



Street tree segmentation from mobile laser scanning data using deep learning-based image instance segmentation

Qiujie Li^{*}, Yu Yan

Nanjing Forestry University, China



ARTICLE INFO

Handling Editor: Dr Cecil Konijnendijk van den Bosch

Keywords:

Street tree segmentation
Mobile laser scanning
Image instance segmentation
Point cloud instance segmentation
Deep learning

ABSTRACT

Street tree inventory is an important part of urban forest inventory. Mobile laser scanning (MLS) technology has a strong data acquisition ability inside the canopy and the trunk and is suitable for parameter estimation at the tree level. The first and key step is to segment individual trees from the street MLS data, which is a challenging problem due to the semantic gap. This paper proposes an efficient and effective method for street tree segmentation from MLS data using deep learning-based image instance segmentation. First, the three-dimensional (3D) street point cloud captured by the MLS system is mapped to a two-dimensional (2D) RGB image. Next, pixelwise tree proposals are segmented from the street image by a trained deep learning-based image instance segmentation model. Then, the 2D segmentation mask is mapped back to the 3D street point cloud to generate pointwise tree proposals. Finally, the proposals are optimized to obtain the final results. To evaluate and verify algorithm performance, an MLS point cloud is collected from a 1481.8 m-long one-side urban street containing various objects. Three deep learning-based image instance segmentation algorithms, YOLACT, BlendMask, and YOLOv8, are carried out, and YOLOv8 achieves the best results in terms of both accuracy and speed. YOLOv8 has the highest segmentation accuracy, with IoU= 0.85:0.05:0.95, with an average segmentation time of 26 ms per image. In the experiments comparing the algorithms to the existing hierarchical segmentation and classification segmentation methods, the proposed method outperforms the other two methods in accuracy and is faster. The precision is 0.9988, the recall is 0.9986, the F₁ score is 0.9987, and the time per scanline is 4.05 ms. Moreover, the proposed method can be applied to MLS data with a broad range of resolutions by introducing image resizing.

1. Introduction

Street trees are an important part of urban forests and urban landscapes and ecosystems (Richards and Edwards, 2017; Revelli and Porporato, 2018). Facing increasingly serious environmental pollution problems in contemporary cities, street trees are of great significance in purifying air (Jeanjean et al., 2017; Buccolieri et al., 2020; Huang et al., 2021), reducing noise (Ba and Kang, 2019; Felipe et al., 2019), conserving water (Grey et al., 2018; Poozan et al., 2022), and regulating climate (McBride and Laćan, 2018; Knapp, Jaganmohan and Schwarz, 2019). Street tree inventory is an important part of urban forest inventory (Liu et al., 2023), which is essential for managing and monitoring urban forests, such as tree growth and failure assessment (Jahani, 2019; Jahani and Saffariha, 2021, 2022; Tian et al., 2024). Parameters at the tree level, such as street tree species, families, genera, diameter breast height (DBH), tree height, and crown width are obtained by manual measurement and sampling inventory (Galle et al., 2021) to

establish assessment models, which is time-consuming and labour-intensive.

Light detection and ranging (LiDAR) sensors can quickly and accurately capture the surface information in the form of a three-dimensional (3D) point cloud (Qi et al., 2023; Xue, 2022; Zhu, 2023). Street tree inventory based on LiDAR has become a current research hotspot and future development trend. Mobile laser scanning (MLS) technology carries the LiDAR scanner on the vehicle and dynamically collects the high-precision vertical structure data of trees during the vehicle driving process (Wang et al., 2019). MLS has a strong data acquisition ability for the inside of the canopy and the trunk and is suitable for parameter estimation at the tree level (Husain and Vaishya, 2019). Street tree segmentation from the MLS data is the first and most important street tree inventory task. Currently, there are two main segmentation methods: hierarchical segmentation and classification segmentation.

Hierarchical segmentation methods manually design the segmentation features and rules according to the morphological characteristics of

* Corresponding author.

E-mail addresses: lqj@njfu.edu.cn (Q. Li), alex1997@njfu.edu.cn (Y. Yan).

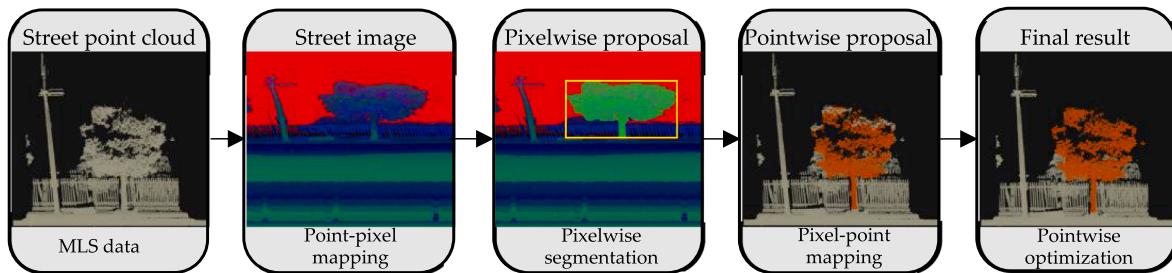


Fig. 1. The segmentation framework contains four steps: point-pixel mapping, pixelwise segmentation, pixel-point mapping, and pointwise optimization. First, the MLS point cloud of the street is mapped to an RGB image. Next, street tree proposals are segmented from the street image by a deep learning-based image instance segmentation algorithm. Then, the pixelwise proposals are mapped back to the street point cloud. Finally, the pointwise proposals are optimized to obtain the final results.

trees and other ground objects and use a variety of point cloud processing algorithms to gradually filter nontree point clouds, such as grounds and buildings. Li et al. (2021a, 2021b) first generated supervoxels to reduce the quantity of original data by oversegmentation. Then, tree crown and trunk candidates were recognized using structural characteristics. Finally, individual trees were segmented through uphill clustering. Safaie et al. (2021) first filtered ground points and detected tree trunks using the Hough transform. Then, individual tree crowns were segmented by a combination of Voronoi tessellation and active contour model. Li et al. (2022a, 2022b) proposed a branch-to-trunk matching method to locate street tree positions, based on which individual tree crowns were segmented by a new hierarchical clustering method. Due to street scene complexity, hierarchical segmentation methods based on artificially designed features and rules have a limited generalization and expression ability.

Classification segmentation methods divide individual tree segmentation into two steps: tree detection and segmentation. These methods first use a pointwise tree detector to classify the whole MLS point cloud into tree points and nontree points (Li et al., 2023). Then, the detected tree point cloud is segmented into individual trees (Weinmann et al., 2017). Weinmann et al. (2017) used the random forest (RF) algorithm to train a tree detector. Then, they applied an adaptive mean shift algorithm and a shape analysis to extract individual street trees. Hua et al. (2022) calculated the optimal projection of the tree point cloud detected by a support vector machine (SVM) classifier to the two-dimensional (2D) image, where the tree pixels overlap the least, and extracted the tree edges. Then, the individual trees were segmented by matching 3D tree contours with 2D tree edges. Li et al. (2020) trained tree crown and trunk detectors separately through the discrete AdaBoost algorithm. Individual trees were segmented at the scanline level by exploiting nonconnectivity between trees and adjacency within a tree. Compared with hierarchical segmentation methods, this kind of method benefits from the ability of machine learning to automatically fuse features and has higher street tree detection accuracy. However, these methods still need to artificially design low-level local features in the classification step and segmentation rules in the segmentation step.

Individual street tree segmentation from MLS data is a 3D point cloud instance segmentation problem, which assigns the same label to points belonging to the same instance (He, Shen and van den Hengel, 2021). In recent years, some end-to-end point cloud instance segmentation algorithms based on deep learning, such as SGPN (Wang and Yu, 2018), PointGroup (Jiang et al., 2020), Mask3D (Schult et al., 2023), etc., have emerged. These algorithms do not require manual design of segmentation features or rules, and they automatically mine multi-level point cloud features to construct output point cloud instance segmentation models. However, compared to their 2D counterparts, i.e., image instance segmentation based on deep learning, these algorithms have higher computational complexity. Currently, various top-performing deep learning-based methods work well for image instance segmentation tasks (Nan2023a; Nan2023b), such as the COCO task (Parekh et al.,

2021), in which the dataset includes more than 3,310,000 images (of which more than 200,000 have been labelled), containing 1.5 million objects divided into 80 categories. Benefiting from the ability of deep learning to automatically learn features, these image instance segmentation algorithms can avoid the semantic gap problem caused by artificial design features or rules and complete image instance segmentation tasks based on high-level semantics. Considering the excellent performance of deep learning-based image instance segmentation algorithms and their lower computational complexity compared to deep learning-based point cloud instance segmentation algorithms, one solution is to map point clouds to images and then perform instance segmentation at the image level, which has been used in the KITTI dataset built for autonomous driving (Geiger et al., 2013). Each point cloud frame collected by a 3D LiDAR sensor is mapped into a gray image through spherical projection, in which a point corresponds to a pixel, and then a deep learning-based image instance segmentation model is used to extract a variety of street objects (Wu et al., 2018; Abbate et al., 2022). This paper focuses on an MLS system using a single 2D LiDAR sensor. Similarly, an inherent correspondence between a 2D LiDAR-based MLS point cloud and an RGB image can be explored, making it possible to use the powerful feature representation ability of deep learning-based image instance segmentation to segment street trees. This paper proposes an efficient and effective method for individual street tree segmentation from a 2D LiDAR-based MLS point cloud utilizing a deep learning-based image instance segmentation algorithm. The contributions are threefold:

1. Through the one-to-one correspondence between points and pixels, bidirectional conversion between a 2D LiDAR-based MLS point cloud and an RGB image is realized. Then, the fast and efficient image instance segmentation algorithm is utilized to generate tree proposals without manual design of segmentation features and rules.
2. Point cloud processing algorithms are used to optimize tree proposals at the point level, improving segmentation accuracy.
3. The proposed method can be applied to 2D LiDAR-based MLS data with a broad range of resolutions.

