diameter at breast height (DBH) estimation was 0.8485 cm.

1. Introduction

Street trees are an important component of urban landscape systems and ecosystems. Street trees enhance urban liveability by reducing stormwater runoff (Gotsch et al., 2018; Szota et al., 2019), improving air quality (Lai and Kontokosta, 2019; Miao et al., 2022), storing carbon (Havu et al., 2022; Kim and Jo, 2022), providing shade (Igoe et al., 2020; Jareemit and Srivanit, 2022), improving urban heat island effects (Huang et al., 2021; Wang and Akbari, 2016) and enhancing biodiversity (Anderson et al., 2023; Liu and Slik, 2022). Street trees are also beneficial for improving the physical and mental health of urban residents (Kabisch et al., 2021; Zhou et al., 2020).

Street tree inventories that measure individual tree parameters such as tree height, crown width, crown volume, gap fraction, trunk diameter at breast height (DBH), and tree species are an important part of the

urban forest inventory system and are crucial for urban forest management (Ma et al., 2021; Zhu et al., 2023). Conventional field assessments are laborious, expensive and cannot reflect the dynamic changes in street trees in a timely and accurate manner (Galle et al., 2021).

Street view (SV) is a new but increasingly popular tool for urban greenery research (Lu et al., 2023; Wang et al., 2018). Street view (SV) images are photographs (including panorama photographs) taken at the street level. Tree parameter data are collected from these images by visual interpretation, generally by professionals or trained personnel, to maximize accuracy. However, several significant limitations exist, including georegistration difficulties, limited image resolutions, and image distortion caused by illumination variations (Branson et al., 2018; Lu et al., 2023). Consequently, errors inevitably occur in SV-based tree parameter measurement.

Light detection and ranging (LiDAR) technologies can capture three-

^{*} Corresponding author.E-mail address: lqj@njfu.edu.cn (Q. Li).

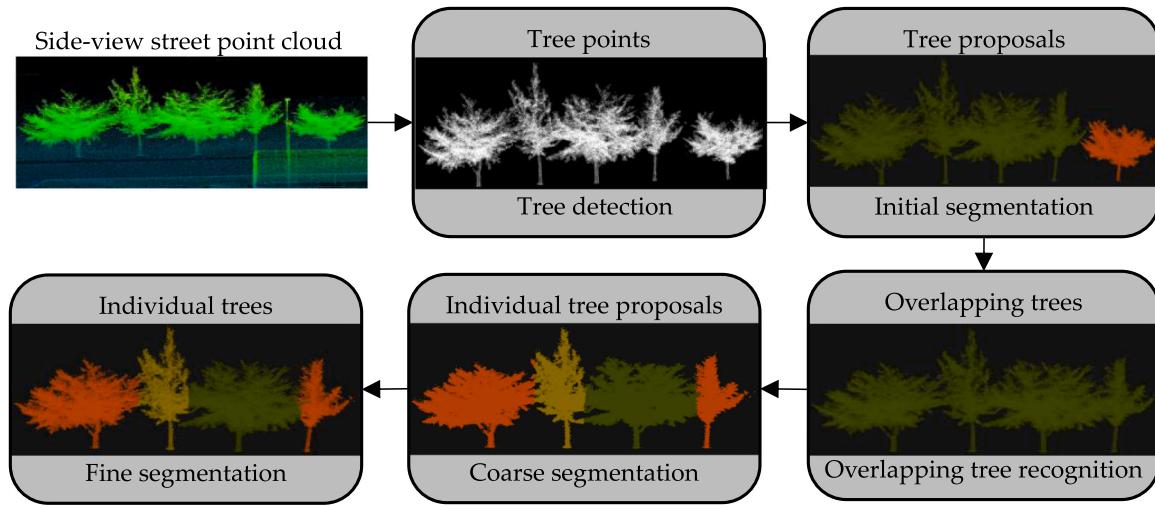


Fig. 1. Flowchart of the proposed method. First, the tree points are detected from the side-view street point cloud by a pointwise classifier. Then, the tree proposals are obtained by an initial segmentation of the tree points, among which the overlapping tree proposals are recognized for further segmentation and the single tree proposals are directly output as individual trees. Next, the overlapping trees are roughly divided into individual tree proposals through a coarse segmentation. Finally, the individual trees with optimized contours are obtained through a fine segmentation.

dimensional (3D) point clouds of the surrounding object surfaces (Raj et al., 2020). Points clouds achieved by state-of-the-art LiDAR sensors capture object details with good accuracy and precision, and the LiDAR-based street tree inventory process has become a research hotspot. LiDAR data are not affected by natural illumination conditions and thus can be gathered at any time of day or night. The morphological parameters of trees extracted from LiDAR point cloud data are very close to the measured values (Sun et al., 2022). Mobile laser scanning (MLS), e.g., vehicle-borne laser scanning and hand-held laser scanning, can dynamically collect high-precision side-view point clouds of all street trees on a street in a short period (Jiang et al., 2022; Wang et al., 2019). MLS enables the vertical structures of individual trees to be accurately extracted and is suitable for a wide range of street tree inventories (Wu et al., 2013).

Segmenting individual street trees from a street side-view point cloud is the first and key step of creating a street tree inventory. The parameters are then measured from the point clouds of individual trees. The individual tree segmentations become 3D instance segmentation problems containing two tasks: detecting the street tree points and distinguishing the points belonging to different trees. Because street scenes contain a variety of ground objects, traditional approaches gradually extract and filter nontree point clouds using prior knowledge about the poses and geometric characteristics of the trees and nontree objects (Li et al., 2021, 2022; Safaei et al., 2021). This process is as follows. First, the ground points are removed through horizontal plane detection; then, the individual objects are obtained through hierarchical point cloud segmentation methods such as voxelization, oversegmentation, region growing and clustering; finally, individual street trees are distinguished by extracting typical object poses and structures.

Traditional approaches require artificially designing segmentation features and rules to describe and distinguish street trees and other objects, which is a complex task due to the diversity of street scenes. Weinmann et al. (2017) proposed a classification-segmentation framework for individual tree segmentation. In this framework, the street point cloud is first divided into two point clouds, one containing tree points and one containing nontree points, using a pointwise classifier; then, the point cloud containing only trees is further divided into individual trees. The pointwise classifier is composed of a set of local geometric features extracted from the neighborhood of the point to be classified. The classification-segmentation method makes tree detection simpler and more accurate than traditional methods. It must design only a set of common low-level features, such as the coordinates' means and

standard deviations. Then, a supervised learning algorithm is used to automatically learn the differences between the tree points and nontree points from a labeled training set and fuses these features to a high-accuracy classifier. After extracting the tree point cloud, Weinmann et al. (2017) segmented individual trees on the horizontal projection of the tree points using mean-shift. Following the above framework, Li et al. (2020) and Hua et al. (2022) proposed methods for individual tree segmentation. Focusing on the MLS data captured by a 2D LiDAR, Li et al. (2020) trained crown and trunk detectors using discrete AdaBoost. The numbers of crowns and trunks per LiDAR scanline were counted, and the individual trees were segmented at the scanline level. Hua et al. (2022) detected tree points by a support vector machine (SVM) classifier. Then, the tree point cloud was projected onto a vertical plane, and the individual trees were segmented using Canny edge detection and snake contour extraction. Generally, current classification-segmentation methods segment tree points into individual trees in a two-dimensional (2D) projection image or at the scanline level instead of in a 3D point cloud. This process introduces segmentation errors, especially when trees overlap.

Following the classification-segmentation framework, a coarse-to-fine method for segmenting individual street trees from a side-view point cloud is proposed in this paper. To more accurately segment overlapping trees, the following research contributions are made in this paper:

- (1) After tree detection and initial segmentation, the overlapping trees are identified for further segmentation.
- (2) The overlapping trees are roughly segmented into individual trees by vertical planes.
- (3) The individual tree contours are improved by iteration optimization.

2. Materials and methods

Fig. 1 shows the flowchart of the proposed method. It contains five main steps: tree detection, initial segmentation, overlapping tree recognition, overlapping tree coarse segmentation and overlapping tree fine segmentation. First, the tree points are detected from the side-view street point cloud by a pointwise classifier. Then, the tree proposals are obtained by an initial segmentation of the tree points, among which the overlapping tree proposals are recognized for further segmentation, and the single tree proposals are directly output as individual trees. Next, the



Fig. 2. Study area (Google Maps, 32°04'55.1"N, 118°48'58.8"E) which is a 290 m-long urban street containing many species of street trees, including *Cerasus yedoensis*, *Ginkgo biloba*, *Celtis sinensis* and *Cinnamomum camphora*.

