Regular Issue

Three Dimensional Tree Modeling Based on the Skeleton Path Optimization and Geometrical Shapes

Xin Li, Xuan Zhou, Sheng Xu*

College of Information Science and Technology & Artifical Intelligence, Nanjing Forestry University, Nanjing (China)

* Corresponding author: xusheng@njfu.edu.cn (S. Xu)

Received 25 April 2023 | Accepted 3 June 2024 | Early Access 10 October 2024

Abstract

Nowadays, the 3D individual tree reconstruction has played a significant role in the phenotypic study of trees. This paper proposes a new automatic method for extracting skeletons of individual trees and reconstructing 3D models. Firstly, the Euclidean clustering is performed to obtain center points of candidate branch regions. Then, the initial skeletons of LiDAR point clouds are obtained by slicing clusters in three dimensions. Secondly, skeleton points are completed by the proposed branch tracking. Then, the radius of the branches is accurately estimated from the branches. Thirdly, optimal points are interpolated in appropriate directions to refine skeletons of individual trees. Then, the Laplacian algorithm is conducted for smoothing branches. After that, optimal geometric shapes are formulated to reconstruct the final 3D tree models. Experimental results show that the average accuracy of our individual tree models is up to 97.49%, which shows a promising algorithm in 3D tree reconstructions.

Keywords

Automatic Tree Modeling, Extraction, Geometric Cones, Optimization, Point Cloud.

DOI: 10.9781/ijimai.2024.10.003

I. Introduction

With the rapid development of technologies in the fields of smart city, agriculture and forestry, the pace of 3D reconstruction has accelerated significantly. Fine 3D geometric models of trees, such as crowns and branches, have become an indispensable and important part of vegetation study. Remote sensing technology has been extensively used in the acquisition of forest information and land cover data since the advent of various remote sensing data, such as satellite imagery and LiDAR (Light Detection and Ranging) point clouds [1], which encourages researchers to pay more attention to the 3D real structure of trees. Considering the advantages of LiDAR scanning technology, i.e. the active sensor principle, the independence of sunlight and weather conditions, and the potential to map wide regions in short time [2], it is possible to display a large and complex regions of vegetation. The development of 3D laser data processing not only promotes the three-dimensional digital modeling technology, but also provides the possibility for tree reconstruction accurately and efficiently [3]. Due to the complexity of tree structures, constructing 3D models from LiDAR point clouds is still a challenging task [4].

Therefore, the purpose of this work is to build a three-dimensional model of a tree accurately from handheld LiDAR point clouds. For this aim, we take Ginkgo, Prunus and Platanus as experimental data. The main steps of this paper are as follows: Firstly, we pre-process the data obtained from the scanner and then extract skeleton points by a clustering algorithm and a slicing method. Secondly, we use the branch tracing algorithm to locate the branch of the skeleton points. The radius of skeletons is estimated by using the least squares method. Thirdly, the skeleton points and the corresponding radius are interpolated and optimized to complete skeletons. Finally, by forming various geometric shapes utilizing the acquired skeleton points and radius information, the 3D model of individual trees is constructed. Contributions of this work are as follows.

a) To find the clustering center points, we suggest a clustering algorithm, and then we use the slicing method to locate the corresponding skeleton points from the X, Y, and Z axes.
b) To ensure the local integrity of trees, a novel algorithm is suggested to branch and track the skeleton points. We propose the geometric shapes method of tree modeling, which makes the modeling more accurate.

II. Related Work

Various laser scanning technologies have been proposed to acquire and reconstruct 3D structural information of trees based on the collected 3D point clouds. Depending on the human-computer interaction in data processing, there are two broad categories. The one is modeling by manual or semi-automated ways, and the other is modeling automatically. Those semi-automated methods often require reference information to be extracted from tree point clouds. These pre-calculated parameters can be input into the existing tree models, so that

Please cite this article as:
X. Li, X. Zhou, S. Xu. Three Dimensional Tree Modeling Based on the Skeleton Path Optimization and Geometrical Shapes, International Journal of Interactive Multimedia and Artificial Intelligence, vol. 9, no. 5, pp. 69-77, 2025, http://dx.doi.org/10.9781/ijimai.2024.10.003

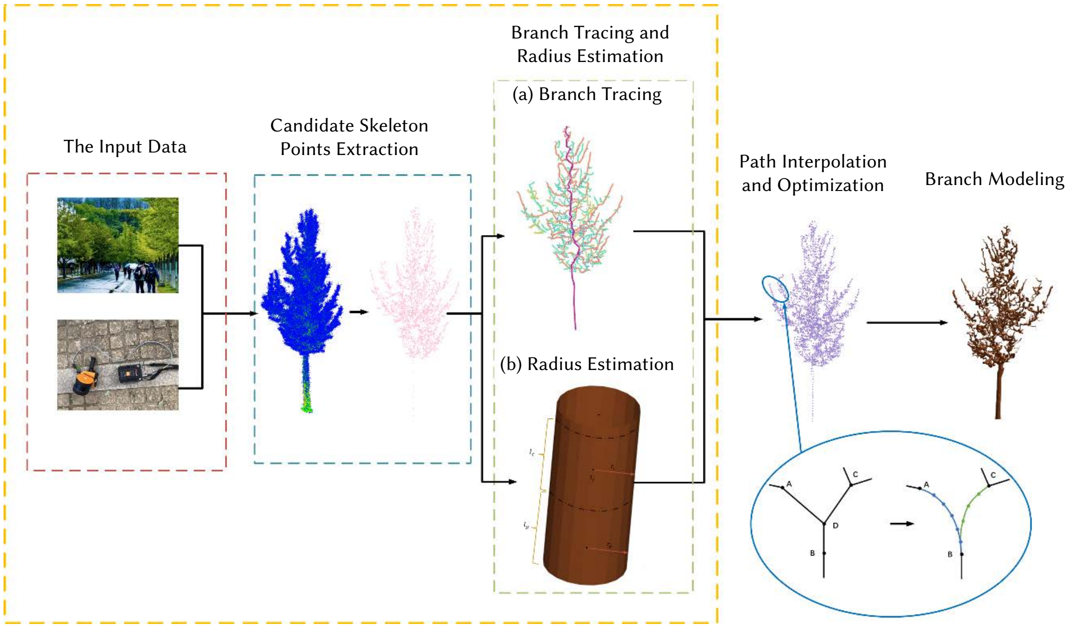


Fig. 1. Overview of the proposed approach.

the 3D tree model can be constructed with high precision. However, those methods are less efficient and realistic [5], which fail to use automatic algorithms for 3D geometric reconstruction. Most automatic approaches [6] reconstruct 3D trees by extracting skeletons first, and then they try to provide a better visualization of the targets' geometry and topology structure. Therefore, Verroust et al. [7] introduced a method for extracting accurate skeletal curves from the collection of unorganized scattered points lying on a surface. This method can help extract the skeleton points, but the output of this method is more sensitive to data incompleteness and changes in density. Xu et al. [8] proposed an algorithm for reconstructing a network model from the original scanned point clouds of individual trees. Point clouds are divided into segments by measuring distances, and then points in each segment are clustered to obtain skeleton points. However, it still requires key parameters to be specified by the user.