2. Materials and methods

The proposed street tree segmentation method contains four steps (Fig. 1): point-pixel mapping, pixelwise segmentation, pixel-point mapping, and pointwise optimization. First, the MLS point cloud of the street is mapped to an RGB image. Next, street tree proposals are segmented from the street image by a deep learning-based image instance segmentation algorithm. Then, the pixelwise proposals are mapped back to the street point cloud. Finally, the pointwise proposals are optimized to obtain the final results.

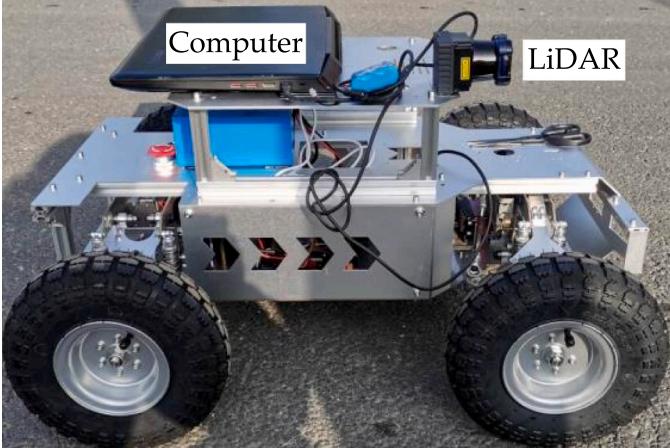


Fig. 2. The MLS system comprises a UTM-30LX-EW LiDAR sensor, a data collection computer, and a remote-control vehicle.

2.1. Experimental data generation

2.1.1. MLS system

The LiDAR used in the experiments is UTM-30LX-EW (Hokuyo Automatic Co., Ltd. Osaka, Japan). It emits a laser beam at 905 nm and measures the ranges and intensities of the first three returning echoes, denoted by r_1 , r_2 , r_3 , I_1 , I_2 , and I_3 . The repeated accuracy is 30 mm in a range of up to 60 m. If the distance between the measured object and LiDAR is more than 60 m, the range is limited to 60 m. The LiDAR continuously scans with an angle range of 270° and an angle resolution of 0.25° and obtains a series of scanlines. Each scanline contains ranges and intensities measured by 1081 laser beams with different angles. The time interval Δt between two scanlines is 25 ms.

The MLS system comprises a UTM-30LX-EW LiDAR sensor, a data collection computer, and a remote-control vehicle (R550 PLUS), as shown in Fig. 2. The LiDAR sensor is mounted on the front end of the vehicle and works in push-broom mode, i.e., its scan plane is orthogonal to the direction of vehicle movement, with the scan centerline parallel to the ground. The street surface data are collected by the LiDAR sensor

while the vehicle is driving. The motion control system in the robotic vehicle ensures that the vehicle travels at a constant speed v .

2.1.2. Street data collection

The vehicle travels back and forth along a street section ($32^{\circ}03'03''\text{N}$, $118^{\circ}39'34''\text{E}$) at a speed of 0.4 m s^{-1} (Fig. 3). The surface data of a 1481.8 m-long side street are collected, containing 148,180 scanlines and a total of 84,512,580 points. There are a variety of objects in the MLS data, including trees, walls, overline bridges, lanes, sidewalks, street lamps, traffic signs, dustbins, grids, bushes, turfs, motorcycles, and pedestrians.

2.1.3. Data preprocessing

A Cartesian coordinate system is first established to obtain the 3D coordinates of the MLS point cloud (Fig. 3): the x -axis points to the direction of vehicle movement, the y -axis is perpendicular to the x -axis and points to the measured street side, and the z -axis points upward. The raw MLS data are stored in chronological order and can be uniquely specified by the inter-scanline number i and the intra-scanline number j (Li et al., 2021b). Given the vehicle speed v , scan circle Δt , initial scan angle θ_0 , angle resolution $\Delta\theta$, and primary echo range $r_1(i,j)$, the 3D coordinates of point $P(i,j)$ can be obtained by

$$\begin{cases} & \&x(i,j) = v\Delta t \bullet i \\ & \&y(i,j) = r_1(i,j)\cos(\Delta\theta \bullet j + \theta_0) \\ & \&z(i,j) = r_1(i,j)\sin(\Delta\theta \bullet j + \theta_0) \end{cases} \quad (1)$$

To focus on single-sided street data, data with an angle range of -90° to 90° are extracted from the raw MLS data of the 270° angle range. The initial scan angle θ_0 is set to -90° , and the beam number j ranges from 1 to 720.

After generating the 3D coordinates of the street MLS data, CloudCompare is used to label a total of 146 street trees from the street point cloud (Fig. 4).

2.2. Image instance segmentation

2.2.1. Three-channel image construction

To apply the image instance segmentation algorithms, the MLS data should first be converted into a three-channel colour image. Since the



Fig. 3. Data collection location (Google Maps, $32^{\circ}03'03''\text{N}$, $118^{\circ}39'34''\text{E}$).

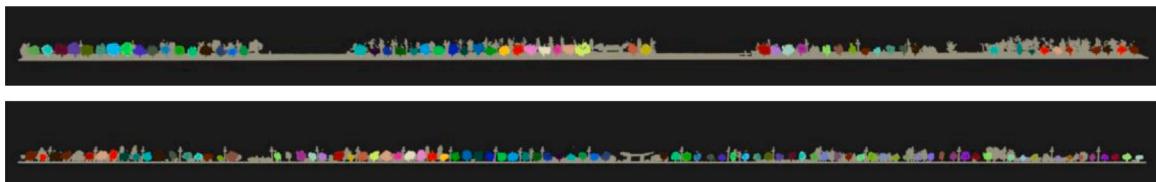


Fig. 4. The street point cloud labelled with individual street trees. Points with $r_1 > 16 \text{ m}$ are not displayed.

