Remote Sensing of Environment 335 (2026) 115246

Contents lists available at ScienceDirect

Remote Sensing of Environment

journal homepage: www.elsevier.com/locate/rse

# A three-stage framework for stand-level automated stem volume estimation in temperate forests using Mobile laser scanning

Jinyuan Shao [a], Dennis Heejoon Choi [a,c], Jidong Liu [b], Xiangxi Tian [b], Bina Thapa [a], Seunghyeon Lee [d,e], Ayman Habib [b], Songlin Fei [a,*]

[a] Department of Forestry and Natural Resources, Purdue University, West Lafayette, United States
[b] Lyles School of Civil and Construction Engineering, Purdue University, West Lafayette, United States
[c] School of Life Resources, College of Bio-convergence, Dankook University, Cheonan, Republic of Korea
[d] Interdisciplinary Program in Landscape Architecture, Seoul National University, Seoul, Republic of Korea
[e] Integrated Major in Smart City Global Convergence, Seoul National University, Seoul, Republic of Korea

## HIGHLIGHTS

- Three-stage framework for automated stem detection, segmentation and reconstruction.
- A generalized deep learning model for complex understory points removal.
- Stem detection and segmentation using circle fitting with similarity optimization.
- Stem reconstruction by connecting sector middle points of cross sections.
- More accurate stem volume estimation than existing methods (2% improvement).

## ARTICLE INFO

Edited by Dr Jing M. Chen

*Keywords:*
Stem volume
Mobile laser scanning
Point clouds
Deep learning
3D reconstruction

## ABSTRACT

Accurate stem-level volume estimation at large scale is highly desired in temperate natural forests due to their economic and ecological significance. Mobile Laser Scanning (MLS) systems (e.g., handheld or backpack) offer the ability to efficiently capture high-density point clouds over large areas, creating opportunities for automated, large-scale individual stem volume estimation. However, effective algorithms that can automatically and accurately analyze the dense and complex MLS point clouds of temperate natural forests are lacking. To address this issue, we propose a novel three-stage method to automatically detect, segment, and reconstruct individual stems, enabling direct volume estimations from MLS point clouds of temperate natural forests. First, a deep learning model is employed to separate understory vegetation from overstory trees, reducing point cloud complexity. Next, we introduce a Bidirectional Section Growing (BSG) method for individual stem detection and segmentation, specifically for the segmentation of merchantable logs and multi-stem scenarios, using a novel Least Squares with Similarity Optimization (LeSSO) algorithm. Finally, the Sector Median Points (SMP) method is developed to reconstruct stem shapes for precise volume estimation. Our method is evaluated on four datasets collected in temperate natural forests across the U.S. and Europe. Experimental results demonstrate its superior performance compared to state-of-the-art algorithms, achieving 89.2% Intersection over Union (IoU) for understory removal, 99.4% F-score for stem detection, 91.5% IoU for stem segmentation, and reconstruction accuracy with a Point-to-Mesh distance of 0.0004 $m^2$ and a Chamfer distance of 0.05 m. Moreover, we record 42 stem locations in the field for one of the U.S. datasets and conduct destructive measurements of section-wise diameters for each of them to serve as independent reference data to evaluate stem detection and volume estimation. Our method is able to detect all 42 trees from the point cloud, and reconstructed stem models yield the most accurate section-wise diameter estimates with a Root Mean Square Error (RMSE) of 2.27 cm and $R^2$ of 0.96, and best volume estimation with RMSE of 0.18 $m^3$ and $R^2$ of 0.97. Our method paves the way for automated and accurate estimation of merchantable stem volume from MLS point clouds collected in complex temperate natural forests.

* Corresponding author.
*Email address:* sfei@purdue.edu (S. Fei).

https://doi.org/10.1016/j.rse.2026.115246
Received 1 January 2025; Received in revised form 4 January 2026; Accepted 10 January 2026
Available online 21 January 2026

## 1. Introduction

Accurate stem volume estimation provides a critical basis for the estimation of various forest ecosystem services. Laser scanning techniques can provide high resolution three-dimensional (3D) structures of forests, allowing accurate forest measurement (Dassot et al., 2011; Newnham et al., 2015; Liang et al., 2016). After years of development, Terrestrial Laser Scanning (TLS) has been widely used for stem volume estimation due to its ability to provide precise 3D representations (Liang et al., 2013; Saarinen et al., 2017; Pitkänen et al., 2021). However, multiple scans are needed to capture the complete shape of trees in order to derive accurate stem volume (Pueschel et al., 2013; Liang et al., 2013; Abegg et al., 2017). Therefore, despite its success, the complexity of TLS data collection results in a time-consuming procedure (Pitkänen et al., 2021), making it less practical for measuring the volume of individual stems on a large scale.