Livny et al. [9] constructed and optimized the skeleton points based on the physiological structure of the tree. Although their method avoids the parameter tuning, they lack sufficient quantitative evaluation experiments. Pfeifer et al. [10] used a cylindrical fitting algorithm to obtain points of skeletons. The radius of branches is based on the achieved skeletons. After that, they used a B spline curve to fit the orientation of the cylindrical model to complete the obtained 3D tree models. However, this algorithm applies to the 3D reconstruction of trees with simple topological structures rather than constructs models with fine tree structure details [8]. Li et al. [11] proposed the skeleton translation method which effectively ensures the centralization of the extracted skeletons, but this method cannot deal with horizontal branching skeleton. Jang et al. [12] utilized the graph geodesic distance to contract the sampling points so that the initial skeleton is extracted while preserving the detailed topology. However, the issues lie in the shrinkage process of the point cloud for general right-angled objects.

III. Skeleton Extraction and 3D Modeling

There are four primary steps in the 3D reconstruction of trees as shown in Fig. 1. The input data are point clouds collected by a handheld scanner. Firstly, we separate the individual trees from the input data, and obtain center points by a clustering algorithm. We slice input point clouds at a fixed distance from X-axis, Y-axis and Z-axis. The achieved point clouds are regarded as initial skeletons. Secondly, a branch tracking algorithm is proposed to find the optimal path from each point to the root as the optimal branch, and the skeleton is captured and the radius is estimated by using the least squares fitting. Thirdly, the achieved skeleton and radius are interpolated and smoothed based on Laplacian smoothing, which makes the skeleton complete and smooth to facilitate the expansion of the skeleton into a realistic 3D model. Finally, we reconstruct the 3D trees based on the prior knowledge of tree growing. We present a geometric shapes method to optimize the structure of trees locally and globally, which enhances the visual effect and improves the accuracy.

A. Candidate Skeleton Points Extraction

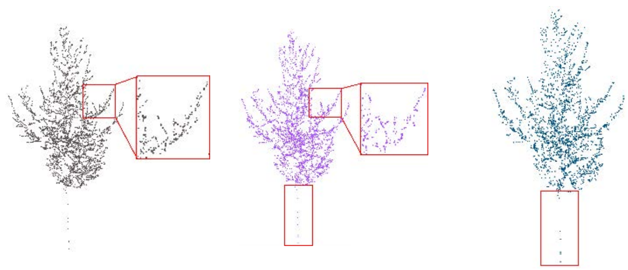
Point clouds are regarded as the basis of 3D tree reconstruction, and the data quality is important to the accuracy of the later modeling. The handheld LiDAR scanning system has good practicability and high flexibility in data collection. It changes the relative position of the scanner and the target object, which successfully obtains the whole scene in all directions. In addition, street trees are an important part of urban ecosystems and urban landscapes [13]. Therefore, we use a handheld LiDAR scanner to obtain the point clouds of the road scene covered by street trees. Our preprocessing step is used to segment the point clouds from an individual tree and remove overlapping point clouds, such as ground points and other objects.

In order to reconstruct trees conveniently, we obtain point clouds of an individual tree by using the existing point clouds processing method [14]. Due to the environmental factor, the point cloud data obtained often has some noise points [15]. Therefore, we also need to process the noise points to establish the basis for modeling realistic 3D tree models. The key idea is to remove outlier points based on the mean and variance of points' coordinates.

Natural trees have complex geometry and topology structures, so the point clouds obtained from the scanning system are massive and unorganized. As we know, a skeleton is a practical way to represent the topological properties of a 3D model. In order to reduce the calculation complexity, we slice and cluster the input point cloud data for searching candidate skeleton points of individual trees. The tree skeleton as shown in Fig. 2(a) is the basis of tree visualization and reconstruction. In this work, the skeleton points of an individual tree are defined at the center of the tree's branches as shown in Fig. 2(b).

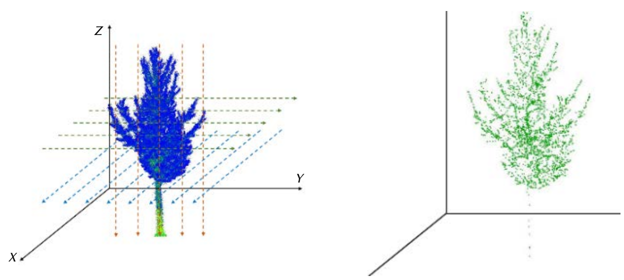
(a) (b)

Fig. 2. Formulation of skeletons. (a) Branches of an individual tree. (b) Skeletons of an individual tree.



(a) threshold=0.06 (b) threshold=0.1 (c) threshold=0.15

Fig. 3. Results using different threshold values.



(a) (b)

Fig. 4. Demonstration of slicing methods and results. (a) The slicing direction for obtaining skeleton points. (b) The skeleton point sets.

Firstly, the candidate skeleton regions are obtained based on the Euclidean clustering. A threshold is set where if several points are neighboring in a cluster, they are merged. This reduces unnecessary calculations. We set the default threshold to 0.1. If the threshold is set too large, the sampled point cloud data is too small, so the tree's topology cannot be correctly extracted. If the threshold is set too small, there are too many initial skeleton points, which increases the calculation amount and fails to improve efficiency. As shown in Fig. 3. Then, we slice points in the direction of the X axis, Y axis and Z axis, respectively. The step size for slicing in each direction is 0.1. We define points where the X axis intersects the Z axis and the points where the Y axis intersects the Z axis as candidate skeleton points. As shown in Fig. 4(a). To obtain more information on trees, we rotate point clouds of an individual tree by using a transformation matrix as shown in Eq. (1).

$$T = \begin{pmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{1}$$

where ϕ is the rotation angle, and we set ϕ as $\frac{\pi}{6}$.

