remote sensing

MDPI

Article

A High-Precision Three-Dimensional Branch Reconstruction Method for Tree Morphological Phenotype

Tingdong Yang [1,2,3,4], Huaiqing Zhang [1,2,3], Yili Zheng [4,5,*], Xueyan Zhu [1,2,3] and Jie Yang [6]

[1] Institute of Forest Resource Information Techniques, Chinese Academy of Forestry, Beijing 100091, China; yangtd@ifrit.ac.cn (T.Y.); zhang@ifrit.ac.cn (H.Z.); xueyan0111@ifrit.ac.cn (X.Z.)
[2] National Forestry and Grassland Science Data Center (NFGSDC), Beijing 100091, China
[3] Key Laboratory of Forestry Remote Sensing and Information System, National Forestry and Grassland Administration, Beijing 100091, China
[4] College of Technology, Beijing Forestry University, Beijing 100083, China
[5] National Key Laboratory-Forest Resource Efficient Production, Beijing 100083, China
[6] College of Forestry, Beijing Forestry University, Beijing 100083, China; nj_yang_jie@bjfu.edu.cn
* Correspondence: zhengyili@bjfu.edu.cn

Highlights

What are the main findings?

- We enhanced skeleton extraction with LOF-based clustering and a seven-stage skeleton refinement scheme for accurate reconstruction at the furcation and other details.
- We combined FAST-LTS to implement a self-adaptive WBE model and achieved radius estimation at the level of the individual tree.

What is the implication of the main finding?

- Technically, by tolerating non-uniform density and noise, the method yields robust, non-destructive 3D trunk and branch reconstruction from a single field scan, avoiding the cross-section distortion and volume inflation of conventional QSM.
- Practically, calibration-free per-tree radius estimation permits repeatable in situ phenotyping across individuals, supplying an automated data basis for breeding-trait evaluation, FSPM, and precision forest management.

Abstract

The three-dimensional architecture of the trunk and branch system of individual trees constitutes essential information for tree breeding trait evaluation, the construction of functional-structural plant models (FSPM), and precision forest management; its morphological traits are typically represented by diameter at breast height (DBH), tree height, branch order, and the 3D branching topology. In recent years, point-cloud-based tree structure parsing has advanced rapidly. However, constrained by field acquisition conditions and sensor principles, tree point clouds commonly exhibit missing data and noise, causing locally non-uniform density and introducing noise interference. Under these conditions, traditional QSM (Quantitative Structure Models) tend to produce cross-section distortion and inflated volume, degrading both modeling accuracy and the reliability of parameter extraction; moreover, their radius estimation usually relies on species-specific calibration, limiting cross-species application. To address these limitations, this study proposes a high-precision three-dimensional reconstruction method for the trunk and branch system to extract tree morphological phenotypes, which achieves robust reconstruction under noise and missing data through density-adaptive clustering and biologically mechanistic self-adaptive WBE radius simulation. Experimental results show that, compared with existing mainstream methods, the proposed method yields better modeling performance and stronger robustness on both artificial and real branch point clouds, reaching centimeter-level accuracy.

Check for updates

Academic Editor: Brenden E. McNeil

Received: 11 September 2026
Revised: 28 October 2026
Accepted: 17 November 2026
Published: 20 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: tree modeling; point cloud data; segmentation-based skeletonization; QSM; WBE model

1. Introduction

Accurate characterization of the three-dimensional (3D) structure of individual trees is of great significance for tree-breeding trait evaluation, the construction of functional–structural plant models (FSPM), and precision forest management [1–3]. Traditional research methods rely on destructive manual measurement of felled trees, which precludes repeated in situ observation of the same tree and yields only discrete, local parameters. At present, terrestrial laser scanning (TLS) can acquire relatively complete trunk surface information [4], and tree point cloud data have become an important data source for 3D tree reconstruction and structural analysis. The quantitative structure model (QSM), which performs 3D modeling using multiple topologically organized generalized cylinder models (GCMs), is consistent with the traditional idea of treating trees as regular geometric objects (particularly cylinders or truncated cones) for tree measurement, and has become the mainstream reconstruction method for 3D tree modeling [5].

However, field point cloud data commonly exhibit noise and localized missing data, leading to locally non-uniform point density; under such conditions, traditional QSM methods are prone to cross-sectional distortion and volume overestimation, which constrains both modeling accuracy and the reliability of parameter extraction.

The production of GCMs can be concluded into three steps: skeleton extraction, skeleton refinement and modeling [6]. Among them, different policies were adopted in the phase of skeleton refinement in different studies without unity. And modeling is almost a paradigm. Therefore, the classification of different tree branch modeling algorithms is based on the category of skeleton extraction, although the classification criteria vary as well [1,2,6]. After comprehensive consideration, the related studies can be reasonably divided into four categories: voxel-based skeletonization [7], contraction-based skeletonization (including Laplacian-based contraction [8–11] and L1 contraction [12–16], segmentation-based skeletonization [17–21] and graph-based skeletonization [22,23]. In some ways, the former two categories can be further grouped into one category named thinning-based skeletonization [24].

While 3D crop reconstruction commonly adopts contraction based skeletonization in agriculture [25], more applications in forestry use segmentation-based or graph-based skeletonization, mainly including TreeQSM (Tree Quantitative Structure Modeling) [18], AdTree (Accurate, Detailed, and Automatic Modeling of Laser-Scanned Trees) [22], AdQSM (Accurate and Detailed Quantitative Structure Model) [23], Yang's method [20,21] and implementation based on Xu et al. [17]. Among them, AdQSM is a comprehensive improvement of TreeQSM and AdTree. Yang's method shares a common idea with AdQSM that Single-Source Shortest Paths (SSSP) are regarded as an ideal policy for trees to transport water and nutrients [26]. Due to different algorithm policies, the output of one method is not always better than the others [21].

In this study, we propose a high-precision three-dimensional branch reconstruction method for tree morphological phenotypes. It belongs to segmentation-based skeletonization. When obtaining the segmentation criterion, both the number of neighbors and reachable distance are considered to reduce the unreasonable edges with different point cloud densities in the graph constructed under a single constraint. On the basis of existing contextual algorithms, skeleton extraction replaces the shallow-recursion iterative method with a fully iterative one, and introduces the LOF algorithm into clustering algorithms to achieve reasonable clustering of different point cloud densities. In terms of skeleton

refinement, seven skeleton refinement methods were integrated to remove unreasonable branching, topological distortion, and node redundancy in the initial skeleton. Specifically, the proposed axis-constrained least-square cylinder fitting algorithm was applied to skeleton refinement at bifurcations and branch segments. In terms of radius estimation, a self-adaptive WBE model was proposed, which utilizes the FAST-LTS algorithm to construct a WBE model that can adapt to the individual tree level. For validating the visual and quantified performance of our method, we conducted the experiments using nine tree branches and comparing our method to AdQSM and Yang's method. Finally, we compared the numerical deviation of Diameter at Breast Height (DBH) and center deviation at breast height among these methods. Each of these components addresses one manifestation of the problem stated above: the dual-criterion graph construction and the density-adaptive clustering act on the non-uniform density itself, whereas the refinement at bifurcations and branch segments and the robust radius estimation act on the cross-sectional distortion and volume overestimation that non-uniform density produces.

In brief, our main contributions encompass the following aspects:

- We propose a segmentation criterion that combines neighbor count with reachable distance to exploit their complementary advantages for better control of edge construction;
- We improve the iterative contextual skeleton extraction algorithm, and introduced the LOF algorithm to achieve more reasonable clustering for distinguishing branches;
- We employ a seven-stage skeleton refinement scheme, and propose an axis-constrained least-squares cylinder-fitting algorithm for more accurate local skeleton reconstruction at bifurcations and branch segments;
- We combine FAST-LTS to implement a self-adaptive WBE model and achieve radius estimation at the level of the individual tree;
- The proposed method improves the accuracy of three-dimensional reconstruction of tree branches to the centimeter level when processing point clouds from multiple sources, and achieves more accurate extraction of breast height position and breast diameter values.

The article is organized as follows. In Section 2, each step of the tree modeling methods is introduced in detail, including the segmentation criterion, skeleton extraction, skeleton refinement, radius estimation, branch order calculation, and model construction. In Section 2.1, we introduce the materials and experimental environment. Sections 3.1 and 3.2 respectively compare the overall model from the perspectives of visual evaluation and quantitative analysis. Section 3.3 further compares several model details to evaluate the robustness. Section 3.4 takes the location at breast height as an example to analyze the numerical accuracy of DBH and center deviation. From Sections 3.1–3.4, we compare our method with AdQSM and Yang's method, two newer open-source structured modeling methods for tree branches.

2. Materials and Methods

Figure 1 shows the overview of the methodology flow. The input, specifically the point cloud of the tree branch, should contain as much comprehensive surface information as possible on the branch, as pointed out by Cao et al. [1].

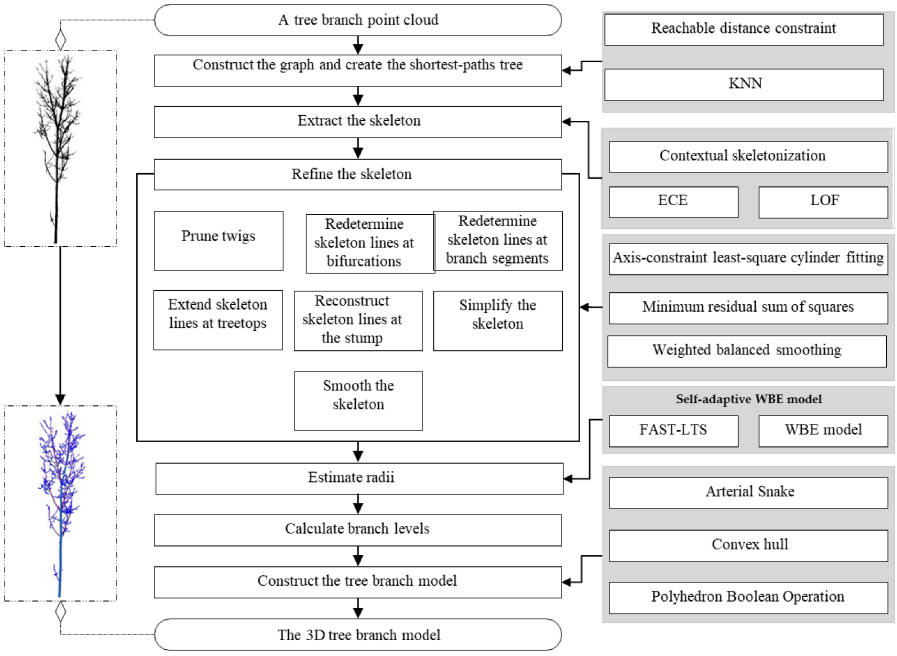


Figure 1. The methodology flow. The input is the point cloud of tree branches, and the output is the corresponding reconstructed 3D models. The gray rectangular area is the main improvement content or important module involved in a step.

2.1. Data Acquisition

In this study, we used a total of nine single tree branch point clouds obtained from multiple sources.