The recent development of Mobile Laser Scanning (MLS) can generate detailed 3D point clouds, enabling versatile stand-level forest estimation. Unlike the multiple static placements required for data collection by TLS, MLS can collect data over large areas while in motion and acquire almost equivalent quality point clouds for forest applications (Bauwens et al., 2016; Hyyppä et al., 2020c). However, analyzing high-density, large-scale MLS point clouds requires the development of automated, intelligent algorithms. Although studies have investigated algorithms for tree detection and point cloud segmentation (Hetti Arachchige, 2013; Zhong et al., 2016; Zhang et al., 2019; Shao et al., 2024), few have addressed automated stem volume estimation (but see Hyyppä et al. (2020b); de Paula Pires et al. (2022)), especially for temperate natural forests, which limits the comprehensive use of MLS data for forest analysis.

Two primary challenges exist in developing algorithms for automated individual stem volume estimation using MLS point clouds. The first is robust individual stem segmentation. Existing methods either use circle or cylinder fitting to detect and segment stem points (Burt et al., 2019; Yrttimaa et al., 2020), or calculate geometric features to identify stem points (Hetti Arachchige, 2013; Zhang et al., 2019). The common problem with these methods is their limited ability to detect and segment stems in forest point clouds with dense understory and complex structures (Muhojoki et al., 2024). Furthermore, these methods require parameter tuning to accommodate varying tree sizes and shapes, which undermines their generalizability.

The second challenge is accurate stem shape reconstruction for accurate volume estimation. Since the raw point cloud is only a set a 3D points, it is not feasible to directly estimate stem volume from the unordered point cloud. Current methods estimate stem volume by constructing a stem curve (Liang et al., 2013; Saarinen et al., 2017; Hyyppä et al., 2020b) or reconstructing cylinders (Brede et al., 2019; Bornand et al., 2023), where the cross sections of the stem are assumed to be a perfect circle. This assumption can sometimes lead to inaccurate estimation of the stem volume for timber and lumber.

In this study, we present a three-stage method to automatically estimate individual stem volume for stand-level MLS point clouds of temperate natural forests. Our method consists of a deep learning model for understory removal, a parameter-free algorithm Bidirectional Section Growing (BSG) using Least Squares Similarity Optimization (LeSSO) for stem detection and segmentation, and a mesh reconstruction approach Sector Median Points (SMP) for accurate stem shape reconstruction. Our method mitigates the impact of complex understory on stem detection and segmentation and improves volume estimation accuracy through the accurate reconstruction of stem shape from MLS point clouds. A comparison between our method and other representative approaches is conducted in four datasets of temperate forests in the US, Germany, and Switzerland. Furthermore, destructive field measurements are used to validate our reconstruction algorithm for stem volume estimation. To the best of our knowledge, this is the first study to automatically estimate individual stem volume for stand-level MLS point clouds of temperate natural forests with complex understory.

## 2. Related work

### 2.1. Stem volume from MLS point clouds

There are a few studies that used MLS point clouds for individual tree stem volume estimation. Hyyppä et al. (2020b) first used a backpack MLS system for boreal forests (pine, spruce, birch in Finland) stem volume estimation. They developed an automated algorithm by detecting arcs that constitute stems, and then built stem curves to estimate stem volume. Muhojoki et al. (2024) then used the same method and proved that using MLS point clouds was able to derive more accurate stem volume in boreal forests than using Airborne Laser Scanning (ALS) point clouds. de Paula Pires et al. (2022) also conducted stem volume estimation on boreal forests (spruce, pine and birch) with car-mounted MLS point clouds. A RANSAC-based algorithm was developed to detect arcs or circles, and stem curves were also used for stem volume estimation. For temperate forests, Vandendaele et al. (2022) used hand-held MLS point clouds for a temperate hardwood forest (sugar maple, yellow birch, balsam fir) with little understory in Canada. Due to the concern for inaccurate automated stem segmentation, the authors chose manual extraction of stem points and used a reconstruction algorithm to estimate the merchantable stem volume. To make the whole process automatic, they then built a pipeline by combining the SimpleTree tool (Hackenberg et al., 2015a) to conduct individual tree segmentation and volume estimation on a sugar maple plantation (Vandendaele et al., 2024), but the accuracy of stem detection or tree segmentation was not reported. Despite success, automatic individual stem volume estimation from MLS point clouds of complex temperate natural forests has yet to be explored.