Finally, all candidate skeleton points are merged which serve as initial target tree points as shown in Fig. 4(b).

B. Branch Tracing and Radius Estimation

In the growth perspective of trees, neighboring points tend to be from the same branch. Therefore, we consider the optimal path from each point to the root as the basis for branch tracking. We evaluate the weight of each path from the current point to the root, compare the weight of each path and the path with the lowest weight is regarded as the optimal branch for the current point. As shown in the following Eq.(2)-(4).

$$TD(p_i, l) = min\{ dis(p_i, l), k\}, l \in S(p_i) \tag{2}$$

$$MD(p_i, t_j) = min\{ dis(p_i, t_j), k\} \tag{3}$$

$$Path = \sum_{i=0}^{m} \{Min\{TD(p_i, l), MD(p_i, t_j)\}\} \tag{4}$$

In the formulation, p_i is the current point, t_j is the point in the set for which the optimal branch is not currently found. $S(P_i)$ is the set includes three points which has the closest Euclidean distance to the current p_i. The $dis(p_i, t_j)$ is the Euclidean distance from p_i to t_j. Path is the path index of the obtained branch, and k is a relative large constant.

Algorithm 1. Find optimal branch

Input: The matrix ***M*** is used to store the coordinates of skeleton points.

Output: The optimal vector ***Path***

1: Initialize the value of a candidate branch from root for each point p_i (x, y, z) as $dis_i = 0$;
2: **for** Each $i \in$ ***M*** **do**
3: Calculate the distance between two neighboring points in groups of the nearest q points;
4: Formulate the index of neighboring points into the set $\mathbb{N}$;
5: **end for**
6: Create the set **fb** of all candidate points.
7: Create the set **tb** of all non-candidate points.
8: **while tb** is not NULL
9: **for** Each $i \in$ **fb do**
10: Consider i as the leading point and formulate neighbors of i into the set $\mathbb{N}$;
11: **for** Each $j \in$ **tb do**
12: **if** $j \in \mathbb{N}$ **then return** $Dis \longleftarrow$ ***Dis***(p,j)
13: **end if**
14: $Dis\_plus \longleftarrow Dis + dis$
15: **if** $MIN > Dis\_plus$ **then return** $MIN \longleftarrow Dis\_plus$, $i^* \longleftarrow i$, $j^* \longleftarrow j$
16: **end if**
17: **end for**
18: **end for**
19: $dis_{j*} \longleftarrow MIN$
20: Add j^* into the set **fb**;
21: Delete j^* from the set **pb**;
22: **Path**$(j^*) \longleftarrow i^*$
23: **end while**

The method of computing skeleton branches is shown in **Algorithm 1**. We input the coordinates of the point cloud, calculated the Euclidean distance between points, and selected the spatially closest 10 points to each point to form the matrix $\mathbb{N}$. We defined **fb** as the set of points and its branches have been assigned, and **tb** as the set of points and its branches are to be assigned. The sum of points in **tb** and **fb** are the total number of skeleton points of a tree. Every point in **fb** is taken as the current point, and the points in **tb** are traversed to determine whether they belong to the neighboring points of the current point in ℕ. If correct, the point is the child point of the current point. If not, we found the nearest neighboring point to the current point in **tb** as its child point.

It is worth noting that the number of neighboring points q may depend on the tree species. In terms of arbor forests, they usually have straight and thick stems, thus, the value q tends to be large (in our case we set q as 10). In the case of bamboo forests or shrubs, q should be decreased to capture more details of thin branches, while at the cost of being time-consuming.

The radius of the branch corresponding to the skeleton point is estimated by using empirical formulation or certain rules. Practically, the radius obtained by these methods does not show the true trunk radius. The branches of the tree can be approximated as consisting of many cylinders. Each cylinder has different sizes of radius. Based on this feature, the least squares fitting method is used to estimate the radius of the corresponding branches at the skeleton point. When the number of skeleton points corresponds to less than three, the radius obtained by the least squares fitting method will be inaccurate. It will lead to imprecise reconstruction. Thus, the radius of individual points is reacquired by Eq.(5),

$$r_c = r_p \times \left(\frac{l_c}{l_p}\right)^{\frac{3}{2}} \tag{5}$$

where r_c indicates the radius of the current sub-skeleton point. The r_p indicates the radius of the current parent skeleton point. The l_c is the length of the branch supported by the sub-skeleton point. The l_p is the length of the branch supported by the parent skeleton point. The radius of all skeleton points are obtained by repeating the above-mentioned method.

C. Path Interpolation and Optimization

The tree skeleton visualization is a powerful tool for illustrating the structure of trees, because its ability to describe the topology and geometry of a tree in a simple and dense form [16]. However, complex environment may weaken the roots or stems of trees [17]. Besides, foliage shading causes LiDAR data to not portray the structural variables of trees in detail [18], so there are still errors in the obtained skeleton point clouds. These errors create gaps between skeleton points, leading to incoherent skeleton connections and affecting the modeling results, so we interpolate and optimize the tree skeleton. Firstly, we interpolate each branch to ensure obtaining a more complete tree skeleton structure. According to the Euclidean distance between two adjacent points, a point is interpolated for every 0.1m in the space, and there is a total of six directions as shown in matrix $\boldsymbol{M}$ of Eq. (6). Equation 6 represents the positive direction of the X axis, the positive direction of the Y axis, the positive direction of the Z axis, the negative direction of the X axis, the negative direction of the Y axis, and the negative direction of the Z axis in the three-dimensional coordinate.

$$M^T = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \tag{6}$$

The radius of the interpolation point is assigned according to the number of interpolation points and the distance between the two points as shown in Eq. (7).

$$r_i = \frac{(r_c - r_p)}{m} \times i + r_c \tag{7}$$

In Eq.(7), n is the number of interpolation points, r_c indicates the radius of the current point, r_p indicates the radius of the previous point, and i indicates that the current point is the *ith* interpolation point between two points.

Then, we use the Laplacian smoothing method to smooth the skeleton, as shown in Eq. (8), which makes the lines between skeleton points smoother and consistent with the bending condition of real tree branches.

$$\mathrm{Point}_i' = \mathrm{Point}_i + \alpha\left(\frac{\mathrm{Point}_{i\text{-}1} + \mathrm{Point}_{i+1}}{2} - \mathrm{Point}_i\right) \tag{8}$$

where $Point_{i-1}$ and $Point_{i+1}$ are two points before and after $Point_i$, α is the coefficient of the Laplacian smoothing method.