The Masson pine and Tung tree were collected in Shanxia forest farm (27°45′N, 114°39′E), Fenyi County, Xinyu. Fenyi county is at the central-western part of Jiangxi Province, southeastern China. The terrain is mainly composed of Jiangnan hills and mountains, with an altitude of 60–1091.8 m. It belongs to a warm and humid monsoon climate in the central subtropical region, with four distinct seasons, an average annual temperature of about 17.2 °C, an annual precipitation of about 1600 mm, and a frost-free period of 270 days throughout the year. Fenyi County has abundant forest resources, with a forest coverage rate of 64.3%. The main forest types include coniferous forests, mixed coniferous and broad-leaved forests, evergreen broad-leaved forests, mixed evergreen and deciduous broad-leaved forests, bamboo forests, dwarf forests, and shrubs. Models of the two trees, produced manually by SpeedTree, were adopted to produce the surface point clouds using HELIOS++ [27] and the ground part was removed using a Cloth Simulation Filter (CSF) [28].

Beech, Ash, and Oak point clouds were sourced from publicly available datasets (used in Yang [21]), which were collected from a mixture of multiple samples across Germany and scanned using various TLSs.

Four poplars were collected in the Malanghu Forest Farm (33°20′N, 118°19′E), Sihong County, Suqian, Jiangsu Province. The location is on the west bank of Hongze Lake, and belongs to both the East Asian monsoon zone and the transitional zone between the North Subtropical and North Warm Temperate zones. They are all clones of Populus deltoides and were scanned using a RIEGL VZ-400i. The LiDAR data were also from the aforementioned existing research. In Section 3.4, we also use a poplar sample plot in the same study area and with same scanning equipment. All laser scanner data were subjected to multi-station splicing, denoising, ground normalization, tree segmentation, and wood-leaf separation, resulting in the tree branch point clouds used in the experiment. These nine point clouds

were not selected to represent species; they were chosen so that the noise and localized gaps that make field density non-uniform are represented in the test set, spanning sparse and defective scans as well as dense high-quality ones. Table 1 shows some detailed information about all the point clouds used.

Table 1. Basic information on experimental materials. The situation is manually interpreted based on point clouds. "pts" refers to the number of a point cloud.

Tree Species	Scanner	Location	Pts	Situation
Masson pine	(Simulated) RIEGL VZ-400	Xinyu, China	73,703	Good quality
Tung tree	(Simulated) RIEGL VZ-400	Xinyu, China	22,290	Good quality
Beech	Zoller and Fröhlich Imager 5006	Unknown, Germany	167,629	Missing back
Ash	Zoller and Fröhlich Imager 5006	Unknown, Germany	94,021	Good quality
Oak	Faro Focus 3D 120	Unknown, Germany	819,556	Good quality
Poplar(1)	RIEGL VZ-400i	Suqian, China	302,120	Canopy ghosting
Poplar(2)	RIEGL VZ-400i	Suqian, China	182,394	Canopy ghosting
Poplar(3)	RIEGL VZ-400i	Suqian, China	142,016	Canopy ghosting
Poplar(4)	RIEGL VZ-400i	Suqian, China	274,688	Canopy ghosting

2.2. Construction of the Graph and Creation of the Shortest-Paths Tree (SPT)

Three methods can be used to determine edges to construct the graph of a point cloud—k-Nearest Neighbor (k-NN) search, radius search and Delaunay tetrahedralization. However, each of them has disadvantages. For collected point cloud data without uniform density distribution, both k-NN search and radius search are too rigid. The former leads to too many unreasonable local long edges in the low-density area, and the latter has the exact opposite effect [24]. Moreover, when the radius of the latter is set too small, the graph is disconnected while the number of edges connected in high-density areas increases sharply and consequently the memory is likely to be insufficient in spite of a slight increase. Some recent studies preferred to use Delaunay tetrahedralization. However, for this research, when the graph is constructed by this method, some skeleton lines pass through areas without point clouds. Because field clouds are not uniformly sampled, a graph built under a single spatial constraint either connects distant points in sparse regions or breaks apart in dense ones; the criterion below therefore fuses the number of neighbors with the reachable distance.

The method of graph construction in this research considers both the number of neighbors and reachable distance. The segmentation criterion is based on the SSSP distances on the constructed graph. Specifically, the steps to obtain the segmentation criterion can be summarized as follows:

Step 1: Let the maximum inter-point spacing $= \min_{\substack{q \in \varphi \\ p \neq q}} \|p - q\|_2$ where φ is the tree branch point cloud, $\varphi \in \mathbb{R}^3$, and the number of allowed maximum neighbors k ($k \geq 3$). Let d_{max} denote the maximum allowed inter-point spacing in the tree-branch point cloud φ, and let k ($k \geq 3$) be the maximum number of neighbors considered for each point. The threshold dmax is defined as a multiple of the average nearest-neighbor distance of φ, so that the graph construction adapts to the local point density: in sparse regions it still permits connection to the nearest points, while in dense regions it bounds the number of connected edges and avoids memory blow-up. The lower bound $k \geq 3$ guarantees that even low-density points can find connectable neighbors, whereas the d_{max} constraint prevents over-connection in dense regions.

Step 2: Supposing that points are not distributed uniformly, in high-density areas, the number of neighbors of a point p with distances less than d_{max} is far more than k, whereas in

low-density areas, points are almost exclusively connected to their nearest points. For each point p, use radius search to find all neighbors with distances less than d_{max} and compose the set $\mathcal{N}_{r \le d_{max}}(p)$. Choose the $\min\left|\mathcal{N}_{r \le (p)}, k\right\}$-nearest neighbors to connect edges of the undirected graph G of the point cloud, where each point corresponds to a unique vertex.

Step 3: At this time, G could be disconnected. To solve this problem, the points corresponding to the vertices on each connected subgraph need to be divided into independent clusters $\mathtt{C}_\mathtt{i}$, i = 1,2,...,n, where n is the number of clusters, $= \mathcal{P}$, $\mathtt{C}_\mathtt{i} \bigcap \mathtt{C}_\mathtt{j} = \Phi$, $= 1, 2, \ldots, n, \mathtt{i} \neq \mathtt{j}$. We construct a new complete undirected graph $G_{\mathtt{C}l}$, where $\mathcal{W}_{v_\mathtt{i} v_\mathtt{j}} = min\left\{ x \middle| x = \|p - q\|_2, p \in \mathtt{C}_\mathtt{i}, q \in \mathtt{C}_j \right\}$, $\forall_{v_\mathtt{i} v_\mathtt{j}} \in G_{\mathtt{C}l}.V, v_\mathtt{i} \neq v_\mathtt{j}$, whose edge weights are assigned as the minimum inter-cluster distance between every pair of clusters. Let W_{max} be the weight of the longest edge within the minimum spanning tree (MST) of $G_{\mathtt{C}l}$, this value corresponds to the minimum search radius required to make G connected. For all, p, q $\in \mathcal{P}$, if they are in different clusters and $\le$, connect the corresponding edge in G. At this time, G becomes a connected graph. Here, W_{max} is the minimum search radius that makes the disconnected graph G connected; a pair of clusters is reconnected only when their minimum inter-cluster distance does not exceed W_{max}, which prevents bridging distant branches.

Step 4: Let the vertex corresponding to one point with the lowest z-value be v_{src}. Find the shortest paths from v_{src} to all the other vertices in G using Dijkstra's algorithm. The result is an SPT. For convenience, the symbol dis ($\mathbf{p}$) is used to represent the shortest path distance between the vertex corresponding to point $\mathbf{p}$ and vertex v_{src}.

2.3. Skeleton Extraction

This research first significantly improves the skeleton extraction method of Yang's method [29] by separating the procedure of segmenting tree branch point clouds and completely replacing recursion with iteration. The improved method is contextual and has low computational complexity. It contains two algorithms.

Algorithm 1 shows some details about segmentation. In particular, the inequality in Line 5 can be used to solve the issue where segments with no points cause program crashes in the previous algorithm. In addition, Line 12 provides a potential solution to deal with center deviation at the stump, which is a deficiency existing in distance-based segmentation [29], by combining several segments close to the ground.

Algorithm 1: Segmentation

Input: P, dis($\cdot$), step length L, number of layers that need to be merged as the part of stump n_{stump}
Output: Sequence of sets $\{\mathcal{A}_\mathrm{n}\}$, which is a segmentation of **P**
1 Get a sequence $\{\mathbf{p}_\mathrm{n}\}$ of **P** to satisfy Dis($\mathbf{p}_\mathrm{i}$) $\le$ dis($\mathbf{p}_\mathrm{j}$), = 1,2, ..., $|\boldsymbol{P}| \wedge i < j$;
2 Initialize $\{\mathcal{A}_\mathrm{n}\}$: $\mathcal{A}_1$, where $\mathcal{A}_1$: = {**p**1} and $|\{\mathcal{A}_\mathrm{n}\}|$ is its current number of terms;
3 Define traveled distance d: = dis($\mathbf{p}_1$), where dis($\mathbf{p}_1$) = 0;
4 **for** $i = 2 \rightarrow |\mathbf{P}|$ do
5 **if** dis($\mathbf{p}_\mathrm{i}$) $> d + 1$ **then**
6 $\mathcal{A}_{|\{\mathcal{A}\mathrm{n}\}|+1}$: = {$\mathbf{p}_\mathrm{i}$};
7 $d \leftarrow$ dis{$\mathbf{p}_\mathrm{i}$};
8 **else**
9 $\mathcal{A}_{|\{\mathcal{A}\mathrm{n}\}|} \leftarrow \mathcal{A}_{|\{\mathcal{A}\mathrm{n}\}|} \cup \{\mathbf{p}_\mathrm{i}\}$;
10 **end**
11 **end**
12 $\mathcal{A}_1 \leftarrow \mathcal{A}_1 \cup \mathcal{A}_2 \cup \ldots \cup \mathcal{A}_{n\mathrm{stump}}$, and remove the 2-nd, 3-rd, ..., n_{stump}-th terms from $\{\mathcal{A}\mathrm{n}\}$;

Algorithm 2 shows the novel iterative implementation. This algorithm requires the design of four parts: the input sequence of clusters, the function $c(\cdot)$ to compute the coordinates of a skeleton point, the clustering algorithm (and its parameters), and the function $f(\cdot)$ for determining which previous vertex should be connected to. In Line 1, the result of $p(v)$ is a point in the form of a three-dimensional column vector, and $c(\cdot)$ used in this research is the centroid of a point set. Lines 5, 9, 10 and 11 reveal that identifying the parts of different branches requires clustering with $\mathcal{A}i \cup \mathcal{A}i + 1$, but producing a skeleton point only requires points in $\mathcal{A}i$, and which skeleton point from the current round the next round's skeleton point should connect to depends on the overlap rate calculated using points in $\mathcal{A}i + 1$. Slices are the basis for determining the coordinates of skeleton points. For clarity, the main symbols used in Algorithm 2 are defined as follows: C denotes the set of clusters {C1, ..., Cm} obtained in a given iteration; $\mathcal{F}(\cdot)$ (written as $f(\cdot)$ in the main text) is the successor-vertex localization function that decides which previous vertex a new skeleton point connects to; $bin(\cdot)$ maps a vertex to the set of slices that belong to it. Throughout the paper $f(\cdot)$ and $\mathcal{F}(\cdot)$ refer to the same function and are used interchangeably. The iterative replacement of recursion makes the skeletonization stage approximately linear in the number of slices and clusters, with no recursion stack; replacing shallow recursion with iteration removes the risk of stack overflow for deeply nested branches, and the estimation error propagates in a bounded manner across iterations, in contrast to the shallow recursion of the original Yang method.