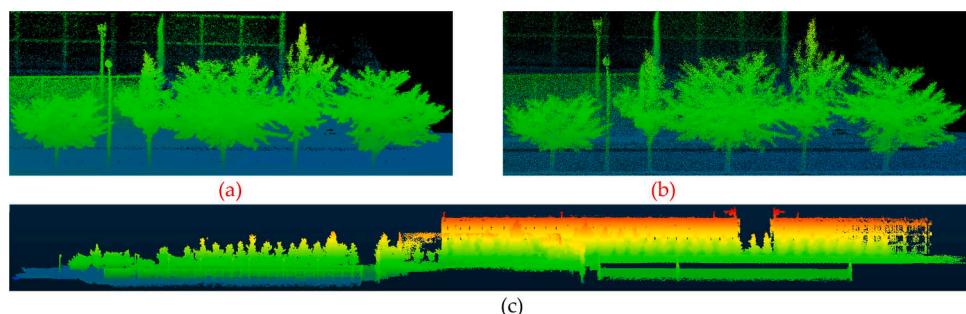


Fig. 3. Street side-view point cloud colorized by elevation values (z coordinates): partial point cloud before (a) and after FPS (b) and entire point cloud after FPS (c).

overlapping trees are roughly divided into individual tree proposals through coarse segmentation. Finally, the individual trees with optimized contours are obtained through fine segmentation.

2.1. Study area and data preparation

ZEB Horizon, a hand-held mobile laser scanner developed by Geo-SLAM, is used to collect the side-view point cloud of the street trees. ZEB Horizon contains simultaneous localization and mapping (SLAM) technology that can address the absence of a global navigation satellite system (GNSS) signal or a poor signal under the forest canopy, making it practical for forest investigations (Sofia et al., 2021). The field of view (FOV) is 360°(H) × 270°(V) with an angular resolution of 0.2°(H) × 2°(V), and it obtains 300,000 scanner points per second. With a range of 100 m and a relative accuracy of up to 6 mm, ZEB Horizon is ideal for outdoor use and provides accurate scans and fine details.

Fig. 2 depicts the study area, a 290 m-long urban street (Google Maps, 32°04'55.1"N, 118°48'58.8"E) containing many species of street trees, including *Cerasus yedoensis*, *Ginkgo biloba*, *Celtis sinensis* and *Cinnamomum camphora*.

To improve the processing efficiency, the raw point cloud collected by ZEB Horizon is downsampled to 10,000,000 points by the farthest point sampling (FPS) algorithm. FPS is an iterative process in which the points to be retained are sequentially selected from the raw point cloud until the number of selected points reaches the set value. During an FPS iteration, the point farthest from the selected points is selected from the remaining point cloud. The sampled point cloud is thus more evenly distributed and covers as much of the raw point cloud as possible. Fig. 3 shows the street side-view point cloud colorized by elevation values (z coordinates); this point cloud contains trees, buildings, lanes, sidewalks, traffic signs, streetlamps, benches, dustbins, grids, cars, bicycles, pedestrians, bushes, turfs and so on.

To train the tree detector and to evaluate the proposed individual tree segmentation method, the individual street trees in the point cloud are annotated by CloudCompare, which is an advanced open-source 3D data processing software. Fig. 4 shows annotated individual trees from the top view. This study area includes 77 street trees with heights of 4.6–8.2 m and crown widths of 2.1–7.4 m. The total number of tree points is 4,030,699, accounting for 40.31 % of the entire point cloud.



Fig. 4. Annotated individual trees displayed from a top view. There are 77 street trees with heights of 4.6–8.2 m and crown widths of 2.1–7.4 m.

Table 1

Local features extracted from a point neighborhood to construct a pointwise tree detector.

Name	Formulae
Elevation	z
Elevation difference	$\Delta z = \max_{n=1,\dots,N} z_n - \min_{n=1,\dots,N} z_n$
Elevation standard deviation	$\sigma_z = \sqrt{\frac{1}{N-1} \sum_{n=1}^N (z_n - m_z)^2}, m_z = \frac{1}{N} \sum_{n=1}^N z_n$
Verticality	$V = 1 - \mathbf{n}_z$
Density	$d = \frac{3N}{4\pi r^3}$
Linearity	$L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1}$
Planarity	$P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1}$
Sphericity	$S_\lambda = \frac{\lambda_3}{\lambda_1}$
Omnivariance	$O_\lambda = \sqrt[3]{\prod_{i=1}^3 \lambda_i}$
Anisotropy	$A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1}$
Eigenentropy	$E_\lambda = -\sum_{i=1}^3 \lambda_i \ln \lambda_i$
Sum of eigenvalues	$\Sigma_\lambda = \sum_{i=1}^3 \lambda_i$
Local surface variation	$C_\lambda = \frac{\lambda_3}{\Sigma_\lambda}$

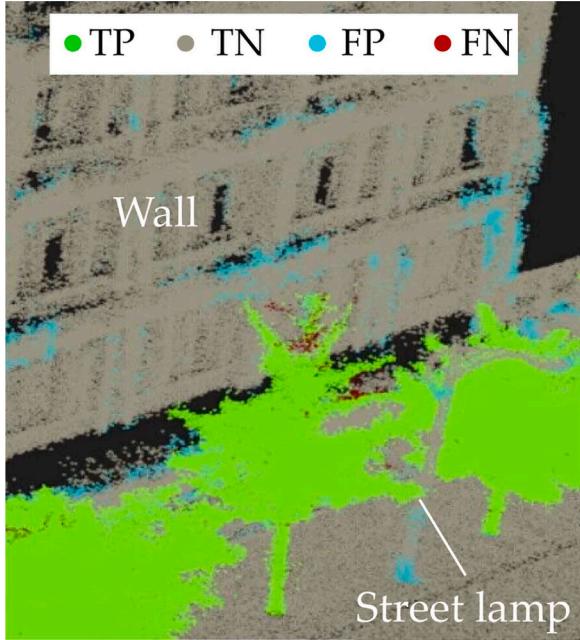


Fig. 5. The four detection types based on the consistency between the prediction and truth classes of a point. There are two misclassification types, FP and FN, because the detection method only utilizes the local information. FPs are mainly uneven walls and streetlamps, in which the local geometric features are similar to trees. FNs are mainly located at the tops of the trees.

2.2. Tree detection

To construct a pointwise tree detector, the local features from the point neighborhood are extracted first. These features are summarized in [Table 1](#). Assuming that there are N neighbors in the sphere with a radius $r = 0.5$ m centered around a point (x, y, z) to be identified, a total of 13 features are extracted from the spherical neighborhood to recognize whether each point belongs to a street tree.

The x and y coordinates describe the ground position of the point, which lacks discrimination when used alone. The point's elevation z , the elevation difference Δz and elevation standard deviation σ_z are computed to help filter out points that are overly high or low and that are on horizontal planes. Considering that urban street environments are characterized by an aggregation of man-made objects, which typically exhibit almost perfectly vertical structures (e.g., building facades, streetlamps, traffic signs), the verticality V , relying on the z component of neighborhood normal vector \mathbf{n}_z , is used as a feature. The density d , represented by the ratio of the number of neighbors and the volume of the neighborhood is used to filter out the points in the area where the local point cloud is too dense or sparse. The normalized eigenvalues λ_1 , λ_2 and λ_3 ($\lambda_1 \geq \lambda_2 \geq \lambda_3$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$) of the 3D covariance matrix of the neighbors describe the neighbors' overall distribution characteristics. Eight shape features are derived from the normalized eigenvalues to explore the local 3D geometrical characteristics: linearity L_λ , planarity P_λ , sphericity S_λ , omnivariance O_λ , anisotropy A_λ , eigenentropy E_λ , sum of eigenvalues Σ_λ and local surface variation C_λ .

Random forest (RF), one of the best supervised learning algorithms, is used to fuse the 13 low-level features into a highly accurate tree detector by learning from a labeled training set. RF integrates multiple decision trees, which are independently trained on bootstrap samples of the training set, into a forest. In the prediction, the predicted values of all the decision trees are averaged to obtain a more accurate prediction value.

There are four detection types, which are based on the consistency between the prediction and truth classes of a point, as shown in [Fig. 5](#). A correctly detected tree point is denoted as a true positive (TP), while a correctly classified nontree point is denoted as a true negative (TN). There are two misclassification types because the detection method utilizes only local information. If a nontree point is wrongly detected as a tree point, it is denoted as a false positive (FP). If a tree point is not detected, it is denoted as a false-negative (FN). FPs are mainly on uneven walls and streetlamps, in which the local geometric features are similar to trees. FNs are mainly located at the tops of the trees. These misclassified points will be further corrected in the next step ([Section 2.3](#)).