D. Branch Modeling

A cylinder can be regarded as a combination of a circular cross-section and a central axis perpendicular to the cross-section. The natural branches of trees are usually cylindrical bodies of different thicknesses. However, the cylindrical shape makes the tree trunk incoherent in modeling. Therefore, we propose a new model using geometrical shapes. Based on the complete skeleton generated, we take each skeleton point as the center of a circle. We define the direction vector from a point in the skeleton to its parent as the local z-axis as shown in Fig. 5(a). A series of discrete points on the cross-section circle are constructed according to the radius parameters. As calculated in Eq. (9),

$$\begin{cases} x = R \cdot \cos\theta \\ y = R \cdot \sin\theta \\ z = 0 \end{cases} \tag{9}$$

where R is the radius of the skeleton point and θ is the angle between the polar axis and the X axis in polar coordinates. The discrete points are transformed by rotation in the skeleton direction to obtain absolute coordinates.

Fitting the tree by using geometrical shapes can address the problem of gaps between branches. The effect is shown in Fig.5 (b). Finally, we use the quadrilateral mesh surface to reconstruct the geometric model of the tree.

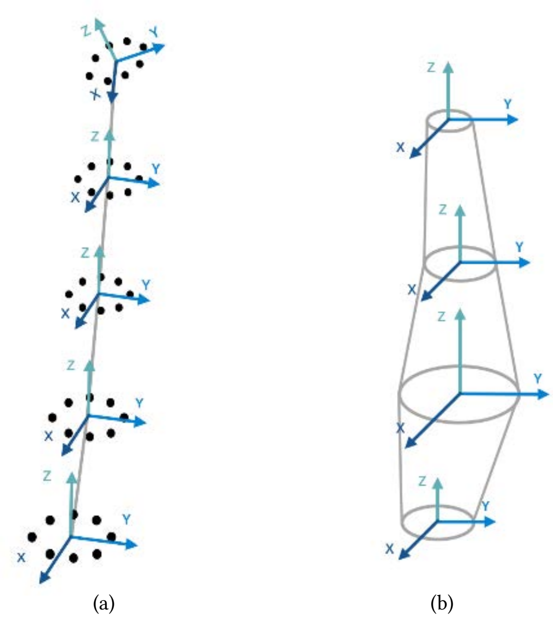


Fig. 5. Local coordinate system. (a) shows the cross-sectional circle, (b) shows the geometrical shapes fitting.

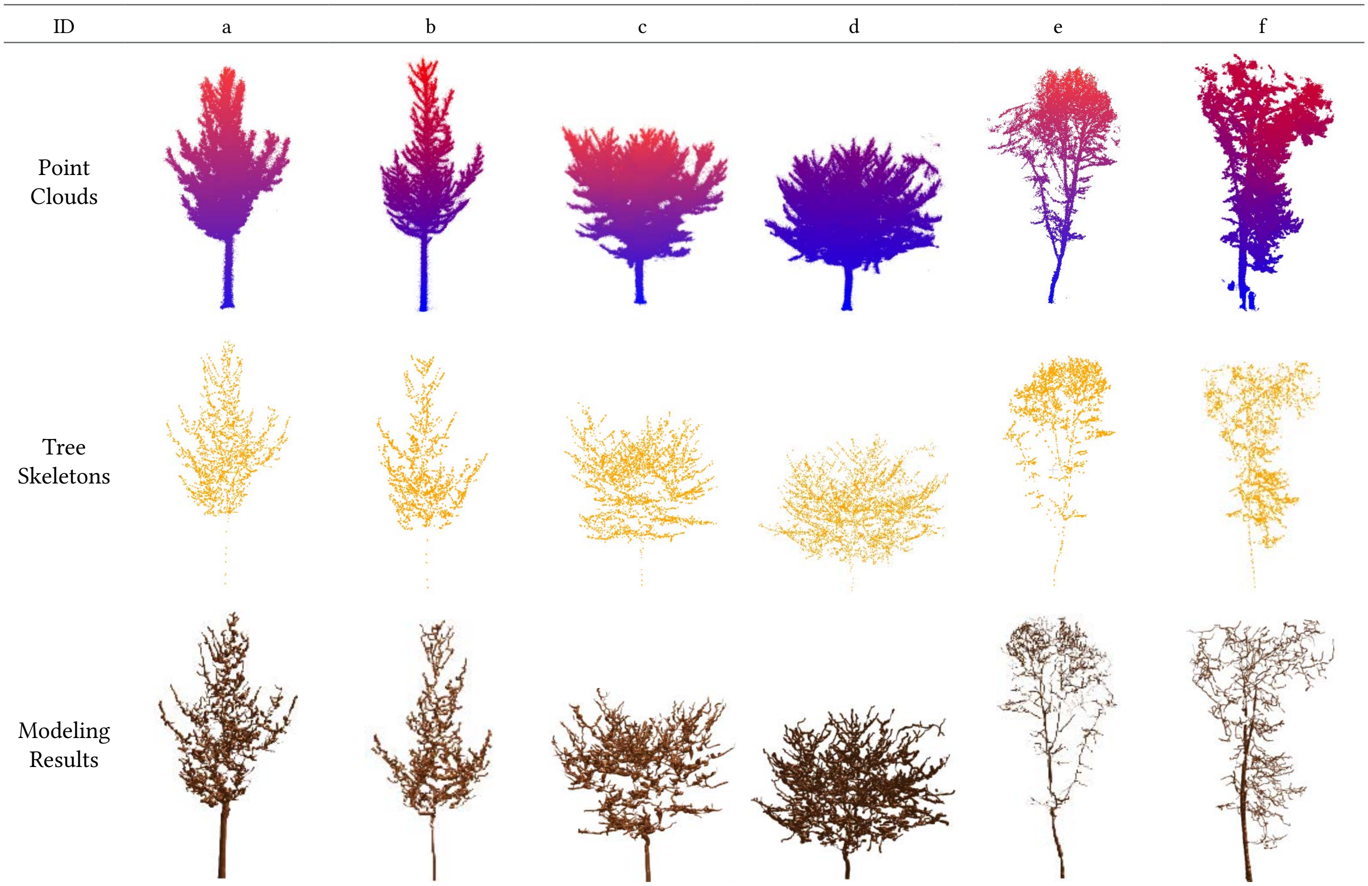


Fig. 6. Results of individual tree modeling. (a) and (b) show the Ginkgo. (c) and (d) show the Prunus. (e) and (f) show the Platanus.