Algorithm 2: Skeletonization

Input: $\{\mathcal{A}n\}$
Output: Skeleton (in the form of directed tree) T, functions to get the point and point cloud slicecorresponding to the vertex in the skeleton $p(\cdot)$, $bin(\cdot)$

1 Define function $f(\cdot)$ to obtain the corresponding previous vertex of a point set, function $c(\cdot)$ to compute the skeleton point according to the input point set, and initialize $p(\cdot)$, $bin(\cdot)$;
2 Initialize T, where $bin(v_1) := \mathcal{A}_1$, $\mathcal{F}(\mathcal{A}_2) :=$, $T.\mathcal{V} := \{v_1\}$, and $p(v_1) := c(\mathcal{A}_1)$;
3 Define set C′ to storage the parts of clusters that were not used for calculating skeleton points in the previous round. Initially, $C' := \{\mathcal{A}_2\}$ as $\mathcal{A}_1$ is assumed to be only able to cluster into one class and consequently $\mathcal{A}_1 \cup \mathcal{A}_2$ can only be clustered into one class;
4 **for** $i = 2 \rightarrow |\{\mathcal{A}n\}|$ **do**
5 Cluster $\mathcal{A}_i \cup \mathcal{A}_{i+1}$ into $C_1, C_2, ..., C_m$, where m is the number of clusters, and $\mathcal{A}_{|\{\mathcal{A}n\}|+1} = \emptyset$;
6 $C'' := \emptyset$;
7 **for** $j = 1 \rightarrow m$ **do**
8 Insert a new vertex v into T, where $bin(v) := C_j \cap \mathcal{A}_i$, $p(v) := c(C_j \cap \mathcal{A}_i)$;
9 $\mathcal{F}(C_j \setminus \mathcal{A}_i) := v$, $C'' \leftarrow C'' \cup \{C_j \setminus \mathcal{A}_i\}$;
10 $\mathcal{D}^* := \arg\max_{\mathcal{D} \in C'} |(C_j \cap \mathcal{A}_i) \cap \mathcal{D}|$
11 $v' := \mathcal{F}(\mathcal{D}^*)$;
12 $T.\mathcal{E} \leftarrow T.\mathcal{E} \cup \{(v', v)\}$;
13 **end**
14 $C' \leftarrow C''$
15 **end**

The clustering algorithm chosen in this research is Euclidean clustering. It has two necessary parameters—eps and MinPts. The idea is to divide points with a distance less than the given threshold eps into the same cluster, and remove clusters with fewer points than MinPts as noise. As the input branch point cloud is assumed to exist only points of branches, here MinPts is set to 1. However, for different segments with different densities, a fixed eps is not suitable. To cope with such differences, the LOF algorithm [30], an unsupervised outlier

detection algorithm, is introduced to dynamically determine eps. It measures the local density deviation of a given data point with respect to the data points near it, which can be used to reduce the influence from close parts on different branches. This choice (MinPts = 1) is reasonable because the input branch point cloud is assumed to contain only branch points with no background noise, so a singleton cluster is already a valid branch segment rather than an outlier. The neighbor number k of LOF is selected from the local-density statistics of φ (we adopt a robust quantile of the k-distance distribution) and is kept distinct from the graph-construction neighbor number k in Step 1; to avoid confusion we denote them as k_LOF and k_graph, respectively. In practice LOF first applies the local-density outlier criterion (bounded by the dmax ($\mathcal{D}$) constraint) to reject cross-branch near-neighbor noise, and then the density-adaptive Euclidean clustering partitions different branches; that is, LOF delimits while Euclidean clustering determines the clusters, which is how adjacent or overlapping branches are separated.

Here is a brief introduction to the principle of the LOF algorithm. For the two points $\mathbf{p}$ and $\mathbf{q}$,

$$\text{reachability_}d\text{istance}_k(\mathbf{p},\mathbf{q}) \stackrel{\text{def}}{=} \max\{k\text{_distance}(\mathbf{q}), \|\mathbf{p}-\mathbf{q}\|_2\}, \tag{1}$$

where k_distance ($\mathbf{q}$) is the distance between q and its k-th nearest neighbor. Then, reachability distances to all of the k-nearest neighbors of point p are calculated to determine the Local Reachability Density (LRD):

$$\text{lrd}_k(\mathbf{p}) \stackrel{\text{def}}{=} \frac{|N_k(\mathbf{p})|}{\sum\limits_{\mathbf{q}\in N_k(\mathbf{p})} \text{reachability_distance}_k(\mathbf{p},\mathbf{q})}. \tag{2}$$

Next,

$$\text{LOF}_k(\mathbf{p}) \stackrel{\text{def}}{=} \frac{\sum\limits_{\mathbf{q}\in N_k(\mathbf{p})} \text{lrd}_k(\mathbf{q})}{|N_k(\mathbf{p})|\text{lrd}_k(\mathbf{p})}. \tag{3}$$

Finally, when the value is approximately 1, p has a density distribution that is similar to its neighbors. And, the smaller the value, the denser the distribution; conversely, the sparser the distribution. Calculate the LOF value for each point in the point cloud in this way and consider the points with LOF below the specified threshold as outliers. According to this principle, two parameters, the number of neighbors k and the allowed LOF lower limit LOF_{min}, are needed and affect the output.

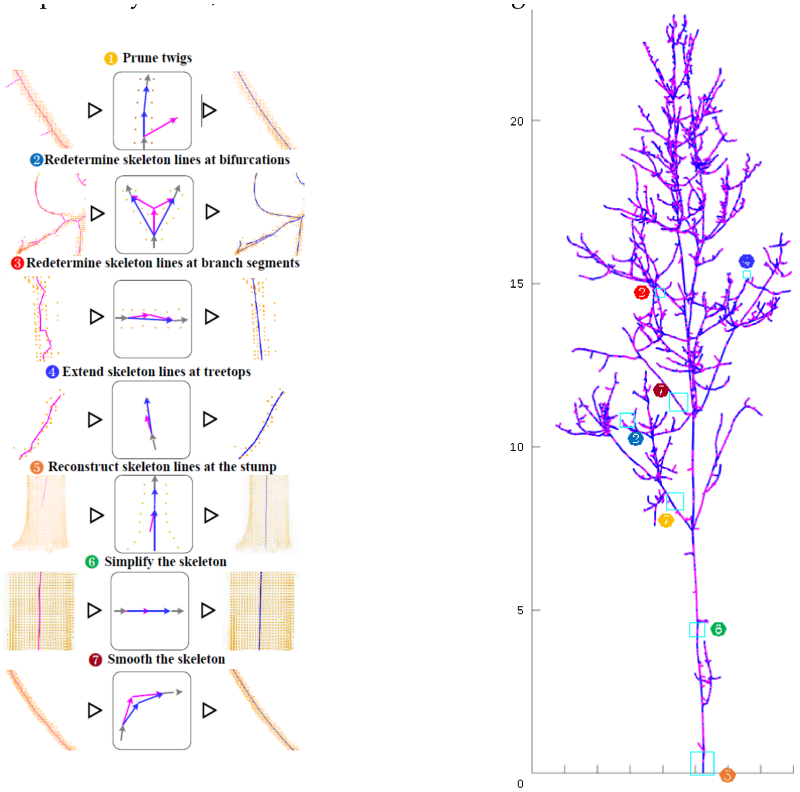
Remove points with LOF exceeding the threshold from the input point cloud, and let the remaining point cloud be D. Using LOF, in Euclidean clustering

$$\text{eps} = \min\{d_{\max}(\mathcal{D}), w_{\max}\}. \tag{4}$$

Adding such restrictions can effectively prevent the occurrence of overly large d_{max} ($\mathcal{D}$) due to sparse points at the canopy. In this way the clustering scale follows the local density instead of being tuned per cloud, which is what allows branches to be separated on clouds whose density varies within a single tree and on clouds whose surface is only partially present.

2.4. Skeleton Refinement

This study adopted seven stages of skeleton refinement to improve the accuracy and naturalness of the initial skeleton. Figure 2 reveals the work required for skeleton refinement and the differences in visual effects before and after. Bifurcations and branch segments are where cross-sectional distortion originates, because taper and diameter are estimated precisely there; the seven refinement stages below are aimed at these locations.



(**a**) Seven stages of skeleton refinement (**b**) Overlapping effect (without point cloud)

Figure 2. Schematic diagram of skeleton refinement. The initial skeleton is a series of purple lines, and the refined skeleton is a series of blue lines. Each algorithm scheme is an abstract schematic diagram for performing operations. In order to better display the effect, viewpoints in some screenshots have been changed.

2.4.1. Pruning Twigs

This stage is designed for eliminating some treetop skeleton points with small slices and corresponding lines. These slices contain noise, but it cannot be effectively removed through denoising during preprocessing.

In this research, two parameters MinPts and difference rate η_{diff} are used to decide which treetop skeleton points should be removed. If there are fewer than MinPts points in the slice and the points are fewer than η_{diff} times the maximum point number in slices of siblings, this treetop skeleton point should be removed. This strategy of adding the difference rate can effectively protect the twigs formed by sparse points at the canopy from being removed.

2.4.2. Redetermining Skeleton Lines at Bifurcations

According to the skeletonization, the connection of skeleton lines occurs in two adjacent segments. While different subbranches have different directions at a bifurcation, the first skeleton point on a subbranch might not connect to its predecessor skeleton point, but rather to one ancestor skeleton point, to make the skeleton line pass through the local branch point cloud. This stage is designed to find more reasonable connections.

This is a technical difficulty. Some studies adopted an interpolation method, such as the most commonly used Hermite interpolation [17], but such a method detaches from the point cloud. It keeps the original connection and just adds some sampling points to make the skeleton look smooth. With regard to the local point cloud, it is hard to optimize all bifurcations to achieve a good result for the point cloud scanned [21].

Before a solution is given, it is necessary to first introduce axis-constrained least-square cylinder fitting. Least squares fitting is the solution of an optimization problem:

$$\hat{\boldsymbol{\beta}}^{*} = \operatorname{argmin}_{\hat{\boldsymbol{\beta}}} \sum_{i=1}^{n} e_i^2, \tag{5}$$

where the i-th residual $e_i = y_i - \hat{y}_i$ and the estimated value of y,$\hat{y}_i$, is the solution to independent variable vector x with parameter vector $\hat{\beta}$. Axis-constrained least-square cylinder fitting means that the direction of the cylinder is given, and the projection of data points on the axis decides the start point and end point, so only the radius needs fitting. Let a slice with the shape distribution of points close to a cylinder be $\mathcal{B}$ and the start and end points of the cylinder axis be $\mathcal{M}$ (x_1, y_1, z_1) and $\mathcal{N}$ (x_2, y_2, z_2). Rotate each point $\mathbf{p} \in \mathcal{B}$ using

$$\mathbf{p}' = \begin{bmatrix} \cos\theta_y & 0 & -\sin\theta_y \\ 0 & 1 & 0 \\ \sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} \mathbf{p}_1 - \frac{x_1+x_2}{2} \\ \mathbf{p}_2 - \frac{y_1+y_2}{2} \\ \mathbf{p}_3 - \frac{z_1+z_2}{2} \end{bmatrix}, \tag{6}$$

where

$$\theta_x = \arccos \frac{z_2 - z_1}{\sqrt{(y_2 - y_1)^2 + (z_2 - z_1)^2}}, \tag{7}$$

$$\theta_y = \arccos \sqrt{\frac{(y_2 - y_1)^2 + (z_2 - z_1^2)}{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}, \tag{8}$$

and let the rotated set be $\mathcal{B}'$. Therefore, the fitting radius $\hat{r}$ can be calculated by

$$\hat{r} = \sqrt{\frac{1}{|\mathcal{B}'|} \sum_{\mathbf{p} \in \mathcal{B}'} (\mathbf{p}_1^2 + \mathbf{p}_2^2)}. \tag{9}$$

and let the rotated set be $\mathcal{B}'$. Therefore, the fitting radius $\hat{r}$ can be calculated by

$$\mathrm{SSe} = \sum_{\mathbf{p} \in \mathcal{B}} (d(\mathbf{p}, MN) - \hat{r})^2, \tag{10}$$

where d $(\mathbf{p}, \mathrm{MN})$ is the distance from point $\mathbf{p}$ to line MN.