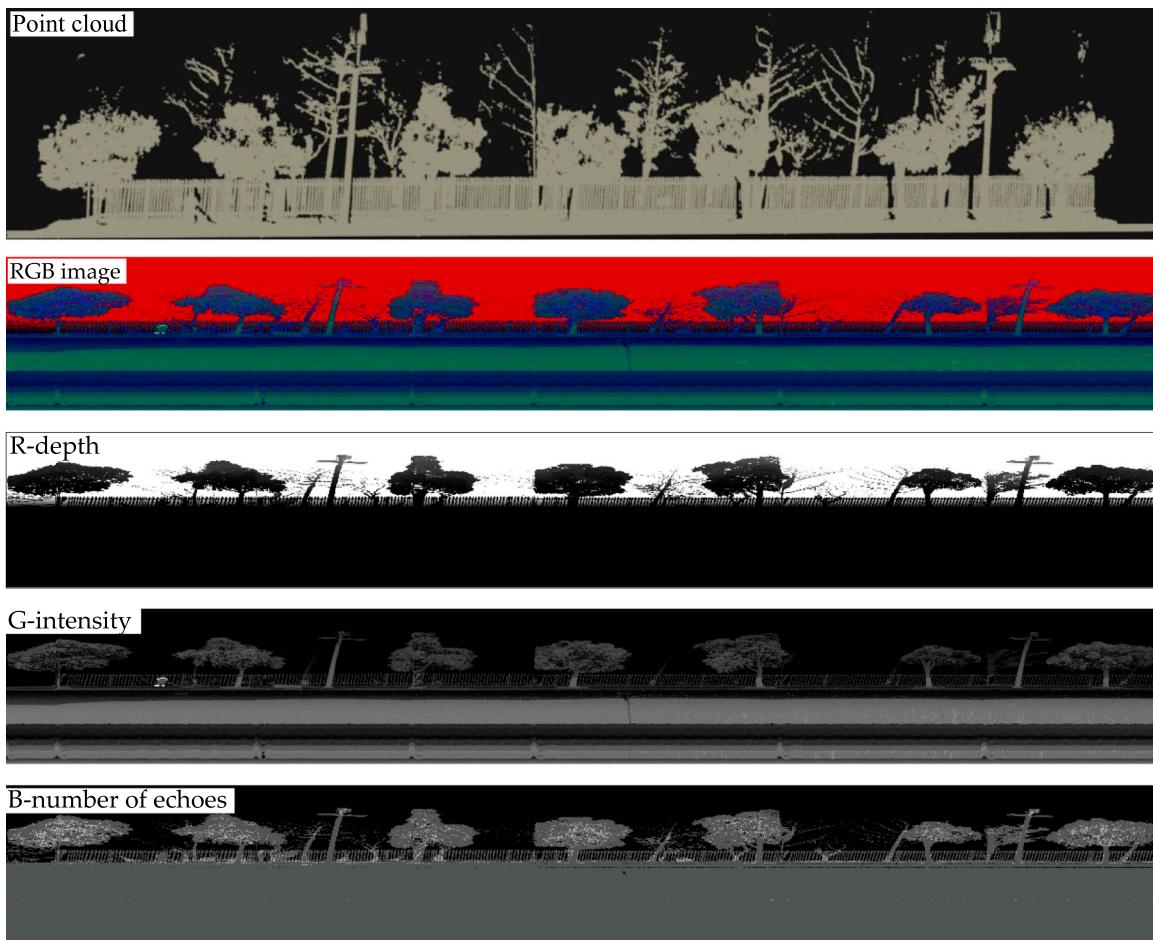


Fig. 5. The three-channel colour images and three single-channel images of a street section.

MLS data use a grid index similar to the image, the transformation is lossless, and the point and pixel can correspond one to one. The point-pixel mapping rule is that the point measured by the j -th laser beam in the i -th scanline corresponds to the pixel in the j -th row and i -th column in the image.

Then, three MLS data attributes should be selected to construct the three-channel RGB image. First, consider the 3D coordinates x, y , and z . The coordinate x of a point represents its distance from the origin along the LiDAR trajectory. According to Eq. 1, the coordinate x is linearly related to the corresponding pixel column coordinate i of the point. Most street objects are upright, and the push-broom mode of LiDAR strongly correlates the vertical height coordinate z of these objects with their pixel row coordinate j . The coordinate y of a point describes its horizontal displacement relative to the LiDAR position. There is a weak relevance between the coordinate y of a point and its pixel coordinates i and j . Additionally, the coordinate y of tree points has a specific range,

which is helpful for tree detection. Therefore, the coordinate y is selected as the red channel, and coordinates x and z are abandoned. Second, the echo intensity can reveal material reflection characteristics, so it is often used for object classification. Thus, the intensity of the first echo I_1 is selected as the green channel. Third, if the laser beam hits object edges, two or three echoes are generated. For trees composed of leaves, the number of echoes n can be used to distinguish them from solid street objects (Li et al., 2022a, 2022b; Li and Xue, 2023). For the UTM-30LX-EW LiDAR, if there is no second or third echo return, the corresponding range and intensity values are recorded as 0, from which n can be calculated, $n \in \{0, 1, 2, 3\}$. Finally, the number of echoes n is selected as the blue channel. Then, min-max normalization is used to change the range of the three selected attributes to [0255].

Fig. 5 shows the three-channel colour images and three single-channel images of a street section. The three channels describe the street in terms of location, reflectivity, and porosity. It can be observed

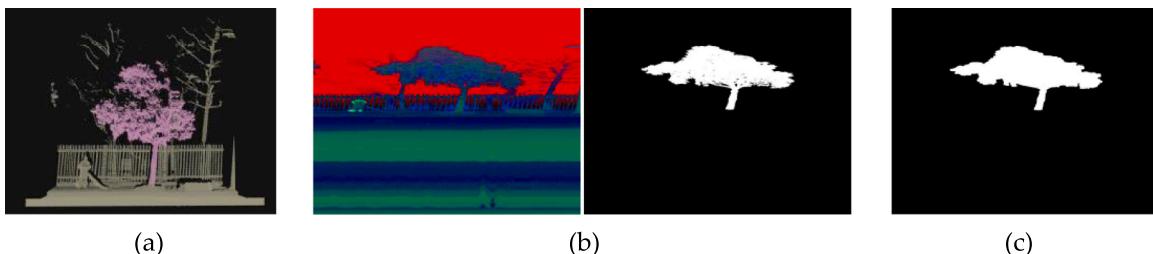


Fig. 6. Image annotation of a tree. Point cloud annotation (a), point-pixel image and its mask (b), and final mask (c). There are nonconnected pixels and holes in the point-pixel mask. The final mask contains only one connected region without holes.

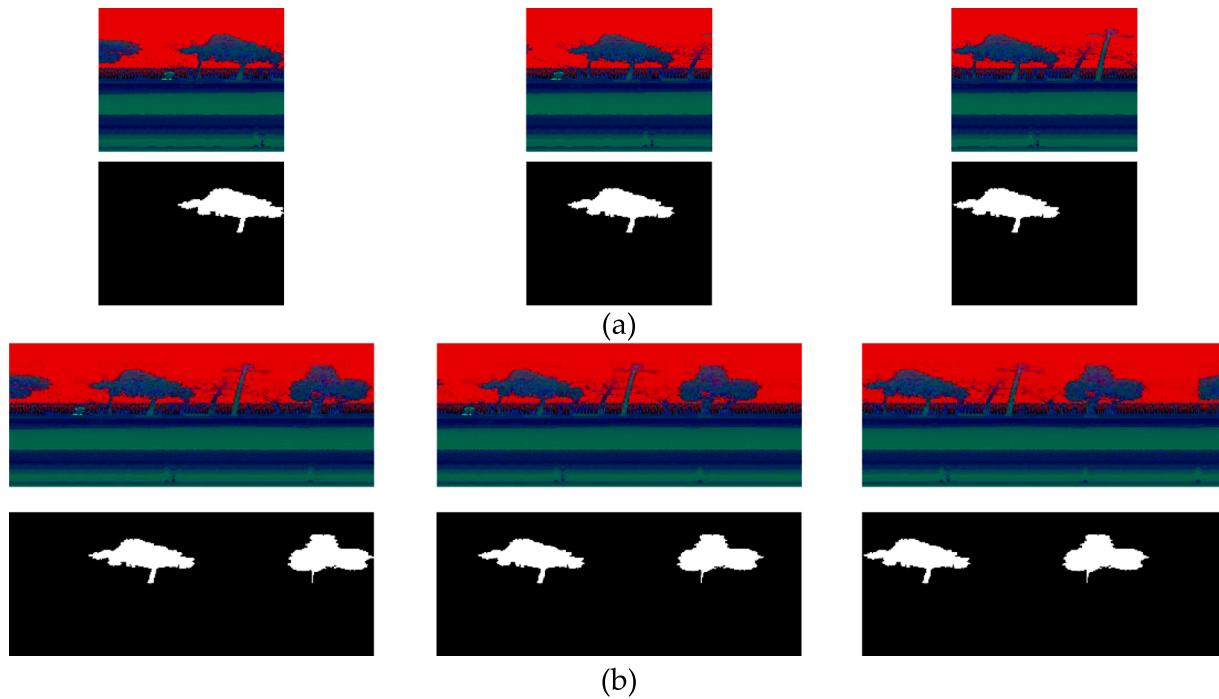


Fig. 7. Image cutting. The RGB image and its mask image of the street are cut into short images containing only one or two trees. The sliding window method is used to generate tree images at 50 different positions with a sliding step of 8 cm and a total of 4 m long margin left on both sides of the trees. The first, twenty-fifth, and fiftieth RGB images and their masks of single tree (a) and two trees (b) are given.

that street trees exist within a certain distance range; lane dustbins, street lamps, and the lower parts of the trunk painted white have a high reflectivity; and there are multiple echoes at crowns and contours.

2.2.2. Image dataset generation

To train the image instance segmentation model and evaluate its performance, an annotated image dataset is needed. Tree image annotation can be represented by a binary image, i.e., a mask image. First, point-pixel mapping is used to generate the initial mask image from the point cloud annotation. Due to the gaps among leaves, there are non-connected pixels and holes in the single tree mask. Therefore, morphology operations, including closed operation, hole filling, and area filtering, are adopted to obtain an effective tree mask in which there

is only one connected region without holes. [Fig. 6](#) shows the point cloud annotation, the point-pixel image and its mask, and the final tree mask.

To facilitate the application of the deep learning-based image instance segmentation algorithm, the RGB image and its mask image of the street are cut into short images containing only one or two trees. A sliding window is used to generate dense tree images with a sliding step of 8 cm and a total of 4 m-long margin left on both sides of the trees ([Fig. 7](#)). Note that incomplete trees in image margins are not reserved in mask images. For each single tree or double tree, there are 50 images in which their positions are different. Then, the generated image dataset is expanded using horizontal flip. Finally, 17,356 tree images are generated, and their encoded polygonal masks in json format are created using pycocreator from their mask images.