2.3. Initial segmentation

After tree detection, a density-based clustering algorithm, namely, the density-based spatial clustering of applications with noise (DBSCAN) ([Ester et al., 1996](#)), is adopted to cluster the detected tree point cloud into disjoint tree proposals. A tree proposal contains an individual tree or several overlapping trees. DBSCAN requires two parameters, the neighborhood radius ε and the minimum number of neighbors N_{min} . If the number of neighbors of a point in a sphere with radius ε is greater than N_{min} , the point and its neighbors are considered to belong to the

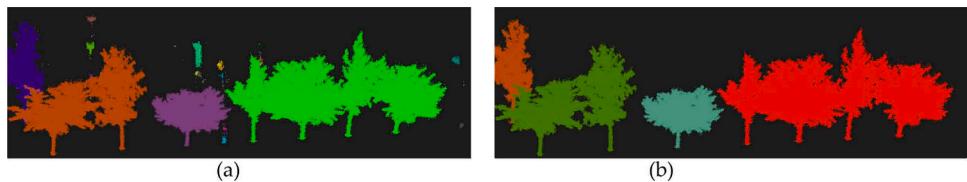


Fig. 6. Initial segmentation of the tree points: disjoint tree proposals generated by DBSCAN (a) and tree proposals after false alarm removal and k -NN reclassification (b).

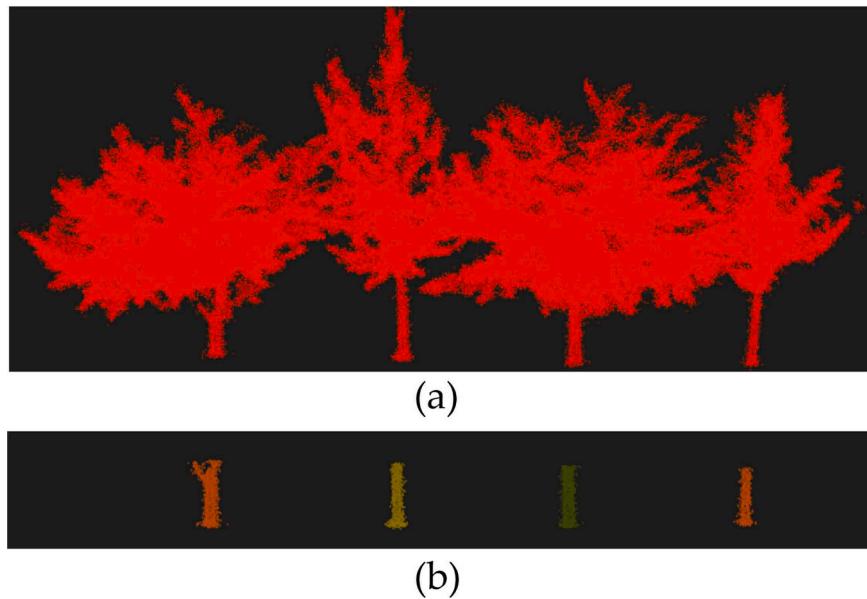


Fig. 7. Tree proposal containing four overlapping trees (a) and their segmented trunks (b).

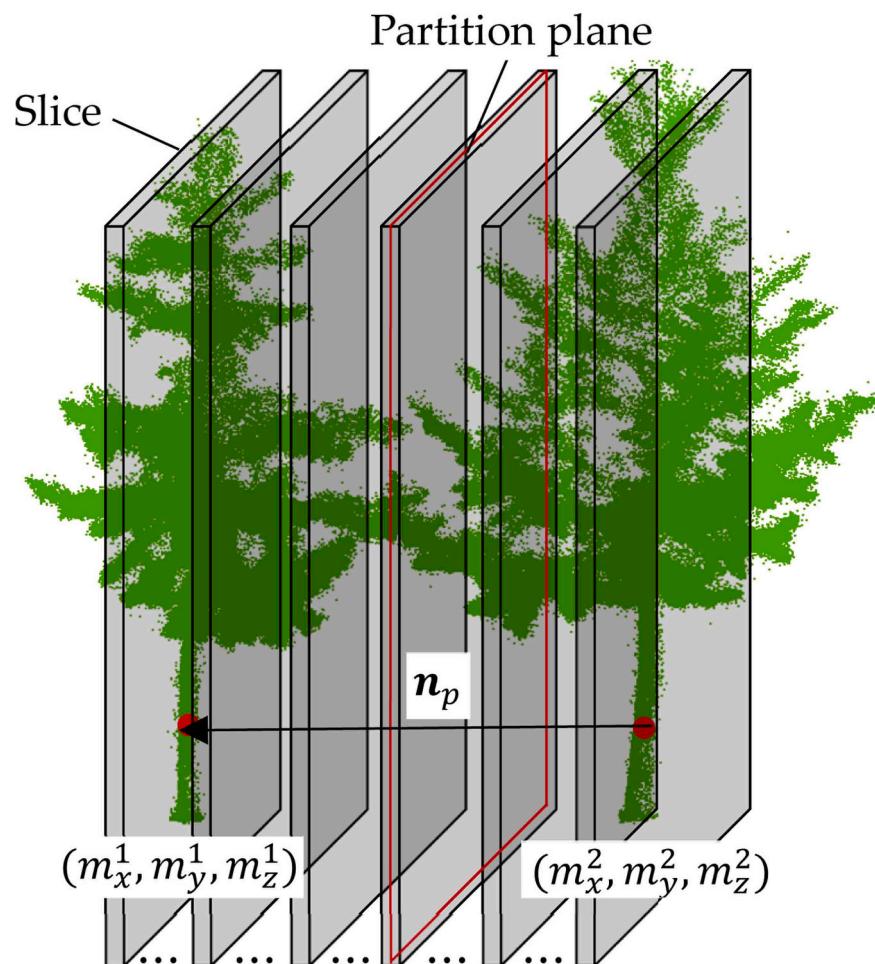


Fig. 8. Partition plane of two overlapping trees. The space between two trunks is cut into slices with a normal vector \mathbf{n}_p , and with a width of 1 cm, where \mathbf{n}_p is the xy-plane projection of the vector connecting trunk centroids (m_x^1, m_y^1, m_z^1) and (m_x^2, m_y^2, m_z^2) . The partition plane is the central plane of the slice with the smallest number of tree points.