IV. Experiments and Analysis

This section tests the proposed method. This experiment is carried out on the Windows 10 operating system, with a memory size of 64GB. We utilize a CPU powered by Intel(R) Xeon(R) W-2145 @3.70GHZ. All experiments as well as analysis are run on MATLAB R2019b. Experimental data are collected by GeoSLAM ZEB-HORIZON. This portable laser scanner is flexible and efficient, which scans points at 300,000 points per second at the field of view 360◦×270◦. The data collection is convenient and brings fewer point registration errors. Although our data are collected from the side-view, our system scanner distance can be over 100 meters, which provides abundant information for our modeling. The scanner is simple to use and provides easy access to collect the points of street trees. The input scene is located at Nanjing Forestry University, Nanjing, China. In this part, we qualitatively and quantitatively assess the proposed strategy.

A. Visualization

This part focuses on evaluating our method using an artificial visual approach. We display the results of Ginkgo, Prunus and Platanus. The first column shows the original point clouds of investigated individual trees, the second column shows the skeleton of tree point clouds, and the third column shows our modeling results. As shown in Fig. 6, we display a better tree morphology evidenced by the completeness and correctness of reconstructions. Technique used in this research is capable of modeling a variety of tree species and is generalizable to trees with various geometries and typologies.

B. Ablation Experiments

To verify the validity of our optimization and modeling, we have made the following two sets of ablation experiments. The first set of experiments compares the optimized reconstruction with the non-optimized reconstruction as shown in the figure. We notice that the non-optimized branches are not modeled smoothly and the results do not conform to the growth pattern of trees in nature as shown in Fig. 7. The optimized modeling has a natural bend in branches, which shows the tree's topology in detail. We find branches fitted into incoherent cylinders depending on the location of the points and the size of the radius.

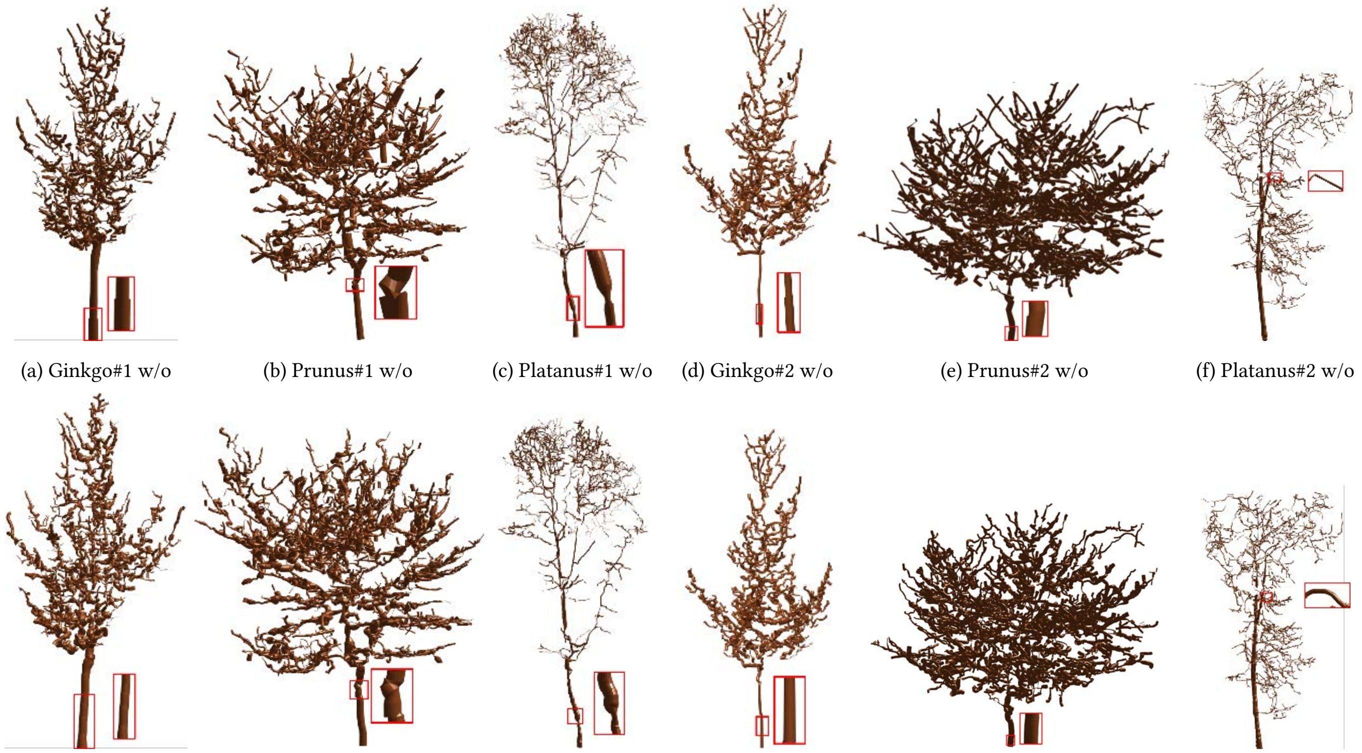
In the second set of experiments, we used different primitives for modeling. Fig. 8 (a-f) shows the modeling by using cylindrical primitives. Fig. 8 (g-l) shows the modeling by using geometrical shapes. The results show that there is a gap between cylinders, which does not allow for accurate modeling. Modeling with cones as the primitive generates a high-quality individual tree model.

C. Quantitative Analysis

We use Eq.(10) to calculate the accuracy of the output model, including the overlapping of the point clouds to the individual 3D branch model. In the evaluation, the accuracy is calculated by the ratio of correctly extracted points as shown in Eq.(10).