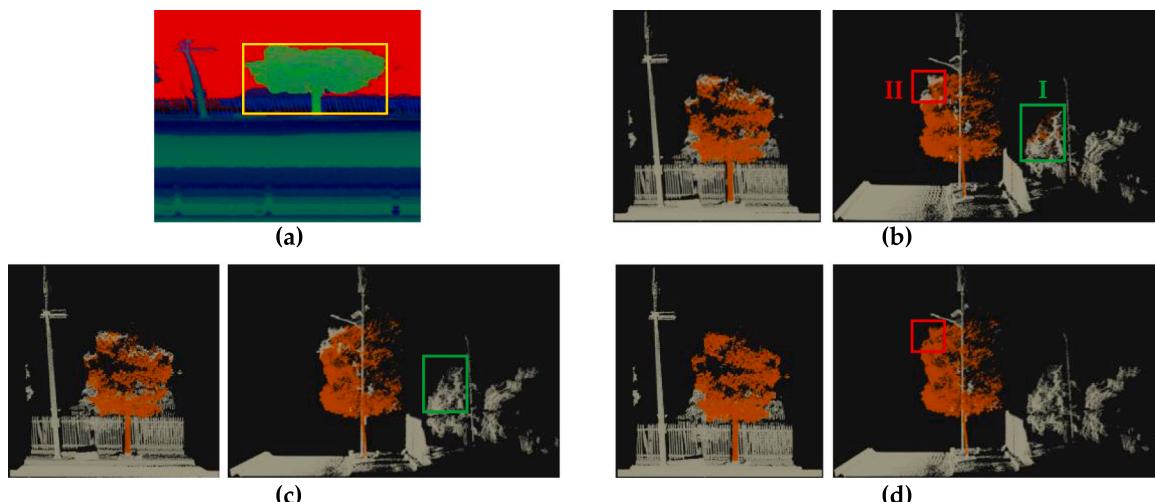


Fig. 8. Point cloud segmentation optimization. Pixelwise tree proposal (a) and its pointwise correspondence with two main types of inaccurate segmentation marked (b), result after DBSCAN (c), and result after k-NN classification (d). For ease of viewing, point clouds were presented from two perspectives. DBSCAN filters out the Type I false alarm points while k-NN classification recognizes the Type II missed detection points.

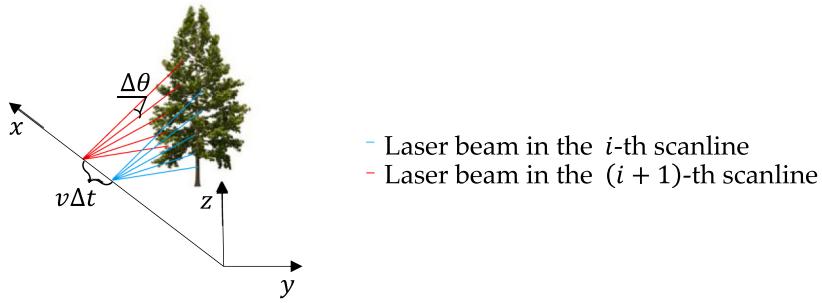


Fig. 9. The two resolutions in the MLS system.

The image dataset is partitioned into a training set, a verification set and a test set at a ratio of 8:1:1. The training and verification sets are used to train a segmentation model using the deep learning-based image instance segmentation algorithm, and the test set is used to evaluate segmentation accuracy. To effectively evaluate the algorithm performance, the street trees contained in the three data sets are not duplicated.

2.3. Point cloud segmentation optimization

After point-pixel mapping, the tree instances are segmented from the RGB image using the trained segmentation model. Then, the pixelwise tree proposals are projected back to the point cloud by pixel-point mapping to obtain pointwise tree proposals. Although the deep learning segmentation model can accurately detect street trees in the image, the tree contours need further optimization in the form of a point cloud. There are two main types of inaccurate segmentation among pixelwise tree proposals: the background points behind trees, which are measured by the beams passing through crown gaps, are incorrectly classified as tree points; and the tree edge points are incorrectly classified as background points due to image sampling and low contour segmentation accuracy.

For the first type of misclassification, considering that most of the false alarm points are not connected with tree points, density-based spatial clustering of applications with noise (DBSCAN) is applied to filter them. There are two parameters in the DBSCAN algorithm, the neighbourhood radius ϵ and the minimum number of neighbours N_{\min} . For each point of a cluster, its neighbourhood with radius ϵ contains at least N_{\min} points. In our method, $\epsilon = 0.5$ m and $N_{\min} = 1$. DBSCAN first clusters the point cloud. Then, the largest cluster is extracted as the tree, and the remaining clusters, which are considered background points, are abandoned.

For the second type of misclassification, considering that these missed detection points are connected with correctly detected tree points, the k -nearest neighbour (k -NN) classifier is used to segment them. For each point classified as background, if more than half of its neighbours are tree points, then this point is reclassified as a tree point.

Fig. 8 shows the two main types of inaccurate image instance segmentation and the final result after point cloud segmentation optimization.

2.4. MLS resolution adaptation

There are two resolutions in the MLS system (Fig. 9): the distance increment $v\Delta t$ between adjacent scanlines and the angle increment $\Delta\theta$ between adjacent laser beams. If the resolutions vary, the MLS data amount and the image size also changes. Since the image instance segmentation model is trained on images with fixed resolutions, an image size transformation named bilinear interpolation is added before and after image instance segmentation to adapt the model to MLS data with different resolutions. The normalized resolutions after image resizing are $v\Delta t = 0.01$ m and $\Delta\theta = 0.25^\circ$. The point cloud segmentation

optimization step (Section 2.3) is robust to variations in point density and does not need to be modified.

2.5. Performance validation

2.5.1. Performance validation for image instance segmentation

To select a deep learning-based image instance segmentation algorithm for the proposed method, the segmentation accuracy of street tree image instance is evaluated at the instance level using average precision (AP) on the test set. Each tree proposal contains a segmentation mask with a confidence value. Tree proposals are first arranged in descending order of their confidence values. If the intersection over union overlap (IoU) between a mask of a tree proposal and a mask of an unmatched ground-truth (GT) tree is greater than a specific threshold, the tree proposal is considered a true positive (TP); otherwise, it is considered a false positive (FP). A GT tree that is not matched with any proposal is considered a false negative (FN).

After evaluating all tree proposals, precision and recall are calculated.

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

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

As the threshold applied to the proposal confidence varies, a precision-recall curve is generated that shows the precision as a recall function, and AP is the area under the curve.

2.5.2. Performance validation for point cloud instance segmentation

The segmentation accuracy of street tree point cloud instance is evaluated at the point level. First, the segmented individual trees are matched with the GT trees according to the pointwise IoU. Comparing a tree with its matching GT tree, its correctly segmented points are considered TPs, and the remaining points are considered FPs. The points in a GT tree that are not correctly detected are considered FNs. The number of TPs, FPs, and FNs on the test set is counted, and the F_1 score is calculated to evaluate the point-level segmentation accuracy:

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

3. Results and discussion

3.1. Developing environment

The software is developed based on the PyTorch deep learning framework with CUDA 11.1 and cuDNN 8.0.4 under Windows 10 and Python 3. The CPU is an Intel Core i9-12900k with 64 G RAM at 3.20 GHz, and the GPU is a NVIDIA GeForce RTX 3090 Ti.

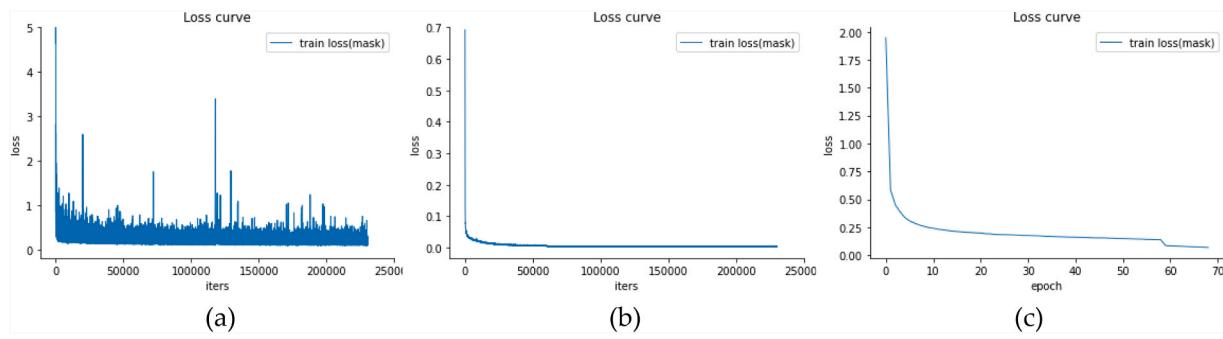


Fig. 10. Loss curve of YOLACT (a), BlendMask (b), and YOLOv8 (c) during the training process.