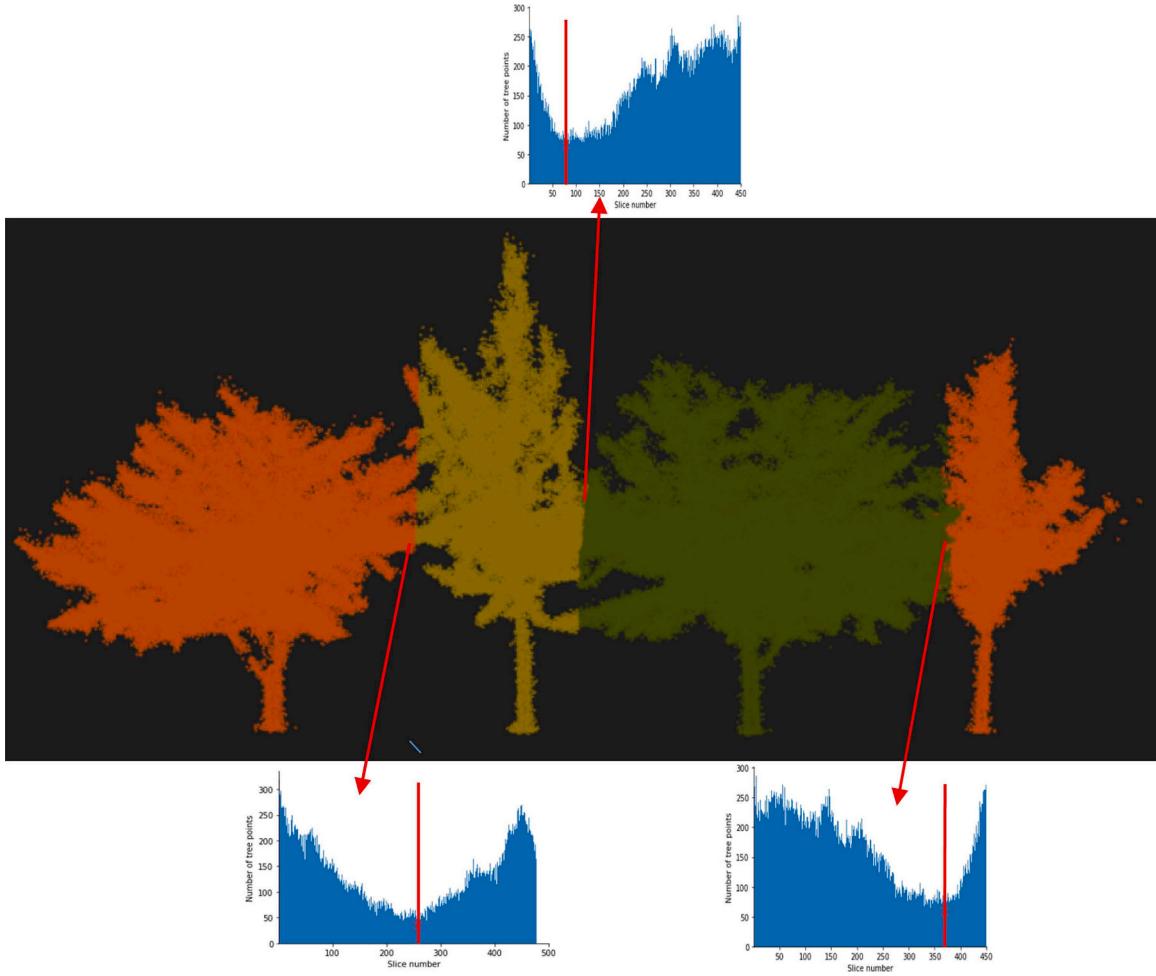


Fig. 9. Coarse segmentation of four overlapping trees. The number of tree points in the slices between two adjacent trees are displayed in the form of a histogram. The red line represents the location of the partition plane, which is centered in the slice with the smallest number of tree points.

same cluster. Due to the presence of holes inside crowns and sparse branches and leaves at the crown edges, the distribution of the crown points is relatively sparse. In our method, ϵ is set to 0.3 m, and N_{\min} is set to 1.

After clustering, further processing is conducted to reduce the classification errors generated in the tree detection step. First, considering the significant elevation differences in the trees, clusters with an elevation difference less than 4 m are removed to filter the false alarms, which are small aggregates typically caused by uneven walls and streetlamps. Then, a multiclass k -nearest neighbor (k -NN) classifier is used to reclassify points with an elevation greater than 6 m to optimize the tree detection accuracy. The number of classes is the number of tree proposals plus one; each tree proposal is assigned a class label, and the nontree points are assigned a class label. A point is reclassified as the class with the highest number of points among its k nearest neighboring points. Here, k is set to 5. Through reclassification, most missed detection points at the tops of the trees can be rediscovered. Fig. 6 shows the initial segmentation of the tree points.

2.4. Overlapping tree recognition

The overlapping tree proposals are identified for further segmentation, while the single tree proposals can be directly output as individual trees. Considering that the trunks of the overlapping trees are not connected with each other, the individual trunks are segmented and counted to recognize overlapping trees. First, points in a tree proposal with an elevation difference from the lowest point in the proposal of less than

1.4 m are extracted as trunk points. Then, DBSCAN is used to cluster the trunk points into individual trunks. The distances between the different trunks are relatively large; therefore, ϵ is set to a small value of 0.1 m, and N_{\min} is set to 1. If there is only one trunk in a tree proposal, the tree proposal is assigned a tree number and directly output as an individual tree. Otherwise, the tree proposal is assumed to contain several overlapping trees and must be further segmented. Fig. 7 shows a tree proposal containing four overlapping trees with their trunks segmented.

2.5. Overlapping tree coarse segmentation

The vertical planes between the adjacent trees are used to roughly divide the overlapping tree proposal into several individual tree proposals. The normal vector and position of these partition planes must be determined. First, the centroid coordinates of each trunk in the overlapping tree proposal are calculated. The partition plane of two overlapping trees is perpendicular to the xy -plane projection of the line determined by the centroids of their trunks. Assuming that the centroid coordinates of two adjacent trunks are (m_x^1, m_y^1, m_z^1) and (m_x^2, m_y^2, m_z^2) , the normal vector of the partition plane is

$$\mathbf{n}_p = \begin{bmatrix} m_x^1 - m_x^2 \\ m_y^1 - m_y^2 \\ 0 \end{bmatrix} \quad (1)$$

Then, the space between two trunks is cut into slices with \mathbf{n}_p as the normal vector and a width of 1 cm. The slices are sequentially numbered

starting from 1, and the number of tree points contained in each slice are counted. The central plane of the slice with the smallest number is used as the partition plane. Fig. 8 shows the partition plane of two overlapping trees.

Fig. 9 shows the coarse segmentation of four overlapping trees. The interlaced branches at the intersection of two trees could not be effectively separated using only a plane, requiring further fine segmentation.

2.6. Overlapping tree fine segmentation

A fine segmentation algorithm combining DBSCAN and the k -NN classifier is proposed for optimizing the contours of the individual tree proposals generated after coarse segmentation. The algorithm has an iterative process with two steps that segment the tree bodies and contours, as shown in Algorithm 1. Assuming N_t trees are present in an overlapping tree proposal, the segmentation result can be represented by $\{T_i\}_{i=1}^{N_t}$, in which T_i denotes the point set belonging to the i -th tree. After initializing $\{T_i\}_{i=1}^{N_t}$ with the individual tree proposals obtained by coarse segmentation, the fine segmentation steps are as follows:

- (1) Body segmentation. For each tree set T_i , DBSCAN is used to cluster the points. Here, $\epsilon = 0.15$ m and $N_{\min} = \frac{20}{t}$, where t is the number of iterations. As the number of iterations increases, the connection tightness requirement for clustering decreases, which can improve the segmentation for branch points at the intersection of two trees. Then, the cluster with the largest number is considered the main body of the i -th tree and retained as T_i . Meanwhile, the other clusters are considered branch tips that may belong to other trees and are placed in an unlabeled point set U .
- (2) Contour segmentation. A multiclass k -NN classifier is used to label the points in U . Let $\{T_i\}_{i=1}^{N_t}$ be the training set and the number of classes be N_t . To classify a point, its k nearest neighbors in the training set are selected, and the point is assigned to the class with the highest number of neighbors. Here, k is set to 11.
- (3) Terminating condition. If the segmentation results $\{T_i\}_{i=1}^{N_t}$ obtained from two consecutive iterations are the same, the process ends.

Algorithm 1. Overlapping tree fine segmentation.

Input: Tree set $\{T_i(0)\}_{i=1}^{N_t}$, where $T_i(0)$ denotes the point set belonging to the i -th tree after overlapping tree coarse segmentation. Iteration count $t=0$.

Output: Tree set $\{T_i(t)\}_{i=1}^{N_t}$.

- ```

1: repeat
2: $t=t+1$, unlabelled point set $U = \emptyset$.
3: for each i
4: Apply DBSCAN to $T_i(t-1)$. Denote the cluster with the largest number of points as M .
5: $U \leftarrow U \cup (T_i(t-1) - M_i)$, $T_i(t) \leftarrow M$.
6: for each point P in U
7: Find its k nearest neighbours in $\{T_i(t)\}_{i=1}^{N_t}$
8: $C_{i'} \leftarrow P$, where i' is the class with the highest number of neighbours.
9: for each i
10: $T_i(t) \leftarrow T_i(t) \cup C_i$
11: until $\{T_i(t)\}_{i=1}^{N_t} = \{T_i(t-1)\}_{i=1}^{N_t}$

```
- 

Fig. 10 shows the overlapping tree fine segmentation process. Taking the individual tree proposals generated and using overlapping tree coarse segmentation as input, the optimization process is completed after two iterations. Three representative inaccuracies in the tree tips are marked with circles in the input. All the inaccuracies are improved in the first iteration. Some erroneous segmentation points remain and are marked by the blue circle; these points are corrected in the second iteration.

## 3. Results and discussion

### 3.1. Tree detection results

The free points are first extracted from the street side-view point cloud by a pointwise detector and clustered into a set of tree proposals.