$$Accuracy = \frac{1}{m} \cdot \sum_{j=1}^{m} \delta(p_i, p_j) \tag{10}$$

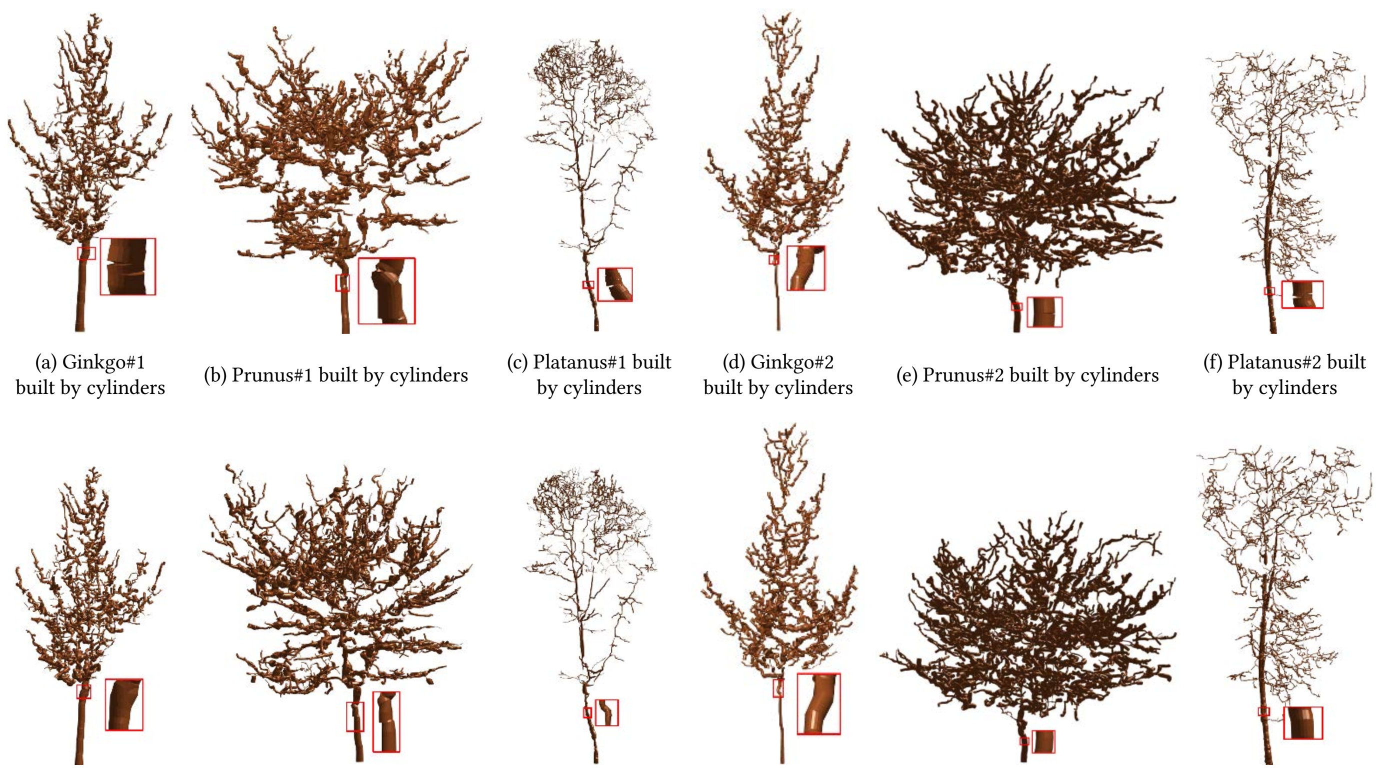
where m is the number of model points. p_i indicates the points of the tree model, and p_j indicates the input points. δ (p_i, p_j) is a binary function. If the Euclidean distance between p_i and p_j is less than accuracy, δ is 1, otherwise, it turns out to be 0. Table I shows that the average completeness of the experiment reached 97.49%.



(a) Ginkgo#1 w/o (b) Prunus#1 w/o (c) Platanus#1 w/o (d) Ginkgo#2 w/o (e) Prunus#2 w/o (f) Platanus#2 w/o

(g) Ginkgo#1 w/ (h) Prunus#1 w/ (i) Platanus#1 w/ (j) Ginkgo#2 w/ (k) Prunus#2 w/ (l) Platanus#2 w/

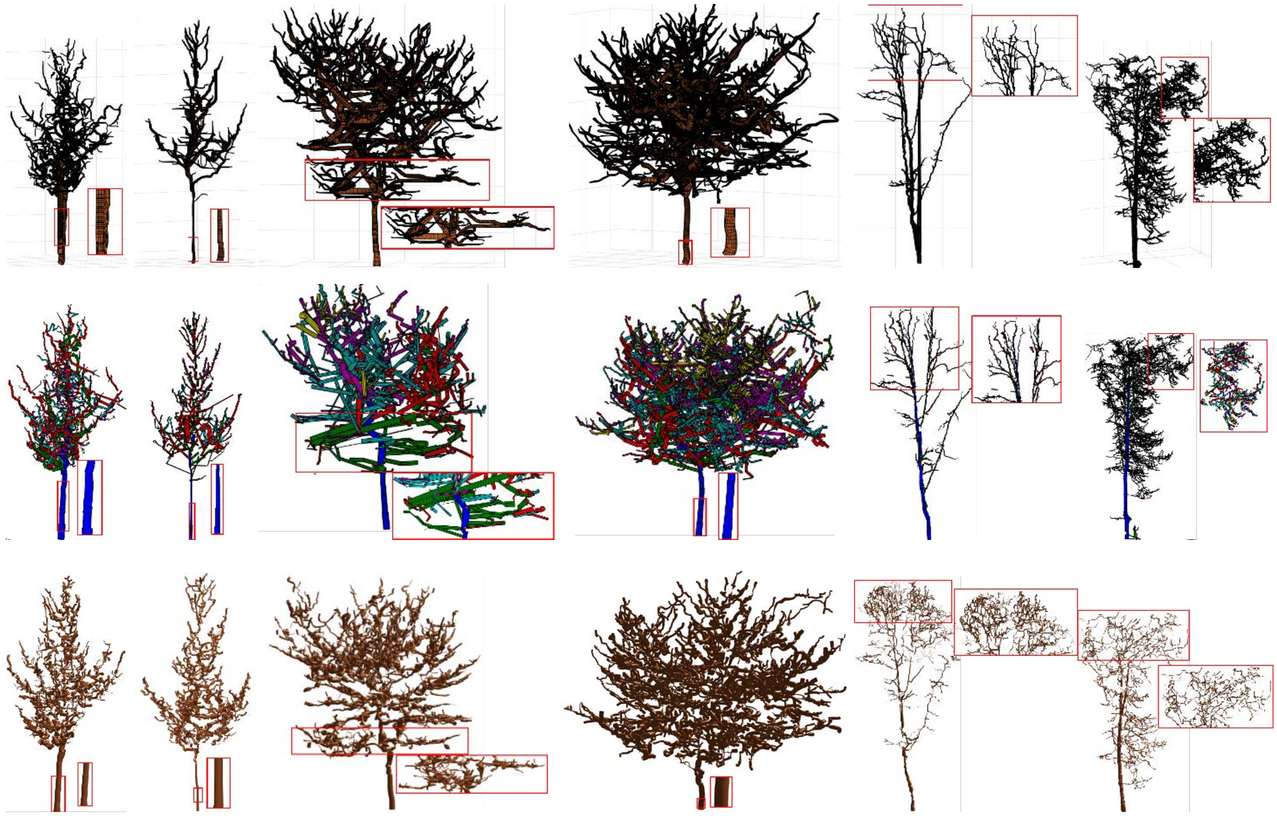
Fig. 7. Comparison of results before and after optimization. (a-f) show the reconstruction results without being optimized (without optimization, w/o). (g-l) show the reconstruction results with being optimized (with optimization. w/).