Table 1

AP (IoU=0.70:0.05:0.95) and the segmentation time of a single image of the three algorithms on the test set.

Algorithm	AP ₇₀	AP ₇₅	AP ₈₀	AP ₈₅	AP ₉₀	AP ₉₅	Time per image /ms
YOLACT	1.000	1.000	1.000	0.985	0.858	0.381	45
BlendMask	0.990	0.990	0.990	0.990	0.879	0.395	128
YOLOv8	0.995	0.995	0.995	0.995	0.946	0.600	26

3.2. Results of image instance segmentation

To ensure the accuracy and speed of street tree segmentation, three state-of-the-art deep learning-based image instance segmentation algorithms, YOLACT (Bolya et al., 2019), BlendMask (Chen et al., 2020), and YOLOv8 (Redmon et al., 2016), are carried out and compared in the experiment. YOLACT generates prototype masks and predicts mask coefficients in parallel mode, achieving real-time image instance segmentation for the first time. BlendMask uses a novel blender module that combines instance-level information with lower-level semantic information to improve mask prediction. YOLOv8 has improved the performance and flexibility of previous YOLO versions.

The base image sizes of YOLACT, BlendMask, and YOLOv8 are set to 550×550 , 550×550 , and 576×576 (YOLOv8 requires the image size to be a multiple of 32), respectively. The batch size is set to 4 for all algorithms. For YOLACT and BlendMask, the number of iterations is set

to 230,000. For YOLOv8, the number of epochs is set to 68. Other hyperparameters are set to default values. The three segmentation models are trained on training and verification sets. Fig. 10 shows the training loss curve of the three algorithms. BlendMask and YOLOv8 converge during the training process, while there are oscillations in the training loss curve of YOLACT.

Table 1 shows AP (IoU=0.70:0.05:0.95) and the segmentation time of the three models on the test set. YOLOv8 performs the fastest segmentation with 26 ms per image. YOLACT achieves the best segmentation accuracy with IoU= 0.70:0.05:0.80, while YOLOv8 achieves the highest accuracy with IoU= 0.85:0.05:0.95. Fig. 11 shows the GT masks and segmentation results of four test images. Affected by image sampling at the beginning and ending in the segmentation models, all proposal masks have an overall offset of several pixels compared with GT masks. Compared with the other two algorithms, YOLOv8 detects incomplete trees not labelled in the GT (Fig. 11 (a), (b), and (d)) with lower confidence, causing FP to increase and AP to decrease. In addition, YOLOv8 can produce masks with more accurate edges than the other two algorithms. In Fig. 11(d), only YOLOv8 accurately segments the right contour of the second tree, which sticks to a street lamp and another tree. In Fig. 11(b), BlendMask fails to detect the bulge at the lower-right corner of the crown and misclassifies some background spots at the lower part of the crown into trees. In Fig. 11(c), YOLACT cannot accurately segment pixels between two close but not connected trees.

In conclusion, YOLOv8 performs best in both segmentation accuracy

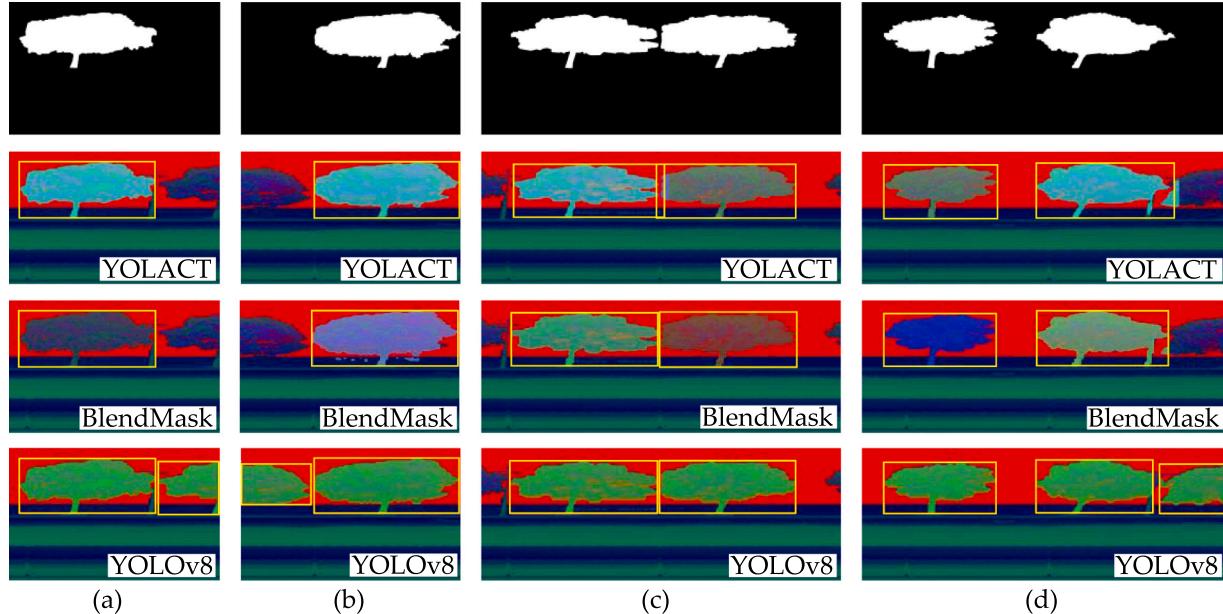


Fig. 11. Segmentation results of the three algorithms on four test images including single tree (a-b) and double trees (c-d).

Table 2

Segmentation accuracy and time of four algorithms on the test set.

Method	Precision	Recall	F_1 score	Time per scanline /ms
HS	0.9621	0.9346	0.9481	27.14
CS	0.9985	0.9597	0.9788	36.14
PS without pointwise optimization	0.9891	0.9921	0.9906	0.43
PS with pointwise optimization	0.9988	0.9986	0.9987	4.05

and speed and is selected for image-level tree instance segmentation. The detected broken trees can be removed through confidence threshold and area filtering, and only the complete tree proposals are reserved.

3.3. Results of point cloud instance segmentation

After image instance segmentation, the pixelwise proposals are

mapped back to the MLS point cloud. Then, the pointwise proposals are optimized to obtain the final results. To illustrate the accuracy improvement by pointwise optimization, the segmentation performances before and after pointwise optimization are both evaluated. To demonstrate the effectiveness of the proposed segmentation method, the existing methods are carried out and compared. There are two main types of methods for segmenting street trees from the MLS data: hierarchical segmentation (HS) and classification segmentation (CS). Here, the proposed segmentation method (PS) is compared with an HS method (Li et al., 2021a) and a CS method (Li et al., 2020). The HS method first partitions the point cloud into supervoxels. Then, planar and linear structures are filtered by region growing. Next, the remaining points are divided into clusters by uphill clustering. Finally, tree crowns are recognized from clusters and matched with tree trunks identified based on the linear structures. The CS method learns a crown detector and a trunk detector on the training set by fusing 16 local statistical features using discrete AdaBoost. Then, individual trees are located by counting the number of trunk points included in each scanline and clustering