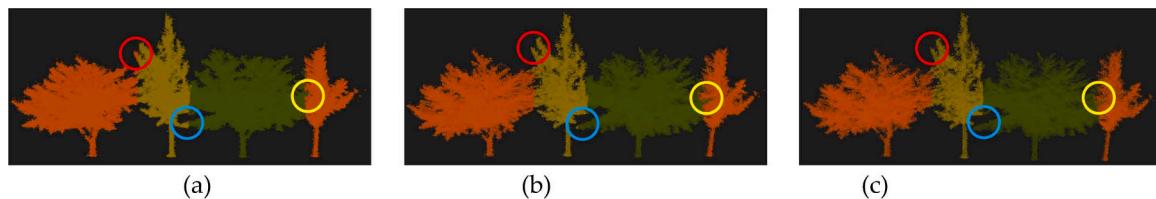
For tree detection, 10 % of the points are randomly selected to construct a training set, and the remaining 90 % are used for testing. Precision, recall, and the  $F_1$  score are used to evaluate the detection accuracy of the pointwise tree detector trained by RF. The three evaluation measurements are calculated by

$$\text{Pr} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

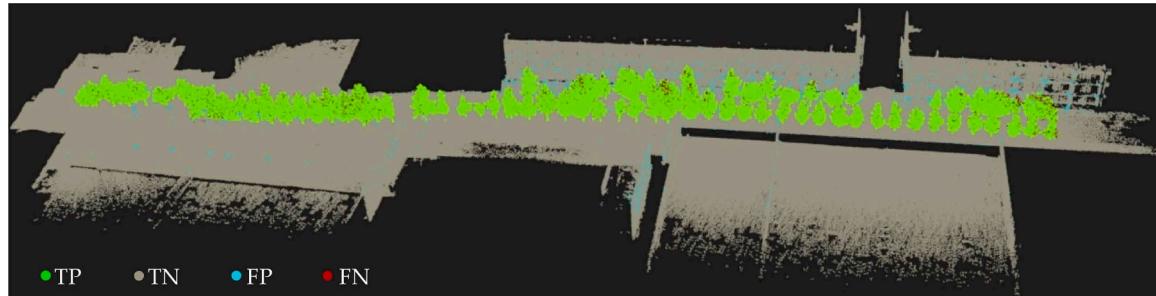
$$\text{Re} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$F_1 = \frac{2\text{Pr}\text{Re}}{\text{Pr} + \text{Re}} \quad (4)$$

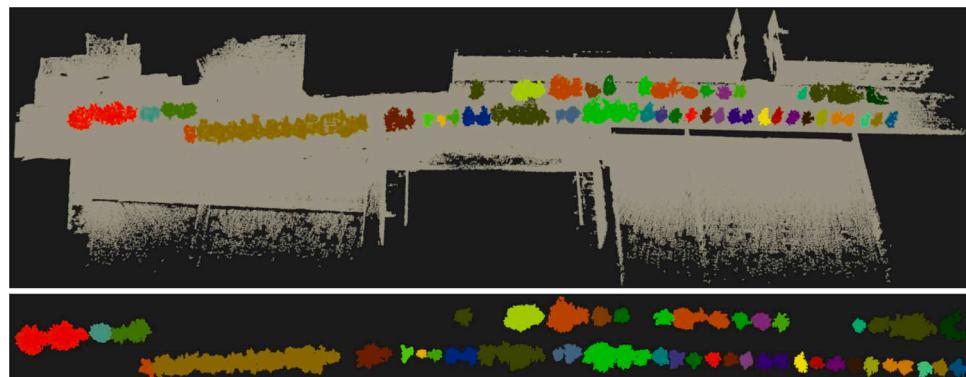
The precision is used to measure exactness (i.e., how many of the points classified as tree points are classified), and the recall is used to measure completeness (i.e., how many tree points are classified correctly out of the total number of actual tree points). The  $F_1$  score combines precision and recall into a single metric using a harmonic mean calculation. The RF algorithm in the sklearn library is used to train the tree detector. The number of decision trees is set to 10, and the other parameters are set to the default values. On the test set, the  $F_1$  score of the tree detection is 0.9734, with a precision of 0.9673 and a recall of 0.9795. Fig. 11 shows the tree detection result for the entire point cloud. The false alarm points (FPs) are concentrated on the walls and street-lamps, and the missed detection points (FNs) are concentrated on the treetops.



**Fig. 10.** Fine segmentation of overlapping trees: input (a), the first iteration (b) and the second and final iteration (c). Three representative inaccuracies in the tree tips are marked with circles in the input. All the inaccuracies are improved in the first iteration with some erroneous segmentation points left in the blue circle, which are corrected in the second iteration.



**Fig. 11.** Tree detection result on the entire point cloud using a pointwise detector trained by RF. The false alarm points (FPs) are mainly on the walls and street lamps, and the missed detection points (FNs) are concentrated on the tree tops.



**Fig. 12.** Initial segmentation result. A total of 42 tree proposals are obtained.

**Table 2**

Numbers of trees included in the tree proposals. Each tree proposal contains 1–14 individual trees.

| No. of trees          | 1  | 2 | 3 | 4 | 14 |
|-----------------------|----|---|---|---|----|
| No. of tree proposals | 28 | 8 | 1 | 4 | 1  |

### 3.2. Results of initial segmentation

After tree detection, the tree points are extracted and clustered into a total of 42 tree proposals. Then, through elevation difference filtering and  $k$ -NN reclassification, the  $F_1$  score of tree detection is increased to 0.9916 with a precision of 0.9989 and a recall of 0.9864. Fig. 12 shows the initial segmentation result.

### 3.3. Results of the overlapping tree recognition

By counting the tree trunks in each tree proposal, 14 overlapping tree proposals are identified, and the remaining 28 proposals are output as individual trees. Table 2 lists the detailed numbers of trees included in all the tree proposals. Each tree proposal contains 1–14 individual trees.

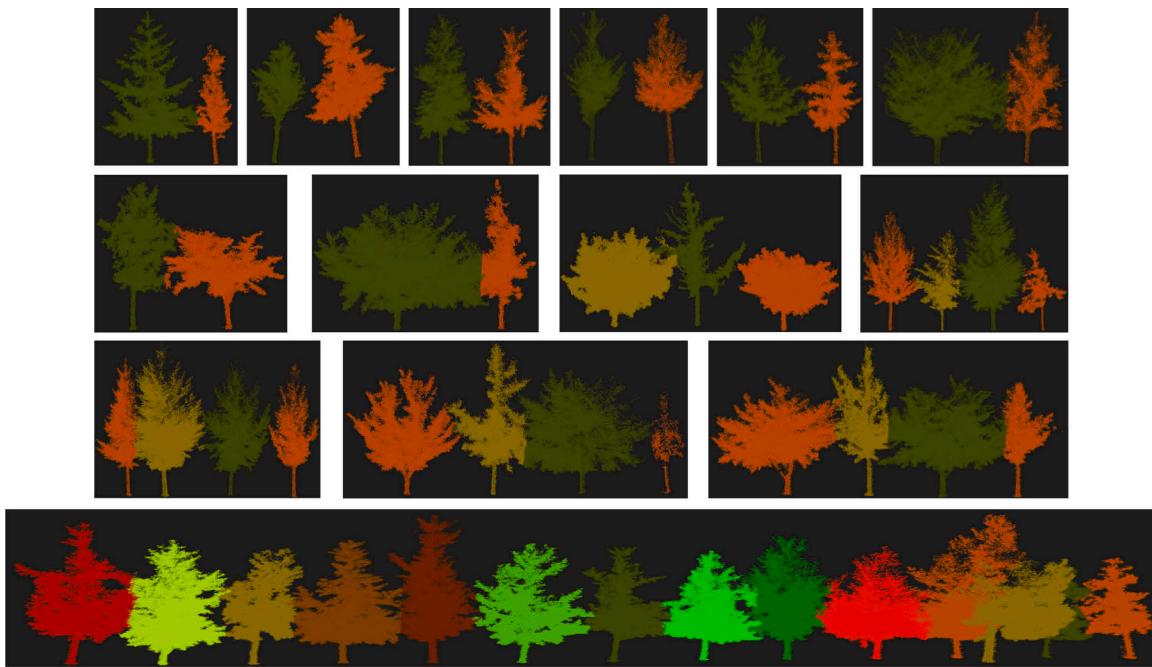
### 3.4. Results of the overlapping tree segmentation

The overlapping tree proposals are further divided into individual trees using coarse and fine segmentation. Fig. 13 shows the segmentation results of the overlapping tree proposals.