(a) Ginkgo#1 built by cylinders (b) Prunus#1 built by cylinders (c) Platanus#1 built by cylinders (d) Ginkgo#2 built by cylinders (e) Prunus#2 built by cylinders (f) Platanus#2 built by cylinders

(g) Optimized Ginkgo#1 (h) Optimized Prunus#1 (i) Optimized Platanus#1 (j) Optimized Ginkgo#2 (k) Optimized Prunus#2 (l) Optimized Platanus#2

Fig. 8. The modeling results with different geometric structures. (a-f) show the reconstruction by using cylinders. (g-l) show the reconstruction results by using the proposed geometrical shapes.



(a) Ginkgo#1 (b) Ginkgo#2 (c) Prunus#1 (d) Prunus#2 (e) Platanus#1 (f) Platanus#2

Fig. 9. The modeling results by using different ways.

TABLE I. Quantitative Results for Ginkgo, Prunus and Platanus Trees

ID	Points	Species	Complexity	Accuracy
a	945333	1	M	**97.69%**
b	632514	1	M	**99.52%**
c	1972532	2	D	**97.64.%**
d	2046869	2	D	**99.67%**
e	93768	3	M	**92.53%**
f	95923	3	M	**97.92%**

D. Comparisons

This section shows the comparison of our method and other commonly used approaches to further show our advantages and disadvantages.

Firstly, we show the comparison of our method with the Circular Truncated Cones Tree [19] and TreeQSM [20]. As shown in Fig. 9, the Circular Truncated Cones Tree (CT Cones Tree) utilizes optimized circular frustum to create the three-dimensional structure of skeletons. It partitions the tree into small blocks after preprocessing and interpolates directly. TreeQSM employs cylinders of various sizes for each branch, which are mostly straight. Thus, the reconstructed 3D model lacks realism. However, we find the best path for each point cloud, and use the Laplace algorithm for smoothing operation. Our approach ensures that the model is according to the tree growth, and is complete in the final model presentation.

Secondly, we show the quantitative comparison. We use Eq.(9) to calculate the accuracy of tree reconstructions by using CT Cones Tree, TreeQSM and our method. Table II demonstrates that the accuracy of our results is more than 90%, and better than CT Cones Tree and TreeQSM. Therefore, more accurate geometric and topological structures can be obtained by using our modeling method.

In the case of the Platanus (e and f of Table II), the stem located centrally is clearly visible based on the clustered centers, and the majority of extended branches are concentrated toward the tree's apex. The slender branches possess a low point cloud density, which suggests that TreeQSM may not be fully applicable. Our method, however, determines the optimal path for each point, utilizes its position information in the tracking process, and results in more realistic modeling. Consequently, the accuracy of Platanus (e and f) is significantly enhanced.

In table I, the 'ID' indicates the sequence number of trees and corresponds to the result in Fig. 6. The 'Points' indicates the input data. The 'Species' includes Ginkgo (*Ginkgo biloba L.*), Prunus (*Prunus yedoensis Matsum.*), and Platanus (*Platanus orientalis L.*). 'M' means moderate modeling complexity and 'D' means difficult modeling complexity. Our ranking system for M and D is determined by the number of original point clouds. Point clouds exceeding 100,000 belong to category D, whereas those between 10,000 and 100,000 belong to category M.

TABLE II. Quantitative Results for Ginkgo, Prunus and Platanus Trees

ID	Points	CT Cones Tree	TreeQSM	Ours	Increase
a	945333	97.28%	92.39%	**97.69%**	+0.41%
b	632514	92.31%	92.06%	**99.52%**	+7.21%
c	1972532	95.04%	81.19%	**97.64.%**	+2.6%
d	2046869	93.97%	90.77%	**99.67%**	+5.7%
e	93768	64.02%	39.43%	**92.53%**	+28.51%
f	95923	92.96%	53.64%	**97.92%**	+4.96%

E. Limitations and Future Works

Our technique produces a precise 3D tree model. However, there are still problems that require additional study.

In the data collection, in order to reduce the impact of different types of foliage on the modeling, we treat the foliage as noise, because the shape of foliage is quite different and presents especially blur in the collection caused by wind. One simple way is to use a fixed template to directly paste foliage in dense points around trunks. However, it does not demonstrate the correct tree structures and it is difficult to evaluate.

In the modeling process of an individual tree, it is easy to model the main trunk, and it is difficult to model multilevel branches, because there are some small gaps at the connection between the branches. In addition, due to the limited modeling methods, some branches with a large degree of curvature are difficult to be completely restored occasionally. These challenges will need to be further addressed in future works. The 3D single tree model reconstructed in this paper be extended and applied to scenes of rapid reconstruction of large trees, providing reliable model support for digital cities and virtual forestry. Besides, future work will also focus on the extension of our work on different species of trees, especially on the fusion of drone point clouds and handheld point clouds to improve the top-view modeling for tall trees.

V. Conclusion

Considering trees as one of the most significant elements of the natural world, it is significant to study 3D trees precisely and effectively. We propose a method to obtain skeleton branches and then reconstruct 3D tree models. Points are sampled by clustering, and then the data obtained from the slices are used as our skeleton points. To ensure the local optimum in modeling, we track the whole skeleton for searching different branches and estimating the radius of skeleton points by the least squares fitting method. The skeleton points are interpolated in different directions, which guarantees the curvature of tree branches is accurate. The experimental results show that the average accuracy of individual tree data is up to 97.49% based on the proposed geometrical shapes method, which provides a promising approach for 3D tree modeling.

Acknowledgment

This work was supported by the STI 2030-Major Projects (Grant NO.2023ZD0405605) and the Graduate Research and Innovation Projects of Jiangsu Province (Grant No. SJCX23_0320).