### 2.2. Individual stem segmentation

Accurate individual stem segmentation is the prerequisite for stem volume estimation. Various research has investigated this task which can be divided into two groups: shape-based and feature-based.

**Shape-based.** Shape-based methods detect stem points by finding cylinders or a deck of circles. For example, Burt et al. (2019) find stem points using RANSAC cylinder fitting on the lowest slice of a forest point cloud and then conduct region growing to identify entire stems. Yrttimaa et al. (2020) developed an algorithm to separate stem and non-stem points for single tree point clouds. The algorithm fits cylinders on vertical surface points for each 20 cm horizontal slice. Pitkänen et al. (2019, 2021) detect stems by fitting cylinders on a series of surface patches. Maas et al. (2008); Tansey et al. (2009); Pueschel et al. (2013); Masuda et al. (2021) fit circles on slices along the vertical direction and then combine them into stems. Although simple and straightforward, the presence of a large amount of non-stem points may significantly hinder the segmentation performance of these shape-based methods (Pitkänen et al., 2019). Moreover, shape-based methods require a set of complex rules to determine whether the current primitive belongs to a stem. Taking a twig with an arc shape as an example, one can successfully fit a circle based on this branch but still cannot tell if it belongs to a branch or a part of the stem.

**Feature-based.** Feature-based methods utilize radiometric or geometric features to identify stem points. Some studies have used radiometric features (e.g., laser intensity) to differentiate woody and foliage points (Béland et al., 2011; Tan et al., 2020; Shao et al., 2023). Due to accurate intensity values requiring complicated calibration, the use of radiometric features for stem segmentation is limited (Zhang et al., 2019). Moreover, woody points need to be further processed to extract the clean major stem. Other studies have used geometric features with pure 3D coordinates for stem segmentation. Point-wise geometric features such as linearity, planarity, density and normality are calculated within a pre-defined radius to identify stem points, and then individual stems are clustered to complete stem segmentation (Liang et al., 2011; Zhang et al., 2019; Tao et al., 2021; Xia et al., 2021; Brolly et al., 2021; Zhang et al., 2022). To further increase segmentation accuracy, point-wise geometric features are fed into supervised classifiers (e.g.,

Support Vector Machine, Random Forest) (Chen et al., 2018; Moorthy et al., 2019; Chen et al., 2023). Feature-based methods are able to identify the complex stem shape to some extent, but they rely on point-wise feature computation and manual parameter tuning, which can be very time-consuming.

In summary, both shape-based and feature-based methods have their limitations in environments with complex and dense understory. A large number of studies have shown that dense and irregular understory can lead to a decrease in the accuracy of stem segmentation (Ryding et al., 2015; Liang et al., 2018a,b; Hyyppä et al., 2020a,c, 2021). In addition, too many parameter adjustments would also make algorithms difficult to generalize (e.g., number of points, search radius or feature thresholds, etc.).

### 2.3. Stem volume estimation

Methods for stem volume estimation can be classified into model-based and curve-based.

**Model-based.** Model-based methods assume that a tree stem is constituted by a series of cylinder models with different lengths and orientations, so the stem volume can be calculated by summing up each cylinder model's volume. Thies et al. (2004); Wang et al. (2016); de Conto et al. (2017) used this method to estimate the stem volume by dividing stem points into multiple parts and fitting a cylinder to each part. Most studies estimated stem volume by reconstructing quantitative structure models (QSM). The representative algorithm is TreeQSM developed for TLS data by Raumonen et al. (2013). It constructs multiple cylinders with different diameters, lengths, and orientations to represent the structure of a tree. Then the stem volume can be regarded as the sum of cylinders that represent the stem (Gonzalez de Tanago et al., 2018; Brede et al., 2019; Demol et al., 2021; Bornand et al., 2023). Studies have shown that TreeQSM can be applied to unmanned aerial vehicle (UAV) laser scanning data (Brede et al., 2019), but the performance is generally lower than its performance on TLS data (Sorokina et al., 2026). Model-based methods have been widely used for volume estimation, but cylinders are fundamental primitives, which sometimes cannot fully represent the real shape of a stem. Another problem is that the cylinder-wise representation would cause discontinuities in the profile of a stem, leading to large variance in volume estimation.