In Section 2.2, TP, FP and FN are defined to evaluate the tree detection accuracy. In this section, TP, FP and FN are redefined to evaluate the individual tree segmentation accuracy. For each individual tree, a correctly segmented point is categorized as a TP, while a wrongly segmented point belonging to the background or another tree is categorized as an FP. If a point belonging to an individual tree is not detected, it is identified as an FN. The number of TPs, FPs and FNs is counted for all the individual trees. Then, the precision, recall, and  $F_1$  score are computed. Table 3 shows the segmentation accuracy of the individual trees. The segmentation accuracy of trees without any overlapping is higher than that of trees with overlapping; the  $F_1$  score of the former is 0.9913, while the  $F_1$  score of the latter is 0.9691.

### 3.5. Comparison with other methods

Two methods using a classification-segmentation framework similar to ours are implemented for comparison. The street point cloud is first classified into tree points and nontree points using a pointwise classifier



**Fig. 13.** Segmentation results of the overlapping tree proposals. All individual trees are correctly recognized, with a small amount of incorrectly segmented points at the treetops.

**Table 3**

Segmentation accuracy of individual trees. The segmentation accuracy of trees without any overlapping is higher than that of trees with overlapping; the  $F_1$  score of the former is 0.9913, while the  $F_1$  score of the latter is 0.9691.

| Type                      | Precision | Recall | $F_1$  |
|---------------------------|-----------|--------|--------|
| Trees without overlapping | 0.9841    | 0.9987 | 0.9913 |
| Trees with overlapping    | 0.9618    | 0.9766 | 0.9691 |
| All trees                 | 0.9672    | 0.9819 | 0.9745 |

**Table 4**

Segmentation accuracy of trees with overlaps. Compared to the “horizontal projection + mean-shift” method and “vertical projection + snake” method, our method increases the  $F_1$  score by 0.0914 and 0.0617, respectively.

| Method                             | Precision | Recall | $F_1$  |
|------------------------------------|-----------|--------|--------|
| Horizontal projection + mean-shift | 0.8575    | 0.8982 | 0.8774 |
| Vertical projection + snake        | 0.9013    | 0.9136 | 0.9074 |
| Our method                         | 0.9618    | 0.9766 | 0.9691 |

trained by a supervised learning algorithm; then, the tree points are segmented into individual trees. The main differences among these methods pertain to the individual tree segmentation. Weinmann et al. (2017) used a “horizontal projection + mean-shift” strategy to segment individual trees. First, the detected tree points were downsampled and projected onto a horizontal plane, i.e., the  $xy$ -plane. The 2D projections of the tree points were treated as discrete points sampled from an empirical 2D probability density function (PDF), in which the tree centers are considered to have the largest local density. Then, the mean shift algorithm, which is an iterative statistical technique for locating the maxima/modes of a PDF without requiring the number of expected modes, was used to segment the 2D tree points into modes, in which each mode corresponds to an individual tree. Finally, the 2D segmentation results were mapped onto the raw 3D tree points, and 3D individual trees were obtained. Hua et al. (2022) proposed a “vertical projection + snake” method to segment tree points into individual trees. First, tree points were projected onto an optimal plane in which the generated tree pixels overlap the least. Obviously, this plane is vertical

and perpendicular to the street line. Then, the individual trees were segmented using an active contour/snake model on the Canny edge of the projection image. Finally, the snake’s convergent curves were matched with tree points to obtain a 3D contour of the individual trees.

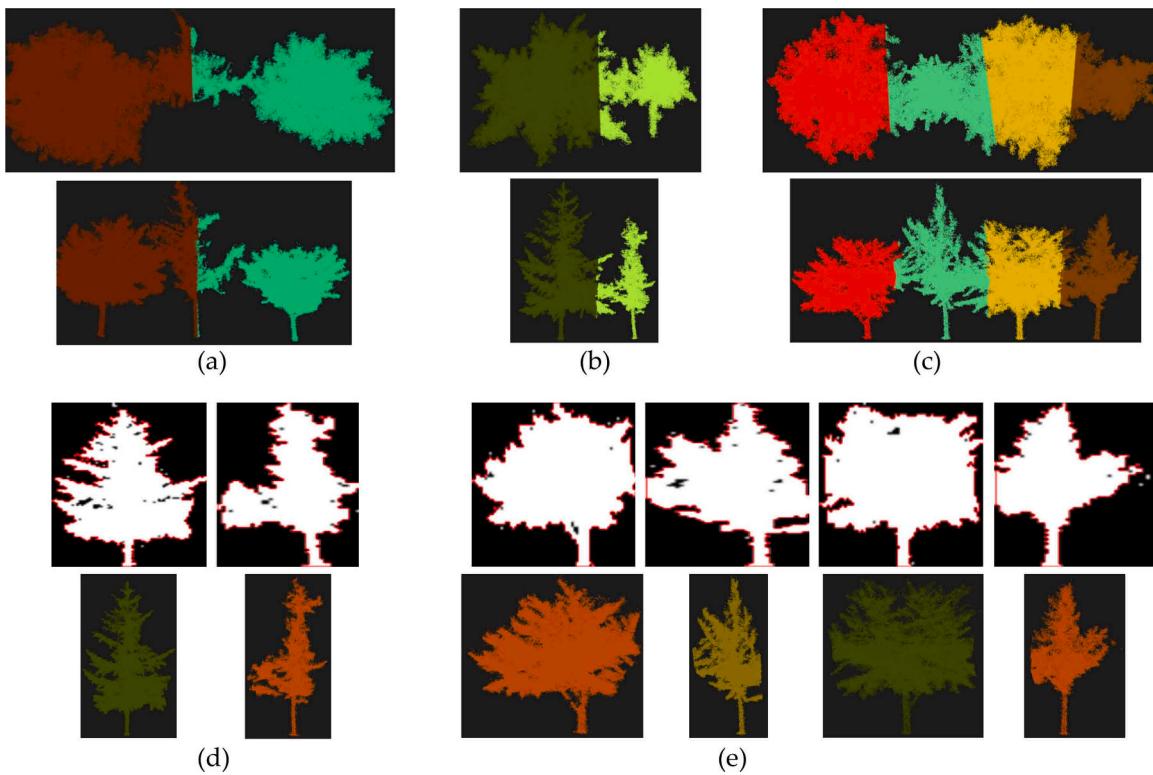
Individual tree segmentation is most challenging during the segmentation of overlapping trees. Table 4 shows the segmentation accuracy of overlapping trees. Compared to the “horizontal projection + mean-shift” method and “vertical projection + snake” method, our method increases the  $F_1$  score by 0.0911 and 0.0617, respectively.

Fig. 14 shows the segmentation results of the two comparison methods. Since a mean shift finds the local maxima of a PDF, it cannot recognize trees with no significant increase in central density. For example, the middle tree in Fig. 14(a) does not have a significant change in the canopy thickness of the canopy and is thus incorrectly detected. In addition, a mean shift classifies 2D tree points into their closest local maxima. Thus, the boundary between the overlapping trees is the perpendicular bisector of the line segment connecting the tree local maxima (see Fig. 14(a-c)), making the segmentation inaccurate. In the vertical projection image, the structure of the trees is clearer and can be segmented more accurately than in the horizontal projection image (see Fig. 14(d-e)). Due to the independent segmentation of each tree using the snake model, the tree points located at the overlap of two trees are determined to belong to both trees.

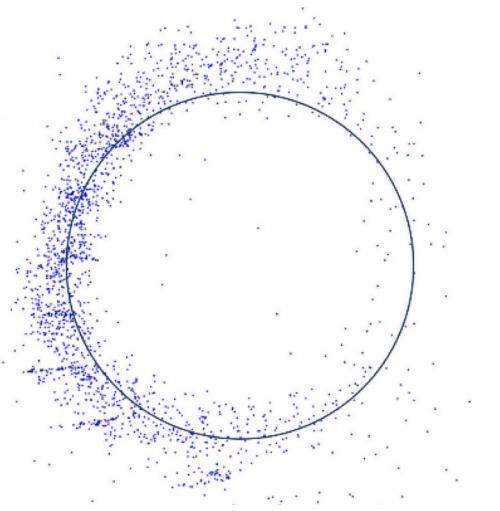
### 3.6. Application in tree parameter extraction

After individual tree segmentation, tree parameters such as tree height, crown width and DBH can be extracted from the point cloud of individual trees. This extraction is an important urban forest inventory task and is crucial for urban forest management. Here, we take the DBH measurement as an example to illustrate the application of this method in tree parameter extraction.