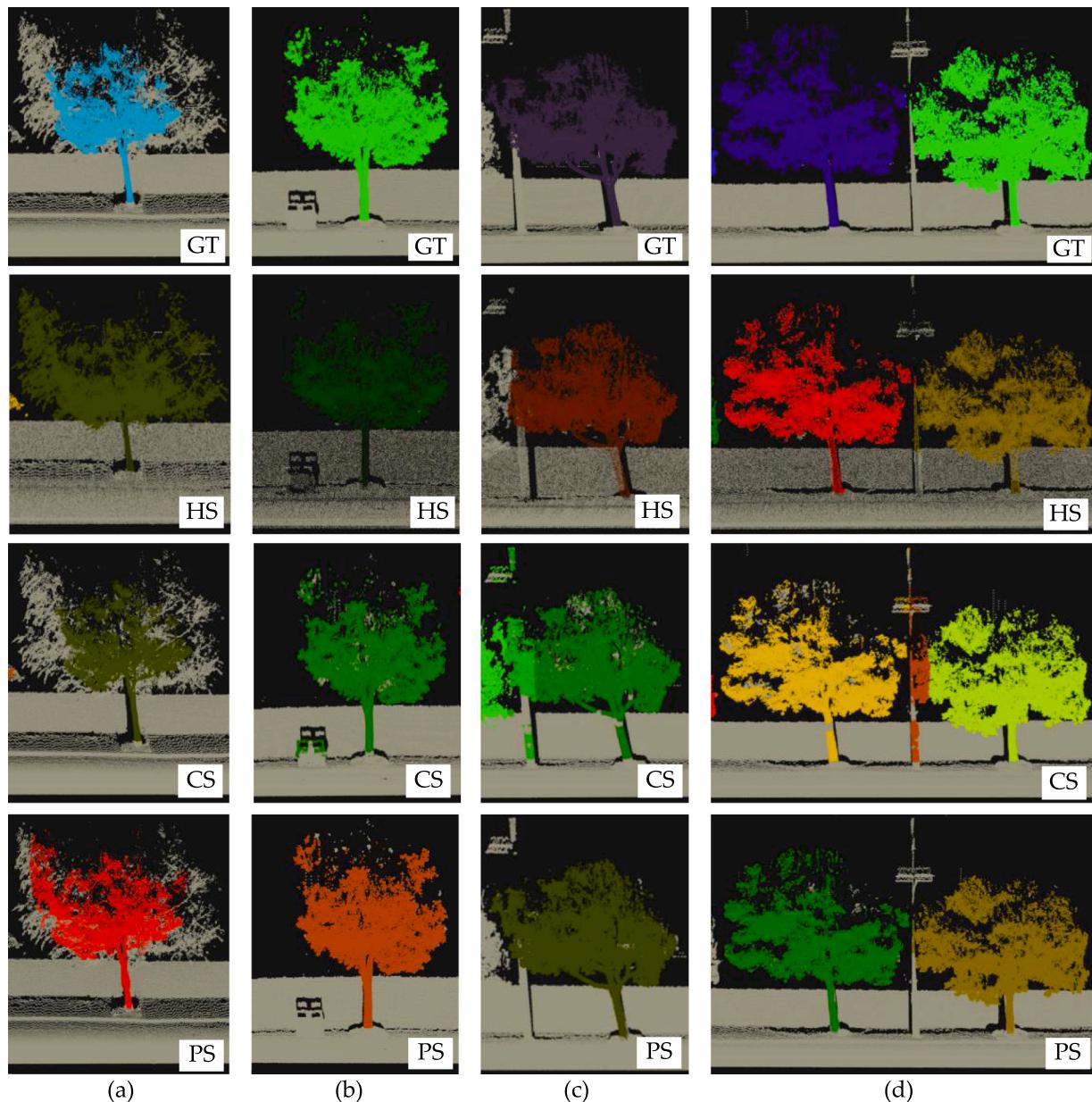
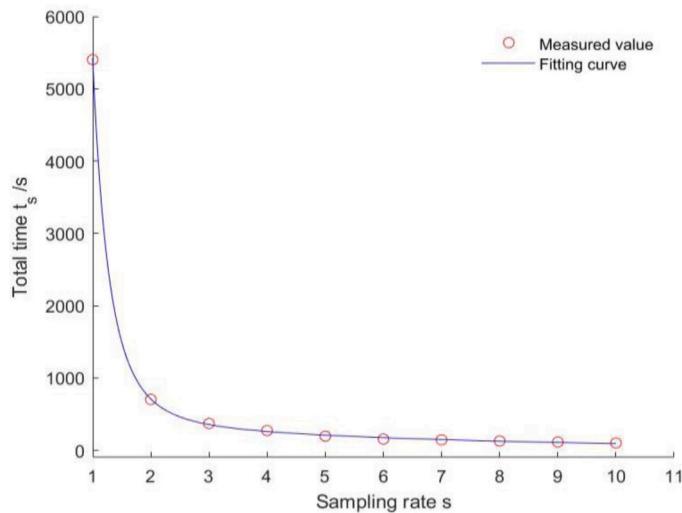


Fig. 12. Ground-truth (GT) and final segmentation results of the HS method, the CS method, and the PS method with pointwise optimization on four typical test point clouds including street tree in front of a tree (a), street tree next to a dustbin (b), and a street lamp (c-d).

Table 3

Segmentation accuracy and time on the sampled test set.

Sampling rates	$v\Delta t$ /m	$\Delta\theta$ /°	Precision	Recall	F_1 score	Total time t_s /s
1	0.01	0.25	0.9988	0.9986	0.9987	5401.272
2	0.02	0.50	0.9988	0.9984	0.9986	702.713
3	0.03	0.75	0.9988	0.9976	0.9982	368.899
4	0.04	1.00	0.9990	0.9969	0.9979	272.587
5	0.05	1.25	0.9985	0.9954	0.9969	194.641
6	0.06	1.50	0.9984	0.9943	0.9963	154.633
7	0.07	1.75	0.9981	0.9915	0.9948	144.357
8	0.08	2.00	0.9880	0.9783	0.9831	127.040
9	0.09	2.25	0.9907	0.9769	0.9837	113.320
10	0.10	2.50	0.9720	0.9542	0.9629	100.900

**Fig. 13.** Total segmentation time at different sampling rates and the fitting curve between them.

trunk scanlines. Finally, the number of crown points included in each scanline is counted, and individual trees are segmented using the scanline with the minimum crown points between adjacent trunks.

Table 2 shows the segmentation accuracy and time of four algorithms on the test set, that is, the HS method, the CS method, the PS method without pointwise optimization, and the PS method with pointwise optimization. The PS methods outperform the other two methods in terms of the F_1 score and processing time per scanline. In addition, the F_1 score of the proposed method achieves an increment of 0.0081 after pointwise optimization at the cost of a time increment of 3.62 ms.

Fig. 12 shows the GT and segmentation results of the HS method, the CS method, and the PS method with pointwise optimization on four typical test point clouds. The HS method based on region growing and clustering has difficulty handling the adhesions of other objects to tree crowns. In Fig. 12(a), a backrow tree, whose trunk is not scanned by LiDAR due to occlusion, is incorrectly identified as the street tree in front of it. In Fig. 12(d), a street lamp stuck to a crown is incorrectly segmented as the street tree to which the crown belongs. The CS method using local low-level features has limitations in recognizing tree trunks. In Figure (b), a dustbin is mistakenly identified as a tree trunk. In Figure (c-d), street lamps are recognized as tree trunks, resulting in a nonexistent street tree instance. In addition, the trunk points are not fully identified. In contrast, the PS method using deep learning to learn high-level semantics can identify individual trees more accurately.

3.4. Results of MLS resolution adaption

To evaluate the adaptability of the algorithm to resolution changes,

the raw point cloud is downsampled to obtain a total of ten point clouds with different resolutions. The sampling is conducted in both directions with sampling rate s varying from 1 to 10. Only one scanline is reserved for every s scanline, and only one beam is reserved for every s beam. Table 3 shows the segmentation accuracy and time on the sampled test set. It is observed that a high segmentation accuracy is maintained over all point clouds with significant decreases in segmentation time as sampling rate increases. Fig. 13 plots the total segmentation time at different sampling rates and the fitting curve between them. The total segmentation time t_s can be fitted using a cubic polynomial of $\frac{1}{s}$. The fitting curve equation is

$$t_s = 9348s^{-3} - 6246s^{-2} + 2393s^{-1} - 94.34 \quad (5)$$

and the coefficient of determination R^2 is 1.0000.

Fig. 14 shows the results of pixelwise segmentation and pointwise optimization at the MLS point clouds with ten resolutions. The pixelwise segmentation result is presented on the resized image with $v\Delta t=0.01$ m and $\Delta\theta=0.25$ °. As the resolution of the point cloud increases, the point cloud becomes increasingly sparse, and the interpolated image becomes increasingly blurred. However, the YOLOv8 model and the pointwise optimization algorithm can still achieve accurate segmentation.

3.5. Discussion

3.5.1. Advantages of proposed method

The proposed method utilizes the deep learning-based image instance segmentation algorithm, which can utilize the GPU's parallelism and learn high-level semantics to efficiently and effectively segment individual trees from complex street scenes. Accurate segmentation can also be achieved when the tree crown adheres to other ground objects, such as street lamps and other crowns. The lossless conversion of point clouds to images and the optimization of the point cloud level ensure that the image segmentation algorithm can be effectively applied to point cloud segmentation.

The proposed method can handle MLS data with a broad range of resolutions with a deep learning-based image instance segmentation model trained using MLS data with fixed resolutions ($v\Delta t=0.01$ m and $\Delta\theta=0.25$ °). This indicates that the method can be applied to data at different vehicle speeds v , scan cycles Δt , and angle resolutions $\Delta\theta$. Meanwhile, the method can be accelerated through undersampling, and the segmentation result of the raw data can be recovered by oversampling or the k-NN classifier.

3.5.2. Comparison with existing methods

On the one hand, both hierarchical segmentation and classification segmentation methods require manual design of segmentation features and rules based on prior knowledge about street objects, making it difficult to represent the differences between street trees and other street objects in complex street scenes. In our experiment, these methods have limitations in removing other street objects that are adhered to street trees and similar to tree trunks. The proposed method utilizes deep learning-based image instance segmentation to automatically mine multi-level features that represent the differences between street trees and other street objects from the training set and combines it with simple point cloud processing to construct a high-precision street tree segmentation model. Compared with existing methods, the F_1 score of the proposed method has improved by 0.0199.

On the other hand, existing methods mainly perform at the point cloud level, resulting in high computational complexity. The proposed method segments street trees at the image level through lossless transformation between point clouds and images, with only a small amount of optimization at the point cloud level, resulting in a relatively low computational complexity. The processing time of the proposed method is less than 15 % of that of existing methods.