Based on the above, the connection method at bifurcations can be determined by the best degree of cylindrical fitting on the slice. For any one of the vertices $v \in T.\mathcal{V}$, if its outdegree $\deg^+(v) \geq 2$, it is the vertex corresponding to a bifurcation point. Let its i-th successor vertex be $v_{\mathrm{succ_i}}$ and its former j-th generation ancestor vertex be $v_{\mathrm{prevgen_j}}$, where $v_{\mathrm{prevgen_0}} := v$. The maximum limit for j, j_{max}, should be set in case of overfitting. The way to redetermine skeleton lines at a bifurcation can be formulated as an optimization problem: for each successor vertex,

$$j^* = \operatorname*{argmin}_{j=0,1,\ldots,j_m} \sum_{\mathbf{p} \in \mathrm{bin}(v_{\mathrm{prevgen_}j})} (d(\mathbf{p}, l(p(v_{\mathrm{succ_}i}), p(v_{\mathrm{prevgen_}j}))) - \hat{r}_i)^2, \tag{11}$$

where l($\mathbf{p}$, $\mathbf{q}$) is the spatial line formed by $\mathbf{p}$ and $\mathbf{q}$, and $v_{\mathrm{prevgen_j}*}$ is a more suitable predecessor vertex of $v_{\mathrm{succ_i}}$. Add edge $(v_{\mathrm{prevgen_j}*}, v_{\mathrm{succ_i}})$ and remove edge $(v, v_{\mathrm{succ_i}})$ when $v_{\mathrm{prevgen_j}*}, \neq v$. If all successor vertices at a bifurcation have already been processed, remove $(v_{\mathrm{prevgen_j+1}}, v_{\mathrm{prevgen_j}})$ until not satisfying $\mathrm{j}+1 \neq \mathrm{j}* \vee \mathrm{j} \neq \mathrm{j}*$, where j increases from 0. This step eliminates excess lines that are no longer connected to any subbranches. At this moment, bin $(v_{\mathrm{prevgen_j}*}) = \bigcup_{j=0}^{j^*} bin(v\mathrm{prevgen_j})$.

2.4.3. Redetermining Skeleton Lines at Branch Segments

Sometimes, the middle one of three consecutive skeleton points is unnecessary. Instead, its existence exacerbates distortion. This stage is designed to detect points that match the feature and remove them.

Suppose $\hat{r}$ is a function of B and two points corresponding to two vertices v_i and v_j in the skeleton, according to Formula (10), we use SSe $(\mathcal{B}, v_i, v_j)$ to represent the SSe determined by the three variables. Let the three consecutive vertices be v_1, v_2 and v_3. Now, the procedure can be described as comparing:

$$\begin{aligned} \mathrm{SSe}_{\mathrm{orig}} &= \mathrm{SSe}(\mathrm{bin}(v_1), v_1, v_2) + \mathrm{SSe}(\mathrm{bin}(v_3), v_2, v_3) \\ &+ \sum_{\mathbf{p} \in \mathrm{bin}(v_2)} \min \begin{Bmatrix} (d(\mathbf{p}, l(p(v_1), p(v_2))) - \hat{r}(v_1, v_2))^2, \\ (d(\mathbf{p}, l(p(v_2), p(v_3))) - \hat{r}(v_2, v_3))^2 \end{Bmatrix}, \end{aligned} \tag{12}$$

and

$$\mathrm{SSe}_{\mathrm{test}} = \mathrm{SSe}(\mathrm{bin}(v_1) \cup \mathrm{bin}(v_2) \cup \mathrm{bin}(v_3), v_1, v_3). \tag{13}$$

(v_1, v_3) replaces original (v_1, v_2) and (v_2, v_3) when SSeorig $\geq$ SSetest.

2.4.4. Extending Skeleton Lines at Treetops

As the skeleton points are the centroids of corresponding slices, the skeleton points at treetops and at the stump are usually not at the actual locations. For a treetop, use the farthest point relative to the skeleton line through its slice to correct the position of the skeleton point.

2.4.5. Reconstructing Skeleton Lines at the Stump

Although multiple slices were merged to prevent excessive proximity to one side of the surface, they are still not in the center. The strategy adopted in this study to solve this problem is to cut a slice that approximates a cylinder at the tree stump, and use the projection of the centroid of this slice on the lowest horizontal plane as the origin point of the skeleton.

2.4.6. Simplifying the Skeleton

In other studies such as Du et al. [22], skeleton simplification focuses more on reducing the edges in skeleton T. In addition, it is also related to studying which edges can be "collapsed" into an edge that can be represented more simply. In this study, skeleton simplification specifically refers to a procedure of merging adjacent skeleton lines with identical directions into one line.

2.4.7. Smoothing the Skeleton

In order to reduce distortion and make the skeleton visually smooth, smoothness is required. In this research, the expression proposed by Xu et al. [31] is modified because the differences in lengths of different skeleton lines were not taken into account. The improved formula is

$$\begin{aligned} \mathbf{p}_{(2);k+1} = \mathbf{p}_{(2);k} &+ \frac{\lambda \|\mathbf{p}_{(1);k} - \mathbf{p}_{(2);k}\|_2 (\mathbf{p}_{(1);k} - \mathbf{p}_{(2);k})}{\|\mathbf{p}_{(1);k} - \mathbf{p}_{(2);k}\|_2 + \|\mathbf{p}_{(3);k} - \mathbf{p}_{(2);k}\|_2} \\ &+ \frac{\lambda \|\mathbf{p}_{(3);k} - \mathbf{p}_{(2);k}\|_2 (\mathbf{p}_{(3);k} - \mathbf{p}_{(2);k})}{\|\mathbf{p}_{(1);k} - \mathbf{p}_{(2);k}\|_2 + \|\mathbf{p}_{(3);k} - \mathbf{p}_{(2);k}\|_2}, \end{aligned} \tag{14}$$

which uses length weighting. In Formula (14), k is the number of the iterations, which should be less than the incoming parameters kmax (normally 5 or 6); λ is a coefficient to balance the smoothness result (set to 0.1 by default); $\mathbf{P}_{(1)}$, $\mathbf{P}_{(2)}$ and $\mathbf{P}_{(3)}$ are three consecutive points and only one of the first and last points is allowed to be a bifurcation point.

2.5. Radius Estimation

After a skeleton is produced, only the radius of the branch cross-section corresponding to each vertex or edge is needed to produce the model. In this research, we propose a self-adaptive WBE model which dynamically determines the reasonable fitting radii and then combines the corresponding Support Lengths (SLs) of branches to achieve self-adaptive parameter selection of the WBE model. It is assumed that the scanned individual trees have not been artificially trimmed or naturally broken.

First, fit circles. With the help of Formula (6), where θ_x and θ_y are calculated by the two vertices of an edge (u, v), rotate the slice formed by points in $\mathrm{bin}(u) \cup \mathrm{bin}(v)$ and between two planes with $\mathrm{p}(v)-\mathrm{p}(u)$ as the normal vector and passing through $\mathrm{p}(u)$, $\mathrm{p}(v)$. Project the slice and fit the least squares circle. The radii of the fitting circles are required. For the convenience of later expression, let the radius of edge (u, v) be $\mathrm{R}(u, v)$. There must be at least 10 points in the projected slice, or the corresponding $\mathrm{R}(u, v)$ should be regarded as noise.

Then, delete data points with low SL values. Generally speaking, the radius fitting closer to the ground is more accurate, which means that the data points with larger supporting length need to be retained.

Next, FAST-LTS [32], a robust regression algorithm, is introduced to detect which radii are reasonable and to fit the self-adaptive WBE model. In this research, the WBE model can be expressed as

$$R(u, v) = a(\mathrm{SL}(v) + \|p(v) - p(u)\|_2)^b, \tag{15}$$

where a and b are fitting parameters. The radius of each pipe is usually measured using the bottom radius. When computing the initial fitting coefficients, FAST-LTS uses the number of points equal to the dimension of the points (i.e., 3), while all points are used here to avoid numerical error. The number of iteration rounds is determined by the minimum value of m in a probability problem

$$\left(1 - \left(\frac{1}{n}\left\lfloor \frac{n+3+1}{2} \right\rfloor\right)^3\right)^m \leq 0.001. \tag{16}$$

Finally, use the self-adaptive WBE model to estimate all radii again. Of course, this is for visual esthetics. To further improve the realism, some reasonable radii calculated by the former fitting can be retained by adding a judgment of relative error. For dimensional consistency, both SL(v) (support length) and the radius R are lengths, so in the WBE form R = a·SL + b the coefficient a is dimensionless and b carries length units, keeping the equation dimensionally homogeneous; this follows the standard allometric form R ∝ L^(1/3) of West et al. [33]. Here SL(v) is defined as the distance along the skeleton from vertex v to the trunk base (or to the nearest trunk-path point). The FAST-LTS parameters are consolidated as follows: the subset size is taken as the point dimension (3) because the slice-radius fitting is a low-dimensional problem, and all points are used to avoid numerical error; the number of iteration rounds is determined by the minimum m that satisfies the coverage probability in Equation (16), with the coverage probability set to an empirical value.

2.6. Calculation of Branch Orders

For tree modeling, the differentiation of branch orders is generally based on monopodial branching, while concrete branching types are no longer taken into account. It is generally believed that all non-thickest branches at a bifurcation are secondary branches. Calculate the order of various branches in this way.

2.7. Modeling

Arterial Snake [34] is commonly used for expressing the surface of a branch, i.e., the branch mesh. It uses the Rotation Minimizing Frame (RMF) to propagate the local coordinate frame from one cross-section to the other cross-sections to adapt to changes in the direction of the skeleton line [35]. Let the RMF be a right-handed orthonormal system $\mathrm{U}(u) = (\mathrm{T}(u), \mathrm{r}(u), \mathrm{s}(u))$, where $\mathrm{x}(u)$ is a spine curve, $\mathrm{T}(u) = \mathrm{x}'(u)||\mathrm{x}'(u)||2$, reference vector $\mathrm{r}(u)$ is a solution to a system of Differential–Algebraic Equations (DAE),

$$\begin{cases} \mathbf{r}'(u) - h(u)\mathbf{T}(u) = 0 \\ \mathbf{r}(u)\mathbf{T}(u) = 0 \end{cases}, \tag{17}$$

$\mathrm{h}(u)$ is a certain function, and $\mathrm{s}(u) = \mathrm{T}(u) \times \mathrm{r}(u)$. The starting vertex of the regular polygon in the cross-section of the branch is in the reference direction $\mathrm{r}(u)$ of the skeleton point, and the other vertices are generated in counterclockwise order within the cross-section.