First, based on the comparison of the lowest elevation of an individual tree point cloud, the trunk point cloud with an elevation difference between 1.25 m and 1.35 m is segmented and projected onto a horizontal plane, i.e., the  $xy$ -plane. Then, a circle is fitted using random sample consensus (RANSAC), where the diameter of the circle is an estimate of the DBH (see Fig. 15). The measured DBH values are collected



**Fig. 14.** Segmentation results of “horizontal projection + mean-shift” (a-c), and “vertical projection + snake” (d-e). The upper row shows the 2D plane segmentation results, and the lower row shows the 3D point cloud segmentation results. The boundary between the overlapping trees generated by mean-shift is the perpendicular bisector of the line segment connecting the tree local maxima, making the segmentation inaccurate. The snake model independently segments each tree and determines the tree points located at the overlap of two trees to belong to both trees.

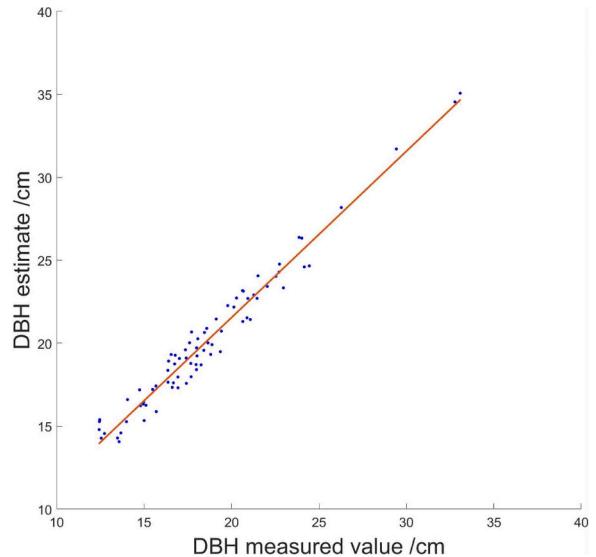


**Fig. 15.** DBH measurement. The trunk point cloud with an elevation difference between 1.25 m and 1.35 m is projected onto a horizontal plane. Then, a circle is fitted using RANSAC, and the diameter of the circle is an estimate of the DBH.

by manually measuring the trunk diameter at a height of 1.3 m. Fig. 16 shows the measured DBH values and estimates of the 77 trees and the root mean squared error (RMSE) is 0.8485 cm, and the coefficient of determination ( $R^2$ ) is 0.9615.

### 3.7. Discussion

In this paper, the segmentation step in the classification-segmentation framework is improved for individual tree segmentation. Overlapping trees are recognized and processed more effectively.



**Fig. 16.** DBH measured values and estimates of the 77 trees and a fitting line. The RMSE is 0.8485 cm, and the  $R^2$  is 0.9615.

The classification-segmentation framework for individual tree segmentation proposed by Weinmann et al. (2017) first classifies the street point cloud into tree points and nontree points using a pointwise classifier; the tree points are then segmented into individual trees. The pointwise classifier is fused from a set of low-level local features and trained by a supervised learning algorithm, making tree detection simple and accurate, which is demonstrated in Section 3.1.

Following the classification-segmentation framework, the main task of second step of the current methods is to segment individual trees in

1D or 2D. Weinmann et al. (2017) projected the tree points onto a horizontal plane, i.e., the  $xy$ -plane. Hua et al. (2022) projected the tree points onto a vertical plane, in which the projected tree pixels overlap the least. Focusing on the MLS data captured by a 2D LiDAR, Li et al. (2020) segmented the individual trees at the scanline level, in which a scanline is defined as a 2D profile measured by a 2D LiDAR. The number of crown and trunk points per scanline were counted, and the split scanline was recognized through 1D histograms of the crowns and trunks. These dimensionality reduction segmentation methods simplify the segmentation process and are effective for nonoverlapping trees. However, due to the loss of one or two information dimensions, when trees overlap, accurately segmenting individual trees in a space after the dimensionality reduction is not possible. This is shown in Fig. 14.

In the proposed method, the space between two adjacent trunks is first cut into slices. The number of tree points per slice is counted, and a 1D histogram is constructed in which the individual trees are roughly segmented. Second, taking the coarse segmentation result as input, the individual trees are finely segmented in the 3D space using an iterative algorithm combined with DBSCAN clustering and  $k$ -NN classification. This  $^1\text{D} - ^3\text{D}$  method balances the simplicity and accuracy of the tree segmentation step.

The proposed method is also applicable to the segmentation of nonstreet trees, which are not necessarily arranged in a straight line. Hua et al. (2022) projected all the tree points onto a vertical plane perpendicular to the street line to segment, which requires the trees to line up. In our method, the proposed 1D projection is performed between two adjacent trees, and the partition plane is perpendicular to the  $xy$ -plane projection of the line determined by the centroids of their trunks. This makes our method suitable for the segmentation of clustered overlapping trees.

While the proposed method performs well in the study area, several limitations and recommendations should be considered for large-scale and complex scenes. For tree detection, an RF classifier described in Section 2.2 is learned on a training set to generate a tree detector, which achieves a  $F_1$  score of 0.9734 in the test set. In the experiment, points in the training set are randomly selected. However, for large-scale applications, a training set should be carefully selected to include as many urban object categories as possible. The learned classifier can then be widely applied to detect street trees. For overlapping tree recognition as described in Section 2.4, a trunk should be segmented first to recognize and locate a street tree. If the trunk is obscured by traffic or infrastructure in the point cloud, then the tree cannot be identified during tree segmentation.

Recently, deep learning networks have been introduced to tree detection and segmentation frameworks (Chen et al., 2021; Jiang et al., 2023). For tree detection, the point clouds are divided into geometrically isotropic patches or supervoxels; then, a supervoxelwise or patchwise network is used to predict the label of a supervoxel or patch. For tree segmentation, an instance-level network trained on individual tree point clouds segments individual trees from the tree point clouds. Due to the large number of parameters in deep learning networks, these networks require a large training set. For example, the dataset in Jiang et al., (2023) was divided into training, validation, and testing sets with a 75/5/25 % split, while the training set in our experiment contains only 10 % of the points in the entire point cloud. For small-scale street scenes, the segmentation effect of deep network-based methods is not prominent, and methods with simpler data processing, such as ours, are more efficient. However, for large-scale and complex scenes, deep learning-based methods may be more competitive.

#### 4. Conclusions

Following the classification-segmentation framework, a coarse-to-fine segmentation method of individual street trees from a side-view point cloud is proposed in this paper. First, the tree points are detected from the side-view street point cloud by a pointwise classifier fused

from 13 local geometric features and trained by RF. Based on the detected tree points, the tree proposals are obtained by DBSCAN clustering and detection error filtering. Then, the overlapping tree proposals are recognized by trunk identification, and the single tree proposals are directly output as individual trees. Next, the overlapping trees are roughly divided into individual tree proposals through vertical planes. Finally, individual trees with optimized contours are obtained using an iterative algorithm combined with DBSCAN clustering and  $k$ -NN classification. For tree detection, the  $F_1$  score is 0.9916 with a precision of 0.9989 and a recall of 0.9864. For individual tree segmentation, the  $F_1$  score is 0.9745 with a precision of 0.9672 and a recall of 0.9819. Compared to two current methods, the proposed method increases the  $F_1$  score of the overlapping tree segmentation by 0.0914 and 0.0617.

After an individual tree segmentation, the tree parameters, such as the tree height, crown width and DBH, can be extracted at the individual tree level. In our experiment, the RMSE of the DBH estimation is 0.8485 cm.

#### CRediT authorship contribution statement

**Qiujie Li:** Conceptualization, Methodology, Formal analysis, Investigation, Resources, Writing - Review & Editing, Funding acquisition, Supervision. **Yu Yan:** Methodology, Software, Formal analysis, Investigation, Data curation, Visualization, Validation Writing - Original Draft. **Weizheng Li:** Conceptualization, Resources, Writing - Review & Editing.

#### Declaration of Competing Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work.

#### Acknowledgements

This research was supported by the National Natural Science Foundation of China under grant number 31901239.

#### Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.ufug.2023.128097.