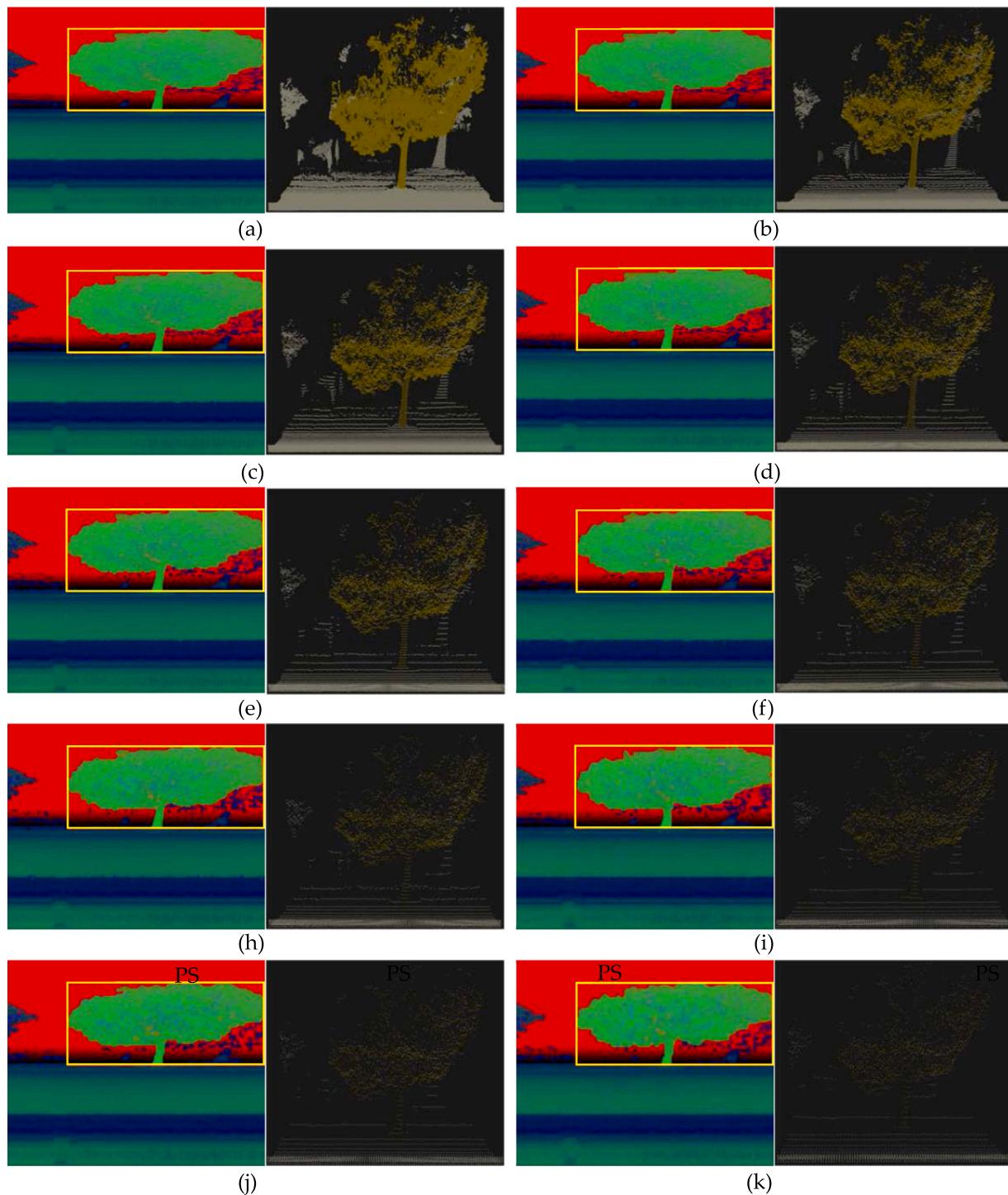


Fig. 14. Results of pixelwise segmentation and pointwise optimization at the MLS point clouds with a variety of resolutions including $v\Delta t= 0.01$ m and $\Delta\theta= 0.25^\circ$ (a), $v\Delta t= 0.02$ m and $\Delta\theta= 0.50^\circ$ (b), $v\Delta t= 0.03$ m and $\Delta\theta= 0.75^\circ$ (c), $v\Delta t= 0.04$ m and $\Delta\theta= 1.00^\circ$ (d), $v\Delta t= 0.05$ m and $\Delta\theta= 1.25^\circ$ (e), $v\Delta t= 0.06$ m and $\Delta\theta= 1.50^\circ$ (f), $v\Delta t= 0.07$ m and $\Delta\theta= 1.75^\circ$ (h), $v\Delta t= 0.08$ m and $\Delta\theta= 2.00^\circ$ (i), $v\Delta t= 0.09$ m and $\Delta\theta= 2.25^\circ$ (j), and $v\Delta t= 0.10$ m and $\Delta\theta= 2.50^\circ$ (k). The pixelwise segmentation result is presented on the resized image. image instance segmentation and Ground truth (GT) and final segmentation results of the hierarchical segmentation (HS) method, the-segmentation (CS) method, and the proposed segmentation (PS) method on four typical test point clouds including street tree in front of a tree (a) and street tree next to a dustbin (b) and a street lamp (c-d).

3.5.3. Limitations and recommendations

Several limitations and recommendations should be considered when applying the proposed method in practice. The first point is that the training set must be constructed from representative street sections that contain typical street trees and urban objects, so that the segmentation model based on deep learning can have good generalization

performance for the entire study area. If the street scene to be processed differs significantly from the training set, the segmentation performance decreases. The second point to note is that in practical applications, the proposed method needs to cut the entire street image constructed from the street MLS data into subimages for image instance segmentation. To ensure the tree integrity in subimages, the subimages must overlap with

each other. This means that a tree has multiple segmentation results for different subimages, and they need to be fused to output a unique segmentation result. The third point is that the proposed method is applicable only to MLS point clouds with known collection sequences, otherwise the MLS point cloud cannot be mapped to a street image by a point serial number. The fourth point is that the 3D coordinates of the MLS data used in the proposed method are not georeferenced (see Eq. (1)). For situations where accurate measurement is required, such as tree parameter extraction for tree hazard assessment (Jahani, 2019; Jahani and Saffariha, 2021, 2022), the raw MLS data should be corrected by sensors, such as inertial measurement units (IMUs) and global navigation satellite systems (GNSSs), to obtain the true coordinates. Then, the raw MLS data are used to segment street trees, and tree parameters are extracted from the georeferenced tree point cloud.

3.5.4. Potential improvements

The MLS system applicable to the proposed method uses a 2D LiDAR sensor in push-broom mode. In this case, the points and pixels correspond individually, which can cause the lossless bidirectional conversion of point clouds and images. The 3D LiDAR sensor, which contains a set of scanlines, can collect more detailed tree-side information from a wider perspective to provide a more accurate tree parameter estimation for tree growth and failure assessment. The proposed method may be improved to extend to an MLS system with 3D LiDAR by adding some processing steps. One method is preprocessing: the scanlines are first sorted by position that is computed based on vehicle speed, scan circle, and angle resolution between scanlines; then, the mapping from the point cloud to the image can be established, with each scanline corresponding to a column of pixels. The other method is postprocessing: only a single scanline captured by the 3D LiDAR is used for tree segmentation; then, the label of a point is spread to its neighbours. In addition, some LiDAR sensors may not measure intensity data and return only a single echo. In this case, by setting the pixel value of the channel with missing data to 0, tree proposals can still be obtained using the image instance segmentation model.

4. Conclusions

A novel method for street tree segmentation from MLS data is proposed by introducing image instance segmentation. The deep learning-based segmentation model at the image level can better capture overall information about tree instances and locate trees effectively; processing at point cloud level features can provide finer contour details. The one-to-one mapping of points and pixels prevents information loss in the mutual conversion of a 2D LiDAR-based MLS point cloud and an RGB image. The automatic generation of image annotation from point cloud annotation reduces labour consumption. Among the three image instance segmentation algorithms, YOLACT, BlendMask, and YOLOv8, YOLOv8 is superior to the other two algorithms in terms of segmentation accuracy and speed. Then, pointwise optimization improves the segmentation precision. The comparison experiment shows that the proposed method is superior to the existing methods in segmentation accuracy and speed. The precision is 0.9988, the recall is 0.9986, the F₁ score is 0.9987, and the time per scanline is 4.05 ms. In addition, the segmentation method trained on MLS data with fixed resolutions can be applied to 2D LiDAR-based MLS data with a variety of resolutions.

Funding

This research was supported by the National Natural Science Foundation of China under Grant Number 31901239.

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, there is no professional or other personal interest of any nature or kind in any product, service and/or company that could be construed as influencing the position presented in, or the review of, the manuscript entitled "Segmenting street trees from mobile laser scanning data using deep learning-based image instance segmentation".