Sometimes, two adjacent cross-sectional polygons intersect and the intersecting parts are not the vertices of either polygon. These two faces obviously cannot be used as two cross-sections of the pipe. In this research, a convex polyhedron was constructed using the vertices of two adjacent cross-sectional polygons on each branch, and a closed branch model was formed using the Boolean operation of the polyhedron.

2.8. Experimental Design and Evaluation Protocol

2.8.1. Purpose

The evaluation in this study is designed to assess the accuracy and robustness of the proposed reconstruction pipeline under field-realistic point-cloud defects, not to characterize tree species. The nine point clouds therefore serve as test cases spanning the conditions a QSM must face, rather than as a species-level sample.

2.8.2. Composition of the Test Cases

The test set comprises nine individual trees covering one conifer (Masson pine) and five broad-leaved species (Tung, Beech, Ash, Oak, and four Populus deltoides clones), captured with four TLS instruments together with two simulated geometries produced by HELIOS++. The number of points ranges from 22,290 to 819,556—nearly a forty-fold span—and the scans include typical field defects such as missing back surfaces, wood–leaf mixing, and canopy ghosting. Each tree was selected to represent a condition (sparse, dense, missing, or noisy) that makes field density non-uniform, not to represent a species.

2.8.3. Unit of Observation and Statistical Reporting

The unit of observation is the reconstructed model and its derived metrics (e.g., DBH, branch order, breast-height center), not the individual tree as a biological replicate. Each tree is an independent test case; we report descriptive statistics (mean, standard deviation, n) over these heterogeneous cases and make no species-level inference or significance test across trees. Plot-level validation in Section 3.4 uses independently measured DBH of multiple individuals within a poplar plot and is reported separately, so it should not be conflated with the nine-case evaluation set.

2.8.4. Computational Environment

The experiment was conducted on a Windows 11 computer with an i7-1165G7 CPU and 16 GB memory. The program is written in C++ and utilizes the PCL point cloud library. Preprocessing followed the standard pipeline described in Section 2.1 (multi-station registration, denoising, ground normalization, tree segmentation, and wood–leaf separation) without additional downsampling. The key parameters were set as follows:

dmax as a multiple of the mean nearest-neighbor distance of φ; k_graph ≥ 3; k_LOF chosen from the k-distance quantile of the local density; MinPts = 1; Wmax as the MST minimum-connectivity radius; and FAST-LTS with subset size 3, all points used, and iteration rounds determined by the coverage probability in Equation (16). It was compiled with MSVC 2019. Regarding computational cost, the pipeline is O(N log N) in the point count N (Section 2.3); the additional LOF screening and seven-stage iterative refinement add a bounded overhead over the O(N) graph-based Yang method, consistent with the statement in Limitations that segmentation-based skeletonization is more time-consuming than AdQSM. For the case sizes studied (0.02–0.82 million points, Section 2.8.2), this additional cost remains acceptable for off-line, non-real-time phenotypic characterization. Because dmax, k_graph, k_LOF and Wmax are all derived from local point statistics, the same parameter settings generalize across the nine heterogeneous cases without per-cloud tuning, as reflected by the consistent results in Section 3.

3. Experiments and Results

3.1. Visual Evaluation

Figure 3 shows these point clouds. The visual effects of each experimental material produced by three methods are shown in Figure 4, where different colors represent different levels of branches. Figure 5 shows the degree of error between point clouds and models, measured by the distance between each point in the point cloud and the model. Generally speaking, the bluer the overall/local color of the rendered point cloud, the smaller the overall/local error level. Figure 6 shows the display effect of overlapping the point cloud and model, which measures the coverage of the model over the point cloud. Figure 6 needs to be analyzed together with Figure 5. When the overall error is small, the more black points in the graph, the thinner the model is compared to the point cloud; when there are fewer black dots, it can only indicate that the model is fatter than the point cloud, but if the overall error is smaller, the model is more in line with the point cloud.

From Figure 4, the Oak model reconstructed by AdQSM has excessively thick branches; the trunk of the Tung tree model is distorted and there is a non-existent branch; the branches constructed by AdQSM are too simplistic due to using a single pipe, which is evident in the Ash model; and in the Poplar (2), (3), and (4) models, the shorter branches near the ground were not reconstructed. The Masson pine and Tung tree models reconstructed by Yang's method show severe distortion, indicating that this method is not suitable for processing artificial point clouds. This method achieves good visual effects on other models. Compared to them, the models produced by our method have achieved good visual effects.

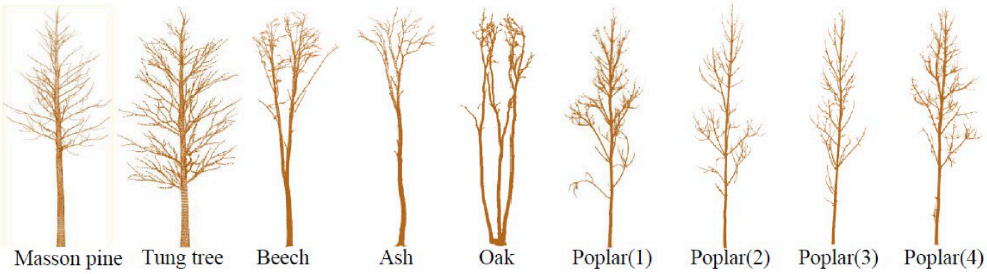


Figure 3. The experimental materials. Masson pine and Tung tree are artificial point clouds and the others are scanned by different TLS scanners.

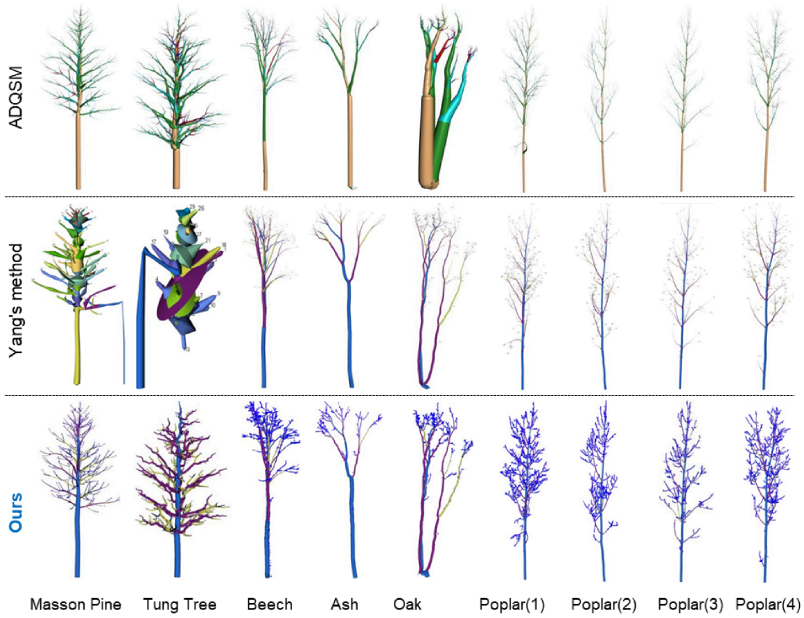


Figure 4. Produced models. Screenshots of three methods are used to show branch orders. In AdQSM, the main trunk is brown, the primary branch is green, and the secondary branch is sky blue; in Yang's method and our method, the main trunk is dark blue, the primary branch is purple, and the secondary branch is yellow.

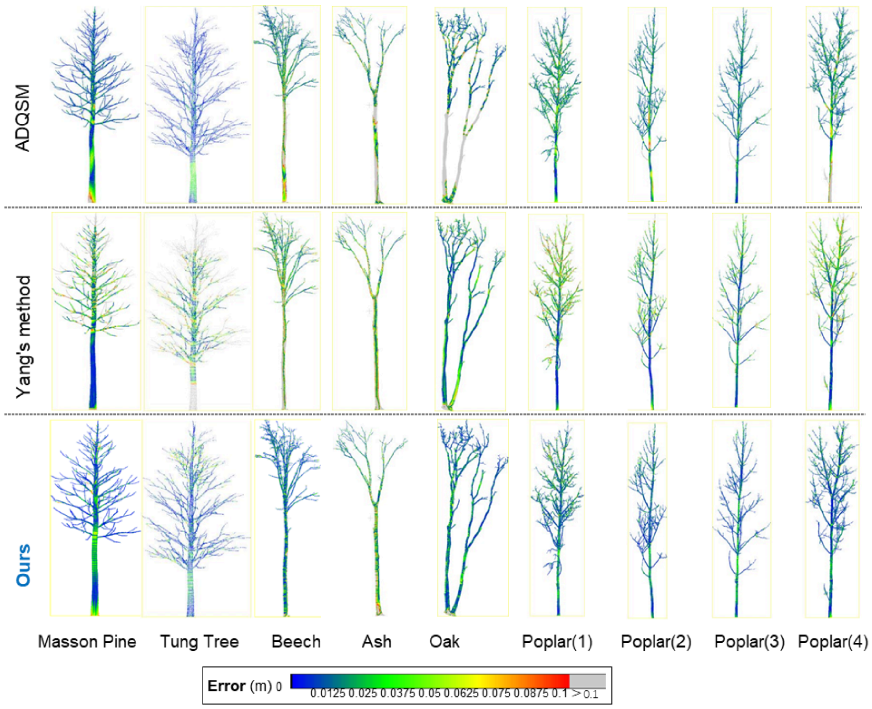


Figure 5. The degree of error between point clouds and models. The distance from each point to the surface of the model is used as a scalar to color all points.

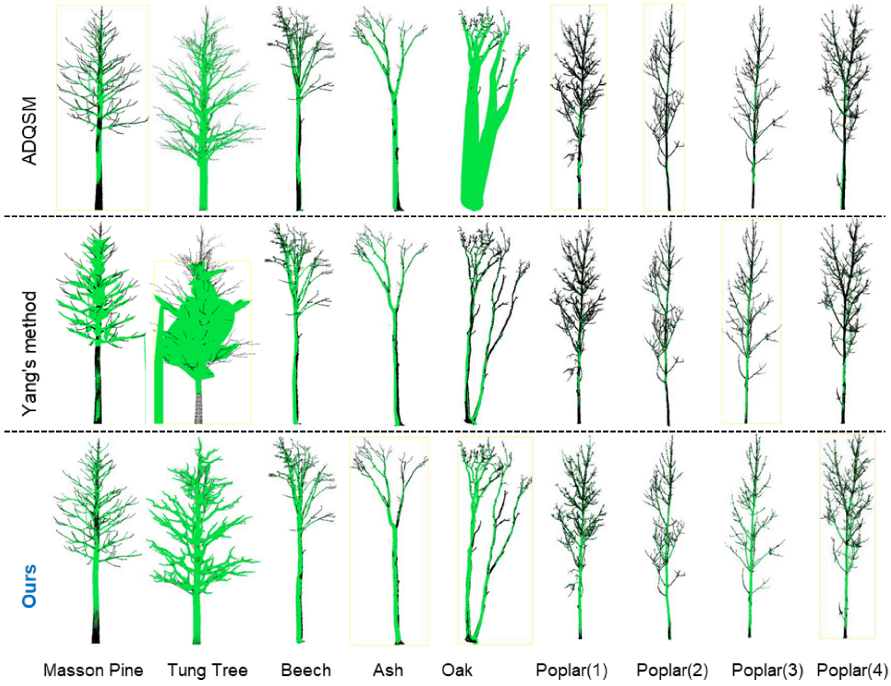


Figure 6. The degree of coverage between the model and the point cloud. The point cloud is black, and the model is green.