#### References

- Anderson, E.C., Locke, D.H., Pickett, S.T.A., LaDoux, S.L., 2023. Just street trees? Street trees increase local biodiversity and biomass in higher income, denser neighborhoods. *Ecosphere*. <https://doi.org/10.1002/ecs2.4389>.
- Branson, S., Wegner, J.D., Hall, D., Lang, N., Schindler, K., Perona, P., 2018. From Google Maps to a fine-grained catalog of street trees. *ISPRS J. Photogramm. Remote Sens.* <https://doi.org/10.1016/j.isprsjprs.2017.11.008>.
- Chen, Y., Wu, R., Yang, C., Lin, Y., 2021. Urban vegetation segmentation using terrestrial LiDAR point clouds based on point non-local means network. *Int. J. Appl. Earth Obs. Geoinf.* <https://doi.org/10.1016/j.jag.2021.102580>.
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining.
- Galle, N.J., Halpern, D., Nitolslawski, S., Duarte, F., Ratti, C., Pilla, F., 2021. Mapping the diversity of street tree inventories across eight cities internationally using open data. *Urban For. Urban Green.* <https://doi.org/10.1016/j.ufug.2021.127099>.
- Gotsch, S.G., Draguljić, D., Williams, C.J., 2018. Evaluating the effectiveness of urban trees to mitigate storm water runoff via transpiration and stemflow. *Urban Ecosyst.* <https://doi.org/10.1007/s11252-017-0693-y>.
- Hau, M., Kulmala, L., Kolari, P., Vesala, T., Riikonen, A., Järvi, L., 2022. Carbon sequestration potential of street tree plantings in Helsinki. *Biogeosciences*. <https://doi.org/10.5194/bg-19-2121-2022>.
- Hua, Z., Xu, S., Liu, Y., 2022. Individual tree segmentation from side-view LiDAR point clouds of street trees using shadow-cut. *Remote Sens.* 14 (22) <https://doi.org/10.3390/rs14225742>.

- Huang, X., Song, J., Wang, C., Chui, T.F.M., Chan, P.W., 2021. The synergistic effect of urban heat and moisture islands in a compact high-rise city. *Build. Environ.* <https://doi.org/10.1016/j.buildenv.2021.108274>.
- Igoe, D.P., Downs, N.J., Parisi, A.V., Amar, A., 2020. Evaluation of shade profiles while walking in urban environments: a case study from inner suburban Sydney, Australia. *Build. Environ.* <https://doi.org/10.1016/j.buildenv.2020.106873>.
- Jareemit, D., Srivani, M., 2022. A comparative study of cooling performance and thermal comfort under street market shades and tree canopies in tropical savanna climate. *Sustainability*. <https://doi.org/10.3390/su14084653>.
- Jiang, K., Chen, L., Wang, X., An, F., Zhang, H., Yun, T., 2022. Simulation on different patterns of mobile laser scanning with extended application on solar beam illumination for forest plot. *Forests*. <https://doi.org/10.3390/f13122139>.
- Jiang, T., Wang, Y., Liu, S., Zhang, Q., Zhao, L., Sun, J., 2023. Instance recognition of street trees from urban point clouds using a three-stage neural network. *ISPRS J. Photogramm. Remote Sens.* <https://doi.org/10.1016/j.isprsjprs.2023.04.010>.
- Kabisch, N., Püffel, C., Masztalerz, O., Hemmerling, J., Kraemer, R., 2021. Physiological and psychological effects of visits to different urban green and street environments in older people: a field experiment in a dense inner-city area. *Landscape. Urban Plan.* <https://doi.org/10.1016/j.landurbplan.2020.103998>.
- Kim, J.Y., Jo, H.K., 2022. Estimating carbon budget from growth and management of urban street trees in South Korea. *Sustainability*. <https://doi.org/10.3390/su14084439>.
- Lai, Y., Kontokosta, C.E., 2019. The impact of urban street tree species on air quality and respiratory illness: a spatial analysis of large-scale, high-resolution urban data. *Health Place*. <https://doi.org/10.1016/j.healthplace.2019.01.016>.
- Li, J., Cheng, X., Wu, Z., Guo, W., 2021. An over-segmentation-based uphill clustering method for individual trees extraction in urban street areas from MLS Data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* <https://doi.org/10.1109/JSTARS.2021.3051653>.
- Li, J., Cheng, X., Xiao, Z., 2022. A branch-trunk-constrained hierarchical clustering method for street trees individual extraction from mobile laser scanning point clouds. *Meas.: J. Int. Meas. Confed.* <https://doi.org/10.1016/j.measurement.2021.110440>.
- Li, Q., Yuan, P., Liu, X., Zhou, H., 2020. Street tree segmentation from mobile laser scanning data. *Int. J. Remote Sens.* <https://doi.org/10.1080/0143161.2020.1754495>.
- Li, J., Slik, F., 2022. Are street trees friendly to biodiversity? *Landscape. Urban Plan.* <https://doi.org/10.1016/j.landurbplan.2021.104304>.
- Lu, Y., Ferranti, E.J.S., Chapman, L., Pfraun, C., 2023. Assessing urban greenery by harvesting street view data: a review. *Urban For. Urban Green.* <https://doi.org/10.1016/j.ufug.2023.127917>.
- Ma, B., Hauer, R.J., Östberg, J., Koeser, A.K., Wei, H., Xu, C., 2021. A global basis of urban tree inventories: what comes first the inventory or the program. *Urban For. Urban Green.* <https://doi.org/10.1016/j.ufug.2021.127087>.
- Miao, C., Li, P., Huang, Y., Sun, Y., Chen, W., Yu, S., 2022. Coupling outdoor air quality with thermal comfort in the presence of street trees: a pilot investigation in Shenyang, Northeast China. *J. For. Res.* <https://doi.org/10.1007/s11676-022-01497-y>.
- Raj, T., Hashim, F.H., Huddin, A.B., Ibrahim, M.F., Hussain, A., 2020. A survey on LiDAR scanning mechanisms. *Electronics*. <https://doi.org/10.3390/electronics9050741>.
- Safaie, A.H., Rastafei, H., Shams, A., Sarasua, W.A., Li, J., 2021. Automated street tree inventory using mobile LiDAR point clouds based on Hough transform and active contours. *ISPRS J. Photogramm. Remote Sens.* <https://doi.org/10.1016/j.isprsjprs.2021.01.026>.
- Sofia, S., Sferlazzza, S., Mariottini, A., Niccolini, M., Coppi, T., Miozzo, M., La Mantia, T., Maetzke, F., 2021. A case study of the application of hand-held mobile laser scanning in the planning of an Italian forest (Alpe di Catenai, Tuscany). *Int. Arch. Photogramm., Remote Sens. Spat. Inf. Sci. - ISPRS Arch.* <https://doi.org/10.5194/isprs-archives-XLIII-B2-2021-763-2021>.
- Sun, X., Xu, S., Hua, W., Tian, J., Xu, Y., 2022. Feasibility study on the estimation of the living vegetation volume of individual street trees using terrestrial laser scanning. *Urban For. Urban Green.* <https://doi.org/10.1016/j.ufug.2022.127553>.
- Szota, C., Coutts, A.M., Thom, J.K., Virahsawmy, H.K., Fletcher, T.D., Livesley, S.J., 2019. Street tree stormwater control measures can reduce runoff but may not benefit established trees. *Landscape. Urban Plan.* <https://doi.org/10.1016/j.landurbplan.2018.10.021>.
- Wang, W., Xiao, L., Zhang, J., Yang, Y., Tian, P., Wang, H., He, X., 2018. Potential of Internet street-view images for measuring tree sizes in roadside forests. *Urban For. Urban Green.* <https://doi.org/10.1016/j.ufug.2018.09.008>.
- Wang, Yanjun, Chen, Q., Zhu, Q., Liu, L., Li, C., Zheng, D., 2019. A survey of mobile laser scanning applications and key techniques over urban areas. *Remote Sens.* <https://doi.org/10.3390/rs11131540>.
- Wang, Yupeng, Akbari, H., 2016. The effects of street tree planting on urban heat island mitigation in Montreal. *Sustain. Cities Soc.* <https://doi.org/10.1016/j.scs.2016.04.013>.
- Weinmann, M., Weinmann, M., Mallet, C., Brédif, M., 2017. A classification-segmentation framework for the detection of individual trees in dense MMS point cloud data acquired in urban areas. *Remote Sens.* <https://doi.org/10.3390/rs903277>.
- Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., Huang, Y., Wu, J., Liu, H., 2013. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sens.* <https://doi.org/10.3390/rs5020584>.
- Zhou, Y., Yuan, Y., Chen, Y., Lai, S., 2020. Association pathways between neighborhood greenspaces and the physical and mental health of older adults – a cross-sectional study in Guangzhou, China. *Front. Public Health*. <https://doi.org/10.3389/fpubh.2020.551453>.
- Zhu, Y., Li, D., Fan, J., Zhang, H., Eichhorn, M.P., Wang, X., Yun, T., 2023. A reinterpretation of the gap fraction of tree crowns from the perspectives of computer graphics and porous media theory. *Front. Plant Sci.* <https://doi.org/10.3389/fpls.2023.1109443>.