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.ufug.2023.128200.

References

- Abbate, G. et al. (2022) 'PointIt: A ROS Toolkit for Interacting with Co-located Robots using Pointing Gestures', in ACM/IEEE International Conference on Human-Robot Interaction. doi: 10.1109/HRI53351.2022.9889486.
- Ba, M., Kang, J., 2019. Effect of a fragrant tree on the perception of traffic noise. Build. Environ. <https://doi.org/10.1016/j.buildenv.2019.04.022>.
- Bolya, D. et al. (2019) 'YOLACT: Real-time instance segmentation', in Proceedings of the IEEE International Conference on Computer Vision. doi: 10.1109/ICCV.2019.00925.
- Buccolieri, R. et al. (2020) 'Characterization of urban greening in a District of Lecce (Southern Italy) for the analysis of CO₂ storage and air pollutant dispersion', Atmosphere. doi: 10.3390/ATMOS11090967.
- Chen, H., et al., 2020. Blendmask: Top-down meets bottom-up for instance segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR42600.2020.00860>.
- Felipe, C.P. et al. (2019) 'Influence of urban trees on noise levels in a central Chilean city', Revista de la Facultad de Ciencias Agrarias.
- Galle, N.J. et al. (2021) 'Mapping the diversity of street tree inventories across eight cities internationally using open data', Urban Forestry and Urban Greening. doi: 10.1016/j.ufug.2021.127099.
- Geiger, A. et al. (2013) 'Vision meets robotics: The KITTI dataset', International Journal of Robotics Research. doi: 10.1177/0278364913491297.
- Grey, V. et al. (2018) 'Establishing street trees in stormwater control measures can double tree growth when extended waterlogging is avoided', Landscape and Urban Planning. doi: 10.1016/j.landurbplan.2018.06.002.
- He, T., Shen, C., van den Hengel, A., 2021. DyCO3D: Robust Instance Segmentation of 3D Point Clouds through Dynamic Convolution. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR46437.2021.00042>.
- 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, Y., et al., 2021. A review of strategies for mitigating roadside air pollution in urban street canyons. Environ. Pollut. <https://doi.org/10.1016/j.envpol.2021.116971>.
- Husain, A., Vaishya, R.C., 2019. Detection and thinning of street trees for calculation of morphological parameters using mobile laser scanner data. Remote Sens. Appl.: Soc. Environ. <https://doi.org/10.1016/j.rsase.2018.12.007>.
- Jahani, A., 2019. Sycamore failure hazard classification model (SFHCM): an environmental decision support system (EDSS) in urban green spaces. Int. J. Environ. Sci. Technol. <https://doi.org/10.1007/s13762-018-1665-3>.
- Jahani, A., Saffariha, M., 2021. Modeling of trees failure under windstorm in harvested Hyrcanian forests using machine learning techniques. Sci. Rep. <https://doi.org/10.1038/s41598-020-80426-7>.
- Jahani, A., Saffariha, M., 2022. Environmental decision support system for Plane trees failure prediction: a comparison of multi-layer perceptron and random forest modeling approaches. Agrosyst. Geosci. Environ. <https://doi.org/10.1002/agr2.20316>.
- Jeanjean, A., et al., 2017. Air quality affected by trees in real street canyons: the case of Marylebone neighbourhood in central London. Urban For. Urban Green. <https://doi.org/10.1016/j.ufug.2017.01.009>.
- Jiang, L., et al., 2020. PointGroup: Dual-set point grouping for 3D instance segmentation. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR42600.2020.00492>.
- Knapp, S., Jaganmohan, M., Schwarz, N., 2019. Climate Regulation by Diverse Urban Green Spaces: Risks and Opportunities Related to Climate and Land Use Change, IAS of Ecosystem Services. https://doi.org/10.1007/978-3-319-96229-0_26.
- Li, J., et al., 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., et al., 2020. Street tree segmentation from mobile laser scanning data. Int. J. Remote Sens. <https://doi.org/10.1080/01431161.2020.1754495>.
- Li, Q., et al., 2021. Pointwise classification of mobile laser scanning point clouds of urban scenes using raw data. J. Appl. Remote Sens. <https://doi.org/10.1117/1.jrs.15.024523>.

- Li, Q., et al., 2022. Street tree crown detection with mobile laser scanning data using a grid index and local features. PFG-J. Photogrammetry Remote Sens. Geoinf. Sci. 90 (3), 305–317. <https://doi.org/10.1007/s41064-022-00208-w>.
- Li, Q., Xue, Y., 2023. Real-time detection of street tree crowns using mobile laser scanning based on pointwise classification. Biosyst. Eng. IAgE 231, 20–35. <https://doi.org/10.1016/j.biosystemeng.2023.05.009>.
- Li, Q., Yan, Y., Li, W., 2023. Coarse-to-fine segmentation of individual street trees from side-view point clouds. Urban For. Urban Green. <https://doi.org/10.1016/j.ufug.2023.128097>.
- Liu, D., et al., 2023. Establishing a citywide street tree inventory with street view images and computer vision techniques. Comput. Environ. Urban Syst. <https://doi.org/10.1016/j.compenvurbysys.2022.101924>.
- McBride, J.R., Lacan, I., 2018. The impact of climate-change induced temperature increases on the suitability of street tree species in California (USA) cities. Urban For. Urban Green. <https://doi.org/10.1016/j.ufug.2018.07.020>.
- Yulong Nan, Huichun Zhang, Zeng Yong, Jiaqiang Zheng, Yufeng Ge. Intelligent detection of multi-class pitaya fruits in target picking row based on WGB-YOLO network. Computers and Electronics in Agriculture. 2023a, 208: 107780.
- Yulong Nan, Huichun Zhang, Zeng Yong, Jiaqiang Zheng, Yufeng Ge. Faster and accurate green pepper detection using NSGA-II-based pruned YOLOv5l in the field environment. Computers and Electronics in Agriculture. 2023b, 205: 107563.
- Parekh, Z. et al. (2021) 'Crisscrossed captions: Extended intramodal and intermodal semantic similarity judgments for MS-COCO', in EACL 2021 - 16th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference. doi: [10.18653/v1/2021.eacl-main.249](https://doi.org/10.18653/v1/2021.eacl-main.249).
- Poozan, A., et al., 2022. Modelling the interaction between vegetation and infiltrated stormwater. J. Hydrol. <https://doi.org/10.1016/j.jhydrol.2022.127527>.
- Qi, L., Huang, L., Zhang, Y., Chen, Y., Wang, J., Zhang, X., 2023. A Real-Time Vessel Detection and Tracking System Based on LiDAR. Sensors 23 (22), 9027. <https://doi.org/10.3390/s23229027> <https://www.mdpi.com/1424-8220/23/22/9027>.
- Redmon, J., et al., 2016. You only look once: Unified, real-time object detection. ', in Proceedings of the. IEEE Computer Society Conference on Computer Vision and Pattern Recognition. <https://doi.org/10.1109/CVPR.2016.91>.
- Revelli, R., Porporato, A., 2018. Ecohydrological model for the quantification of ecosystem services provided by urban street trees. Urban Ecosyst. <https://doi.org/10.1007/s11252-018-0741-2>.
- Richards, D.R., Edwards, P.J., 2017. Quantifying street tree regulating ecosystem services using Google Street View. Ecol. Indic. <https://doi.org/10.1016/j.ecolind.2017.01.028>.
- Safaie, A.H., et al., 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>.
- Schult, J., et al., 2023. Mask3D: Mask Transformer for 3D Semantic Instance Segmentation. Proceedings - IEEE International Conference on Robotics and Automation. <https://doi.org/10.1109/ICRA48891.2023.10160590>.
- Wang, W., Yu, R., 2018. SGN: Similarity Group Proposal Network for 3D Point Cloud Instance Segmentation University of California, San Diego'. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.
- Tian, Q., Zhang, H., Bian, L., Zhou, L., Ge, Y., 2024. Three-Dimensional Quantification and Visualization of Leaf Chlorophyll Content in Poplar Saplings under Drought Using SFM-MVS. Forests 15, 20. <https://doi.org/10.3390/f15010020>.
- Wang, Y., et al., 2019. A survey of mobile laser scanning applications and key techniques over urban areas. Remote Sens. <https://doi.org/10.3390/rs11131540>.
- Weinmann, Martin, et al., 2017. A classification-segmentation framework for the detection of individual trees in dense MMS point cloud data acquired in urban areas. Remote Sens. doi: [10.3390/rs903277](https://doi.org/10.3390/rs903277).
- Wu, B., et al., 2018. SqueezeSeg: convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. Proceedings - IEEE International Conference on Robotics and Automation. <https://doi.org/10.1109/ICRA.2018.8462926>.
- Xue, X., et al., 2022. Shortwave radiation calculation for forest plots using airborne LiDAR data and computer graphics. Plant Phenomics. <https://doi.org/10.34133/2022/9856739>.
- Zhu, Y., et al., 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>.