In Figure 5, the Beech, Ash, Oak and Poplar (4) models reconstructed by AdQSM have large-scale error exceeding 1 decimeter on the trunk; The Masson pine and Tung tree models reconstructed by Yang's method have a large-scale error exceeding 1 decimeter on the canopy; the models reconstructed by our method have an overall error of almost no more than 1 cm. Varying degrees of significant errors only appear in the stump of the Beech, Ash and Oak models, as well as those reconstructed by AdQSM and Yang's method. This can be explained as the failure of the WBE model at ground positions, as the WBE model is an equation with fixed convexity. Overall, our method produces a bluer error point cloud, which indirectly indicates that the model reconstructed by our method better matches the point cloud.

In Figure 6, we overlay the branch point cloud with the model and display it in Cloud-Compare. It can be observed that our method can reconstruct a model that includes more point clouds within itself. Of course, only analyzing the degree of coverage is meaningless. For example, although the Oak model reconstructed by AdQSM can cover the vast majority of point clouds, it does not fit well with the point clouds. Therefore, it is necessary to analyze Figure 6 together with Figure 5. Due to the smaller error of the model reconstructed by our method and the fact that the model can contain more points, the reconstructed model is not much coarser than the point cloud, which means it is closer to the surface of the point cloud. This is similar to the result of real-world plot surveys—the measured diameter at breast height and the calculated cross-sectional area are larger, but not much larger than the measured data, which can be acceptable.

3.2. Quantitative Analysis

In Figure 5, each distance is visualized as a point cloud display, which can be used to quantify the overall geometric accuracy of the model based on these distances. Mean error and standard deviation are the two most frequently used evaluation indices. Mean error (a.k.a accuracy) is defined as the mean distance between the produced model and the original point cloud, i.e., all reported distances are point-to-model (from each cloud point

to the nearest model surface), quantifying geometric fit and coverage; the complementary model-to-cloud distance and the resulting bidirectional Chamfer distance are deferred to future work.

$$\text{Acc} = \frac{1}{n}\sum_{i=1}^{n} \|\mathbf{p}_i^* - \mathbf{p}_i\|_2, \tag{18}$$

where P_i^* is the point closest to the model surface from P_i. The standard deviation here is defined as

$$\text{stddev} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (\|\mathbf{p}_i^* - \mathbf{p}_i\|_2 - \text{Acc})^2} \tag{19}$$

From the two indices of geometric accuracy in Table 2, it can be seen that our method achieves the minimum average error and variance simultaneously on most experimental materials. This indicates that the produced models not only fit better with the surface of the point clouds, but also have a relatively stable overall deviation distance difference. Compared to the other two methods, the error distribution has more uniformity. As shown in Figure 5, our method's error point cloud color distribution is more uniform than that of the other two methods, without visually obvious prominent areas. For the Tung tree, despite being numerically slightly less advantageous, the model reconstructed by our method does not have the non-existent binary, which means more accurate in fact. In Table 2, the Accs of our method are all below 1 decimeter, with a maximum of 6.43 cm and generally around 2 cm. This reveals that our method improves the accuracy of QSM on different tree species, which can further enhance the existing decimeter level to the centimeter level. In addition, the number of branches identified by our method falls between AdQSM and Yang's method, which will be further discussed in the robustness testing section. As the nine clouds differ in density and completeness, this stability indicates that the accuracy is retained when the density is non-uniform and part of the surface is missing, rather than only on clean scans.

Table 2. Quantitative analysis of models produced by three tree branch modeling methods. The produced model information is not comparable while Acc and stddev can evaluate the modeling effect in numerical form. Acc is the accuracy and stddev is the standard deviation. The best item has been bolded.

Tree Species	Skeleton Lines (Num.)			Identified Branches (Num.)			Acc ± stddev (cm)		
	AdQSM	Yang's Method	Ours	AdQSM	Yang's Method	Ours	AdQSM	Yang's Method	Ours
Masson pine	7213	268	1635	584	114	263	1.91 ± 2.03	6.85 ± 12.76	**1.79 ± 1.73**
Tung tree	7674	120	3596	866	55	323	**0.99 ± 1.22**	16.00 ± 22.03	1.49 ± 1.82
Beech	7235	978	7075	742	252	734	6.14 ± 5.12	11.99 ± 8.78	**2.51 ± 3.17**
Ash	4432	359	1035	307	67	108	13.28 ± 10.44	10.62 ± 9.58	**6.43 ± 5.77**
Oak	9590	1553	3866	757	257	706	19.58 ± 18.92	4.16 ± 5.16	**2.58 ± 3.37**
Poplar(1)	8777	892	5705	1456	282	618	2.97 ± 7.16	2.72 ± 3.89	**2.08 ± 1.62**
Poplar(2)	5094	611	3195	602	175	306	3.34 ± 8.77	1.97 ± 3.09	**1.69 ± 1.23**
Poplar(3)	4546	620	2897	487	159	285	2.75 ± 9.21	2.03 ± 2.77	**1.88 ± 1.36**
Poplar(4)	8144	828	5061	1148	268	552	4.33 ± 5.87	2.77 ± 4.26	**1.89 ± 1.52**

3.3. Robustness Testing

Robustness testing focuses on the details of the reconstructed model. For a tree branch point cloud, we mainly focus on details at the stump, breast height, the bifurcation and the upper part of the treetop. Due to the fact that the input branch point cloud in actual tree modeling comes from the scanning of the sample plot and must be as complete as possible, we chose Poplar (4) as the research object to further compare the robustness of the three methods. The result is shown in Figure 7.

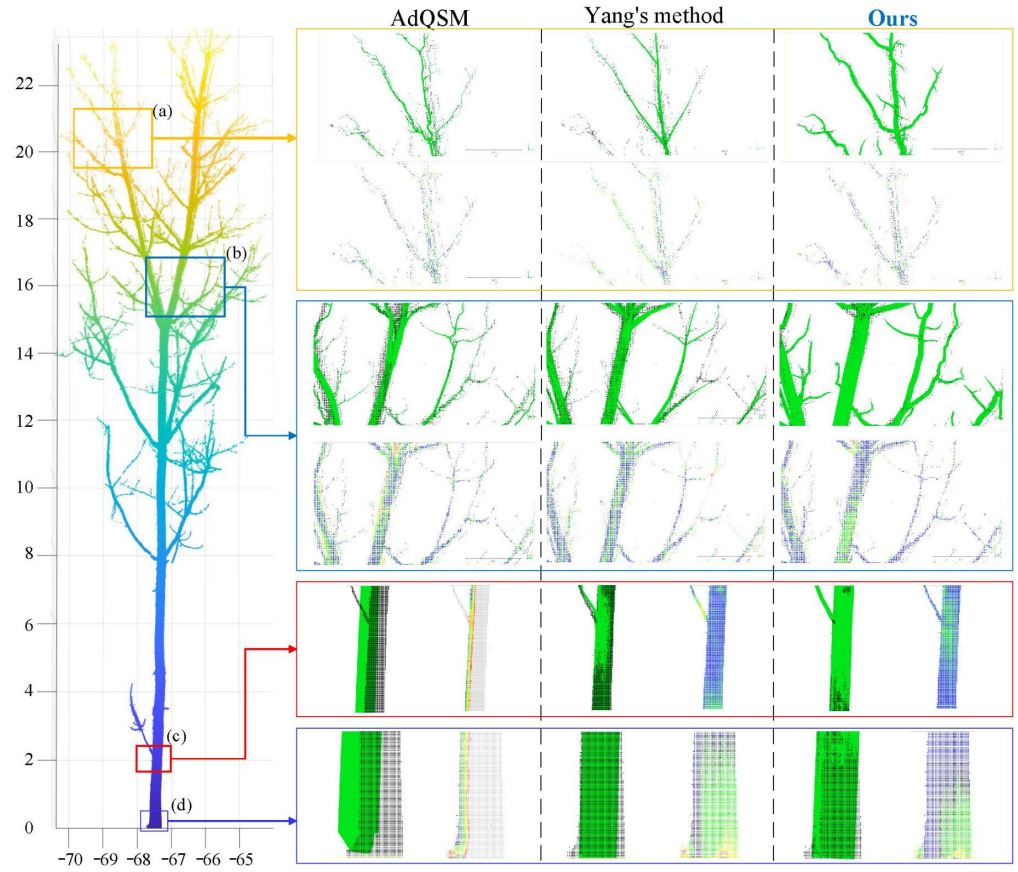


Figure 7. Robustness comparison of reconstruction details at four representative tree locations. Panels (**a**–**d**) correspond to the treetop apex, a branch bifurcation, breast height, and the stump, respectively. Close-up reconstructions using AdQSM, Yang's method, and the proposed method ("Ours") are arranged from left to right for each region.

In Figure 7, when AdQSM processed canopy ghosting, it mistakenly treated the ghosting points as new branches, resulting in many non-existent branches in the model. Yang's method, on the other hand, overlooks many small branches and has obvious topology errors. Our method overcomes the first two problems by using the LOF algorithm to detect noise during skeleton extraction and removing fine branches during skeleton refinement stage through comparison. At bifurcations, the three methods all exhibit the problem of individual branches not being accurately connected to varying degrees. Among them, AdQSM has the most severe branch position error at the bifurcation, followed by Yang's method, while our method has the smallest error. In addition, our method reconstructs branch segments that are both curved and smooth, without the excessive bending of AdQSM and the excessive smoothness of Yang's method, which is in line with the actual situation of trees in nature. This indicates that the skeleton reconstruction method proposed in this article for bifurcation and branch segments, as well as the improved skeleton smoothing, are effective. At breast height, the deviation in AdQSM is severe, while Yang's method and our method have a relatively similar effect. At the stump, the same is true, but our method has a smaller error, indicating that the reconstruction of the skeleton at the stump is effective. We can intuitively summarize this robustness analysis into five aspects, as shown in Table 3. The case of canopy ghosting is directly relevant: ghost points are noise, and treating them as branches—as single-criterion reconstruction does—is precisely the failure that noise produces in the field.

Table 3. Comparison of modeling robustness. Handling small branches, model topology, overall effect and handling adjacent branches belong to the evaluation of algorithm processing capability. Small center deviation belongs to the evaluation of algorithm performance. The number of symbol "+" indicates the goodness of fit.

Items	AdQSM	Yang's Method	Ours
Handling small branches	+		+
Model topology	+		++
Overall effect			+
Handling adjacent branches		+	++
Small center deviation		+	++

3.4. Overview of Sample Plot Reconstruction Effect

Figure 8 shows the branch model reconstructed by our method from partial point clouds of a poplar plot in Malanghu Forest Farm, as well as the models after adding texture and leaves. This plot is located in the same forest farm as the poplar trees shown above.