References

[1] C. Sun, C. Huang, H. Zhang, B. Chen, F. An, L. Wang, and T. Yun, "Individual Tree Crown Segmentation and Crown Width Extraction From a Heightmap Derived From Aerial Laser Scanning Data Using a Deep Learning Framework" *Frontiers in plant science*, vol. 13, 2022, doi: https://doi.org/10.3389/fpls.2022.914974

[2] M. Schmitt, M. Shahzad, and X.X. Zhu, "Reconstruction of individual trees from multi-aspect TomoSAR data," *Remote Sensing of Environment*, vol. 165, pp. 175-185, 2015.

[3] G.Z. He, "The Trees Skeleton Extraction Based on Point Cloud Contraction," *Applied Mechanics and Materials*, vol. 475, pp. 355-360, 2014.

[4] E. Bournez, T. Landes, M. Saudreau, P. Kastendeuch, and G. Najja, "From TLS point clouds to 3D models of trees: a comparison of existing algorithms for 3D tree reconstruction," in *3d Virtual Reconstruction and Visualization of Complex Architectures*, vol. 42, no. 2, pp. 113-120, 2017.

[5] R. Maalek, D.D. Lichti, R. Walker, A. Bhavnani, and J.Y. Ruwanpura, "Extraction of pipes and flanges from point clouds for automated verification of pre-fabricated modules in oil and gas refinery projects," *Automation in Construction*, vol. 103, pp. 150-167, 2019.

[6] S. Xu, X. Li, J.Y. Yun, and S.S. Xu, "An Effectively Dynamic Path Optimization Approach for the Tree Skeleton Extraction from Portable Laser Scanning Point Clouds," *Remote Sensing*, vol. 14, no. 1, pp. 94, 2022.

[7] A. Verroust, and F. Lazarus, "Extracting skeletal curves from 3D scattered data," in *Proceedings Shape Modeling International '99. International Conference on Shape Modeling and Applications, Aizu-Wakamatsu, Japan*, pp. 194-201, 1999, doi: 10.1109/SMA.1999.749340.

[8] H. Xu, N. Gossett, and B. Chen, "Knowledge and heuristic-based modeling of laser-scanned trees," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 4, 2007, doi: 10.1145/1289603.1289610.

[9] Y. Livny, F. Yan, M. Olson, B. Chen, H Zhang, J El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," in *ACM SIGGRAPH Asia 2010 papers (SIGGRAPH ASIA '10)*. Association for Computing Machinery, New York, NY, USA, 2010, Article 151, pp. 1–8. https://doi.org/10.1145/1866158.1866177.

[10] N. Pfeifer, B. Gorte, and D. Winterhalder, "Automatic reconstruction of single trees from terrestrial laser scanner data," in *Proceedings of the 20th ISPRS Congress*, Istanbul, Turkey, 2004, vol. 35, pp. 114-119, 2004.

[11] J. Li, H. Wu, Z. Xiao, and H. Lu, "3D modeling of laser-scanned trees based on skeleton refined extraction," *International Journal of Applied Earth Observation and Geoinformation*, vol. 112, pp. 102694, 2022, doi: 10.1016/j.jag.2022.102943.

[12] A. Jiang, J. Liu, J. Zhou, and M. Zhang, "Skeleton extraction from point clouds of trees with complex branches via graph contraction," *The Visual Computer*, vol. 37, pp. 2235–2251, 2021, doi: 10.1007/s00371-020-01983-6.

[13] Q. Li, X. Li, Y. Tong, X. Liu, "Street tree crown detection with mobile laser scanning data using a grid index and local features," *PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science*, vol. 90, no. 3, pp. 305-317, 2022.

[14] S. Xu, S.S. Xu, N. Ye, and F. Zhu, "Automatic extraction of street trees' nonphotosynthetic components from MLS data," *International Journal of Applied Earth Observation and Geoinformation*, vol. 69, pp. 64-77, 2018.

[15] X.D. Hu, C.H. Hu, J.G Han, H. Sun, and R. Wang, "Point cloud segmentation for an individual tree combining improved point transformer and hierarchical clustering," *Journal of Applied Remote Sensing*, 2023.

[16] Y.F. Xu, C.H. Hu and Y.N. Xie, "An improved space colonization algorithm with DBSCAN clustering for a single tree skeleton extraction," *International Journal of Remote Sensing*, vol. 43, pp. 3692-3713, 2022.

[17] B. Zhang, X.J. Wang, X.Y. Yuan, F An, H.Q. Zhang, L Zhou, J Shi, and T Yun, "Simulating Wind Disturbances over Rubber Trees with Phenotypic Trait Analysis Using Terrestrial Laser Scanning," *Forests*, vol. 13, no. 8, pp. 1298, 2022.

[18] X. Xue, S. Jin, F. An, H. Zhang, J. Fan, M.P. Eichhorn, C. Jin, B. Chen, L. Jiang, and T. Yun, "Shortwave Radiation Calculation for Forest Plots Using Airborne LiDAR Data and Computer Graphics," *Plant Phenomics*, 2022.

[19] X. Li, X. Zhou, and S. Xu, "Individual Tree Reconstruction Based on Circular Truncated Cones From Portable LiDAR Scanner Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 20, pp. 1-5, 2022.

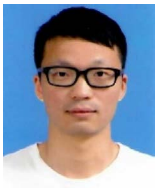
[20] P. Raumonen, M. Kaasalainen, M. Åkerblom, S Kaasalainen, H. Kaartinen, M. Vastaranta, M. Holopainen, M. Disney, and P. Lewis, "Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data," *Remote Sensing*, vol. 5, 2013.

Xin Li

Xin Li graduated from Sanjiang University with a bachelor's degree in Software Engineering in 2021. During her undergraduate years, she was awarded 'First Class Sholarship' for three times. Besides, she was elected as 'Outstanding Student'. Now she is pursuing for a master's degree in the College of Information Science and Technology, Nanjing Forestry University. Her research interests include Computer Vision and 3D Modeling.

Xuan Zhou

Xuan Zhou received her bachelor degree in software engineering in 2021 from Audit University, Nanjing, China. During four years of her undergraduate study, she won the first-class scholarship and was awarded the honorary title of 'Excellent Graduates'. Now, she is studying for the master degree at the College of Information Science and Technology at Nanjing Forestry University. Her research interests include Computer Vision and Point Cloud Processing.



Sheng Xu

Sheng Xu received his Ph.D degree from the University of Calgary in 2018, He joined the College of Information Science and Technology at Nanjing Forestry University. At present, he has published more than 40 papers on international and domestic journal conferences. In recent five years, he has published more than 10 SCI papers on IEEE Trans. TPAMI, IEEE Trans. TIST, IEEE Trans. His research interests include Three-dimensional spatial information processing, point cloud data analysis and computer vision.