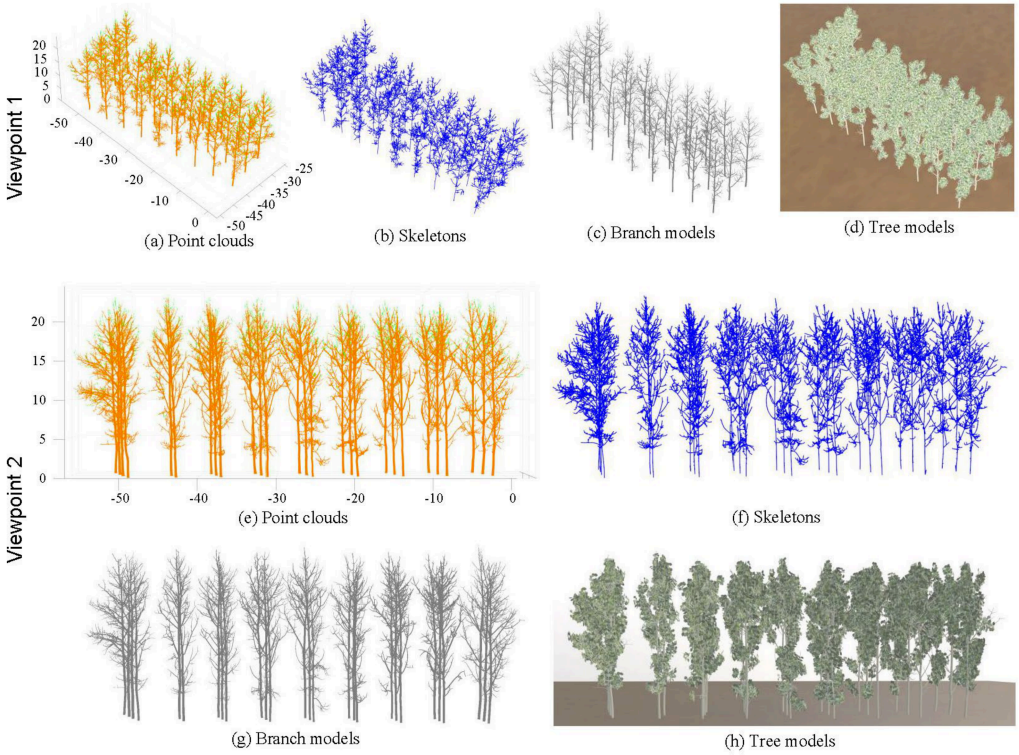


Figure 8. Overview of the sample plot and its reconstruction effect. (**a**,**e**) were displayed in MATLAB (R2022a); (**b**,**f**) were displayed in Point Cloud Magic (V2.0); (**c**,**g**) were displayed in Cloud Compare (2.12.4); (**d**,**h**) were displayed in Blender(3.6 LTS).

We analyzed the DBHs and centers at breast height in models produced by three different methods, and compared them with the actual DBHs and centers at breast height. The results are shown in Table 4.

It can be found that our method is more accurate in extracting DBHs in the selected sample plot, not only in terms of numerical accuracy, but also in achieving the minimum center deviation. The numerical deviation of DBH and the center deviation at breast height decrease respectively to 1.19 ± 0.74 (cm) and 3.97 ± 1.64 (cm), which is better than existing methods. This indicates that our method more accurately approximates the breast height cross-section. Since DBH and the breast-height center are both derived from the fitted cross-section, this deviation is a direct measure of how far the reconstructed cross-section

departs from the real one—that is, of the cross-sectional distortion under field defects. The plot-level comparison uses the independently measured DBH of multiple individuals within the plot, so its observation unit is the measured individual rather than the nine-case evaluation set.

Table 4. Numerical deviation of DBH and center deviation at breast height among the three methods. Two indices in the quantitative analysis are used to evaluate.

Items	AdQSM	Yang's Method	Ours
Numerical deviation of DBH: Acc ± stddev (cm)	4.30 ± 7.50	1.56 ± 1.21	**1.19 ± 0.74**
Center deviation at breast height: Acc ± stddev (cm)	6.96 ± 4.60	**3.97** ± 2.69	**3.97 ± 1.64**

4. Discussion

4.1. Construction of Skeletonization

The two methods we chose for comparison are relatively new open-source algorithms, both of which have certain applications, and they perform differently in handling different tree point clouds.

AdQSM is an attempt to bring points closer to the local center using spatial surrounding points after obtaining a skeleton on the surface. If the search scale of different regions cannot be reasonably determined, it will lead to the skeleton not being at the center of the branch. On the other hand, not all edges of each tetrahedron can serve as paths during Delaunay tetrahedralization. Finally, the principle of its construction path is relatively rigid, leading to some of the main branches being pruned. Then, clustering in Yang's method uses a fixed size and cannot adapt to different spatial density distributions of point clouds. The refined part of its skeleton is relatively mechanical, and there are not many aspects to be considered. The existence of these factors has caused some topology errors and removed too many details.

Our method can achieve better results than both methods on a complete point cloud because it overcomes the drawbacks of these two methods and makes the skeleton topology more accurate. Therefore, centimeter-level reconstruction was achieved on the test dataset. The comparison also shows where the two reference methods lose accuracy: both build edges or search neighbors under a single spatial criterion, so their errors concentrate where density is non-uniform—a condition field clouds rarely avoid.

4.2. Improvement on Radius Estimation

However, the radius estimation is a difficult task. The movement of small branches in the TLS process could easily cause the stratification of the registered point cloud data and, in turn, result in the overestimation of branch diameter (or radius) [10]. What is more, due to the missing data and limited resolution, it is difficult to place all skeleton points at the local centers, and the existence of deviation and noise significantly decreases the accuracy of estimated radii as well [22,35]. As a result, using a known radius to estimate the radii of other positions is a main method. It involves two questions:

How to determine the known radius. Normally, the point cloud at the trunk can be scanned completely, so ground diameter [22] or breast-height diameter [21] can be calculated more accurately. More frequently used fitting [22] or less frequently used point distance [36] and convex hull [37] methods can be applied to calculate the value of the radius.

How to estimate the other radius. Except methods like You et al. [19]**, allometric theory, specifically the WBE model** [38]**, is a** commonly adopted and highly persuasive

method in which radii at any other positions are computed based on the radius at one position. Sometimes, Murray's law [39], a formula that reveals the relationship between branch radii before and after branching, is adopted. Most previous studies have used default coefficients in formulae further organized by Xu et al. [17] without much thought. However, recent studies have found that these coefficients are not fixed. An increasing number of studies indicate that these coefficients are not always default values [34,40,41].

This research no longer uses only one radius for estimation. The self-adaptive WBE model using statistical learning methods preserves more robust values to obtain the required model parameters. In this way, when facing different tree species and even different samples of the same tree species, individual and differentiated models can be constructed. Therefore, when handling the point cloud of incomplete tree branches, it can better identify the main parts from the point cloud and construct a better overall model. This is also why inflated volume is suppressed: the radii that would be overestimated from sparse or noisy cross-sections are down-weighted by the robust regression and then re-estimated from the individual-tree allometry.

4.3. Practical Implications and the Degree of Problem Resolution

The method benefits: (i) tree breeding trait evaluation, through non-destructive, high-throughput acquisition of diameter at breast height (DBH), tree height, branch order, and 3D branching topology; (ii) the construction of functional-structural plant models (FSPM), by providing accurate 3D trunk-and-branch architecture as input; and (iii) precision afforestation and silviculture (target-tree management, pruning, light-environment assessment) via structure-based decisions. Regarding the degree of resolution: at the precision level, under nine heterogeneous point-cloud conditions including noise and missing-induced density non-uniformity, DBH deviation reaches 1.19 ± 0.74 cm and breast-height center 3.97 ± 1.64 cm, outperforming mainstream QSM [42], and cross-section distortion and inflated volume are mitigated through density-adaptive clustering and biologically mechanistic radius simulation; at the workflow level, automatic extraction without manual correction allows repeatable non-destructive re-measurement; at the scope level, self-adaptive WBE with FAST-LTS removes the need for species calibration [37]. We state honestly that the bottleneck of non-destructive, repeatable acquisition of individual-tree trunk-and-branch architecture is substantially eased under the conditions covered; full cross-species generality and extremely defective point clouds remain future work.

Four transferable lessons emerge from this study. (i) Density-adaptive graph construction and clustering constitute a prerequisite rather than an optional refinement for obtaining correct graphs and skeletons under non-uniform point density; the deficiencies of single-criterion methods are systematic, not incidental. (ii) LOF-based adaptive clustering eliminates the manual per-cloud tuning of clustering scale. (iii) Derived-parameter accuracy is governed by the treatment of bifurcations and branch segments rather than by global skeleton topology; local geometric fidelity dominates global shape. (iv) A fixed allometric coefficient imposes an implicit barrier to extending QSM to novel species; an individual-tree-adaptive radius model dispenses with species-specific calibration and is the element most apt to generalize beyond the present dataset.

The present approach advances non-destructive, repeatable acquisition of individual-tree trunk-and-branch architecture at the accuracy, workflow, and scope levels under the conditions examined. Full cross-species generality and the reconstruction of severely defective point clouds remain open problems.

5. Conclusions

Addressing the real-world and technical problem that the 3D architecture of the trunk and branch system of individual trees has long depended on destructive or non-repeatable acquisition and that traditional QSM tends to produce cross-section distortion and inflated volume [42] under point-cloud noise and missing data, the proposed high-precision reconstruction method reduces DBH and breast-height center deviations to the centimeter level under nine heterogeneous point-cloud conditions (1.19 ± 0.74 cm and 3.97 ± 1.64 cm); through density-adaptive clustering and biologically mechanistic self-adaptive WBE radius simulation, it achieves robust reconstruction under noise and missing data and enables automatic, non-destructive, repeatable, calibration-free [37] parameter extraction from LiDAR point clouds. Thereby, tree breeding trait evaluation, functional-structural growth simulation, and precision afforestation and silviculture gain an automatable 3D structural data basis; full cross-species generality and robustness under extremely defective point clouds remain directions for future work. The extent to which each real-world limitation is resolved is graded in Section 4.3 across the accuracy, workflow, and scope levels.

Limitations and future work. Segmentation-based skeletonization is more time-consuming than graph-based skeletonization; therefore, AdQSM is faster in modeling speed. The former requires a clustering process, which can reduce the negative impact of an incomplete local point cloud spatial distribution, but greatly increases computational complexity. In addition, clustering introduces challenges in class partitioning. In this research we provide a solution by using LOF. More intelligent optimization algorithms can be further attempted. Furthermore, the radius estimation method adopts a unique equation. When the point cloud is complete, the modeling effect is good, but when it is incomplete, there can be a poor effect on the branches and trunk. Last but not least, this study mainly focuses on model reconstruction. Tree modeling is a necessary preparation for extracting phenotypic parameters. Our method has made significant improvements in modeling, which means that in future work, using this method can more accurately extract individual tree measurement factors. Its application to extracting phenotypic parameters remains to be explored. The test set contains a single individual for most species considered, which is sufficient to demonstrate robustness across conditions but not to quantify species-specific behavior. In particular, radius estimation remains less stable for severely incomplete point clouds and at the stump/base region, because the WBE model has a fixed convexity at ground positions; this limitation is partially mitigated by the stump reconstruction in Section 2.4.5 and the plot-level DBH validation in Section 3.4, but extreme missing data still degrade accuracy and are left as future work.

Author Contributions: Conceptualization, T.Y., Y.Z., H.Z. and J.Y.; methodology, T.Y., X.Z. and J.Y.; software, T.Y. and J.Y.; validation, T.Y., X.Z. and J.Y.; formal analysis, T.Y. and X.Z.; investigation, T.Y. and J.Y.; resources, T.Y.; data curation, T.Y. and X.Z.; writing—original draft preparation, T.Y. and J.Y.; writing—review and editing, T.Y., Y.Z. and H.Z.; visualization, T.Y. and X.Z.; supervision, T.Y.; project administration T.Y.; funding acquisition, T.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Fundamental Research Funds of CAF, grant number CAFYBB2022SY035, the Guangdong Forestry Science and Technology Innovation Project,2024KJCX008.

Data Availability Statement: The Beech, Ash, and Oak point clouds, used in this study, is publicly available and can be accessed via https://doi.org/10.25625/FOHUJM. Additional datasets generated or analyzed during the current study, are available from the corresponding author on reasonable request.

Acknowledgments: The authors would like to express their gratitude to the Shanxia forest farm and Malanghu Forest Farm, for providing data collection site. Thank the Institute of Forest Resource Information Techniques, Chinese Academy of Forestry for providing acquisition equipment.

Conflicts of Interest: The authors declare no competing interests.

References

1. Shi, Y. 3D Simulation Theory and Technology of Botanic Tree Based on Scattered Point Cloud. Ph.D. Thesis, Tongji University Press, Shanghai, China, 2018.
2. Niknejad, N.; Bidese-Puhl, R.; Bao, Y.; Payn, K.G.; Zheng, J. Phenotyping of architecture traits of loblolly pine trees using stereo machine vision and deep learning: Stem diameter, branch angle, and branch diameter. *Comput. Electron. Agric.* **2023**, *211*, 107999. [CrossRef]
3. Yang, Y.; Zhang, J.; Wu, K.; Zhang, X.; Sun, J.; Peng, S.; Li, J.; Wang, M. 3d point cloud on semantic information for wheat reconstruction. *Agriculture* **2021**, *11*, 450. [CrossRef]
4. Cao, W.; Chen, D.; Shi, Y.; Cao, Z.; Xia, S. Progress and prospect of lidar point clouds to 3d tree models. *Geomat. Inf. Sci. Wuhan Univ.* **2021**, *46*, 203–220.
5. Sun, Y.; Miao, L.; Zhao, Z.; Pan, T.; Wang, X.; Guo, Y.; Xin, D.; Chen, Q.; Zhu, R. An efficient and automated image preprocessing using semantic segmentation for improving the 3d reconstruction of soybean plants at the vegetative stage. *Agronomy* **2023**, *13*, 2388. [CrossRef]
6. Cárdenas, J.L.; Ogayar, C.J.; Feito, F.R.; Jurado, J.M. Modeling of the 3d tree skeleton using real-world data: A survey. *IEEE Trans. Vis. Comput. Graph.* **2023**, *29*, 4920–4935. [CrossRef]
7. Sun, J.; Wang, P.; Li, R.; Zhou, M.; Wu, Y. Fast tree skeleton extraction using voxel thinning based on tree point cloud. *Remote Sens.* **2022**, *14*, 2558. [CrossRef]
8. Cao, J.; Tagliasacchi, A.; Olson, M.; Zhang, H.; Su, Z. Point cloud skeletons via laplacian based contraction. In Proceedings of the 2010 Shape Modeling International Conference, Aix-en-Provence, France, 21–23 June 2010; pp. 187–197. [CrossRef]
9. Tagliasacchi, A.; Alhashim, I.; Olson, M.; Zhang, H. Mean curvature skeletons. *Comput. Graph. Forum* **2012**, *31*, 1735–1744. [CrossRef]
10. Li, Y.; Su, Y.; Zhao, X.; Yang, M.; Hu, T.; Zhang, J.; Liu, J.; Liu, M.; Guo, Q. Retrieval of tree branch architecture attributes from terrestrial laser scan data using a laplacian algorithm. *Agric. For. Meteorol.* **2020**, *284*, 107874. [CrossRef]
11. Meyer, L.; Gilson, A.; Scholz, O.; Stamminger, M. Cherrypicker: Semantic skeletonization and topological reconstruction of cherry trees. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Vancouver, BC, Canada, 18–19 June 2023; pp. 6244–6253.
12. Huang, H.; Wu, S.; Cohen-Or, D.; Gong, M.; Zhang, H.; Li, G.; Chen, B. L1-medial skeleton of point cloud. *ACM Trans. Graph.* **2013**, *32*, 65. [CrossRef]
13. Ai, M.; Yao, Y.; Hu, Q.; Wang, Y.; Wang, W. An automatic tree skeleton extraction approach based on multi-view slicing using terrestrial lidar scans data. *Remote Sens.* **2020**, *12*, 3824. [CrossRef]
14. Jiang, A.; Liu, J.; Zhou, J.; Zhang, M. Skeleton extraction from point clouds of trees with complex branches via graph contraction. *Vis. Comput.* **2020**, *37*, 2235–2251. [CrossRef]
15. Zhou, J.; Liu, J.; Zhang, M. Curve skeleton extraction via k–nearest–neighbors based contraction. *Int. J. Appl. Math. Comput. Sci.* **2020**, *30*, 123–132. [CrossRef]
16. Cao, W.; Wu, J.; Shi, Y.; Chen, D. Restoration of individual tree missing point cloud based on local features of point cloud. *Remote Sens.* **2022**, *14*, 1346. [CrossRef]
17. Xu, H.; Gossett, N.; Chen, B. Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. Graph.* **2007**, *26*, 19. [CrossRef]
18. Raumonen, P.; Kaasalainen, M.; Åkerblom, M.; Kaasalainen, S.; Kaartinen, H.; Vastaranta, M.; Holopainen, M.; Disney, M.; Lewis, P. Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sens.* **2013**, *5*, 491–520. [CrossRef]
19. You, L.; Sun, Y.; Liu, Y.; Chang, X.; Jiang, J.; Feng, Y.; Song, X. Tree skeletonization with dbscan clustering using terrestrial laser scanning data. *Forests* **2023**, *14*, 1525. [CrossRef]
20. Yang, J.; Wen, X.; Wang, Q.; Ye, J.S.; Zhang, Y.; Sun, Y. A novel scheme about skeleton optimization designed for isttwn algorithm. *Remote Sens.* **2022**, *14*, 6097. [CrossRef]
21. Yang, J. Tree Skeleton Extraction and Modeling Based on TLS Data and Their Applications. Master's Thesis, Nanjing Forestry University, Nanjing, China, 2023.
22. Du, S.; Lindenbergh, R.; Ledoux, H.; Stoter, J.; Nan, L. Adtree: Accurate, detailed, and automatic modelling of laser-scanned trees. *Remote Sens.* **2019**, *11*, 2074. [CrossRef]

23. Fan, G.; Nan, L.; Dong, Y.; Su, X.; Chen, F. Adqsm: A new method for estimating above-ground biomass from tls point clouds. *Remote Sens.* **2020**, *12*, 3089. [CrossRef]
24. Cárdenas, J.L.; López, A.; Ogayar, C.J.; Feito, F.R.; Jurado, J.M. Reconstruction of tree branching structures from uav-lidar data. *Front. Environ. Sci.* **2022**, *10*, 960083. [CrossRef]
25. He, W.; Ye, Z.; Li, M.; Yan, Y.; Lu, W.; Xing, G. Extraction of soybean plant trait parameters based on sfm-mvs algorithm combined with grnn. *Front. Plant Sci.* **2023**, *14*, 1181322. [CrossRef]
26. Fan, G. Development and application of Tree Quantitative Structure Models Based on LiDAR Point Clouds. Ph.D. Thesis, Beijing Forestry University, Beijing, China, 2021.
27. Winiwarter, L.; Esmorís Pena, A.M.; Weiser, H.; Anders, K.; Martínez Sanchez, J.; Searle, M.; Höfle, B. Virtual laser scanning with helios++: A novel take on ray tracing-based simulation of topographic full-waveform 3d laser scanning. *Remote Sens. Environ.* **2022**, *269*, 112772. [CrossRef]
28. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An easy-to-use airborne lidar data filtering method based on cloth simulation. *Remote Sens.* **2016**, *8*, 501. [CrossRef]
29. Yang, J.; Wen, X.; Wang, Q.; Ye, J.S.; Zhang, Y.; Sun, Y. A novel algorithm based on geometric characteristics for tree branch skeleton extraction from lidar point cloud. *Forests* **2022**, *13*, 1534. [CrossRef]
30. Breunig, M.M.; Kriegel, H.P.; Ng, R.T.; Sander, J. Lof: Identifying density-based local outliers. *ACM SIGMOD Rec.* **2000**, *29*, 93–104.
31. Xu, S.; Li, X.; Yun, J.; Xu, S. An effectively dynamic path optimization approach for the tree skeleton extraction from portable laser scanning point clouds. *Remote Sens.* **2021**, *14*, 94. [CrossRef]
32. Rousseeuw, P.J.; Driessen, K.V. Computing lts regression for large data sets. *Data Min. Knowl. Discov.* **2006**, *12*, 29–45. [CrossRef]
33. Wang, W.; Jüttler, B.; Zheng, D.; Liu, Y. Computation of rotation minimizing frames. *ACM Trans. Graph.* **2008**, *27*, 1–18. [CrossRef]
34. Li, G.; Liu, L.; Zheng, H.; Mitra, N. Analysis, reconstruction and manipulation using arterial snakes. *ACM Trans. Graph* **2010**, *29*, 152. [CrossRef]
35. Li, J.; Wu, H.; Xiao, Z.; Lu, H. 3d modeling of laser-scanned trees based on skeleton refined extraction. *Int. J. Appl. Earth Obs. Geoinf.* **2022**, *112*, 102943. [CrossRef]
36. Hui, Z.; Cai, Z.; Liu, B.; Li, D.; Liu, H.; Li, Z. A self-adaptive optimization individual tree modeling method for terrestrial lidar point clouds. *Remote Sens.* **2022**, *14*, 2545. [CrossRef]
37. Hua, W. Trunk Point Extraction and Stem Taper Function Fitting of *Populus* L. Based on Terrestrial Laser Scanner. Master's Thesis, Nanjing Forestry University, Nanjing, China, 2021.
38. West, G.B.; Brown, J.H.; Enquist, B.J. A general model for the structure and allometry of plant vascular systems. *Nature* **1999**, *400*, 664–667. [CrossRef]
39. Murray, C.D. A relationship between circumference and weight in trees and its bearing on branching angles. *J. Gen. Physiol.* **1927**, *10*, 725–729. [CrossRef]
40. Xu, J.; Shan, J.; Wang, G. Hierarchical modeling of street trees using mobile laser scanning. *Remote Sens.* **2020**, *12*, 2321. [CrossRef]
41. Wang, X.; Xiang, H.; Niu, W.; Mao, Z.; Huang, X.; Zhang, F. Oblique photogrammetry supporting procedural tree modeling in urban areas. *ISPRS J. Photogramm. Remote Sens.* **2023**, *200*, 120–137. [CrossRef]
42. Morhart, C.; Schindler, Z.; Frey, J.; Sheppard, J.P.; Calders, K.; Disney, M.; Morsdorf, F.; Raumonen, P.; Seifert, T. Limitations of estimating branch volume from terrestrial laser scanning. *Eur. J. For. Res.* **2024**, *143*, 687–702. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.