**Curve-based.** Curve-based methods aim to build a relationship between stem height and its corresponding diameter, then the stem volume can be derived from this relationship. This method fits circles at regular height intervals along the stem point cloud to estimate diameters, thereby obtaining the stem curve. (Henning and Radtke, 2006; Maas et al., 2008; Liang et al., 2013; Yu et al., 2013). Due to the irregular shape of the cross section or inaccuracies in circle fitting, there are often outliers in a stem curve. To obtain a smoother stem curve, obvious outliers from the diameter set are removed and then a continuous curve can be fitted using spline fitting algorithms (Pueschel et al., 2013; Saarinen et al., 2017, 2019; Hyyppä et al., 2020b; Yrttimaa et al., 2020; Pitkänen et al., 2021). After the smoothed stem curve is derived, the stem volume can be calculated using a predefined formula. The commonly used formula for volume estimation is Huber's formula, which is the product of the height of the cross section and the cross section area, then the whole stem volume can be obtained by accumulating a series of these cross sections. Although effective, all curve-based methods assume that the cross section is circular-shaped, which does not often represent the real shape of the cross section and thus overestimates or underestimates the area, leading to inaccurate volume estimation. In addition, spline curves may also distort actual stem profiles, leading to further errors.

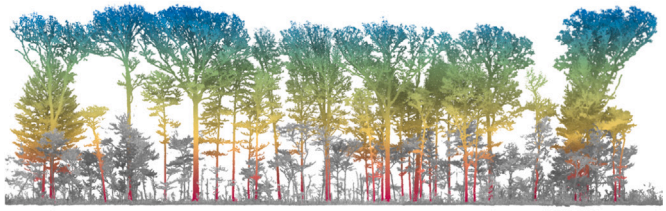
Model-based methods focus on the perspective of shape reconstruction while curve-based methods focus on the relationship between diameters and heights. The common problem with both model-based and curve-based methods is that they don't fully recover the real shape of the stem. Circles and cylinders are simplifications of tree trunk shapes and therefore may overestimate or underestimate volume.

## 3. Materials

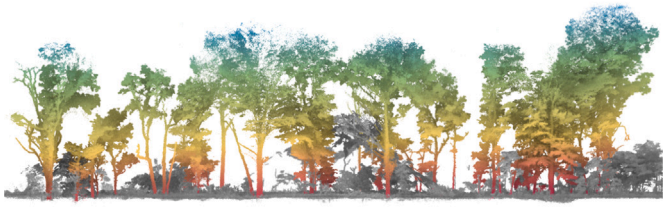
### 3.1. Datasets

Four temperate forests from the U.S. and Europe were selected to evaluate our method. We collected data from two sites in the U.S. and obtained two European datasets from public sources. Fig. 1 shows MLS point clouds from these datasets. Specific information about these datasets is provided in Table 1 and technical specifications for MLS systems are presented in Table 2.

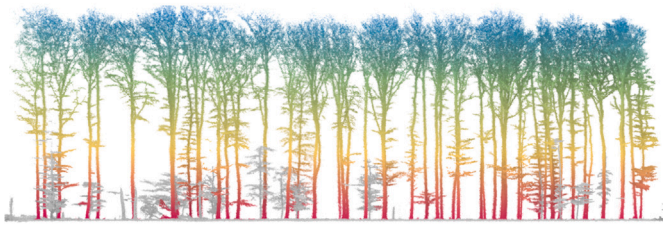
**Dataset I.** The first dataset was collected from Miller Woods (study site I), a temperate natural forest located in Grant County Indiana, USA (40°28'40"N, 85°27'08"W). The major species in Miller Woodland are oak (*Quercus* spp.), black walnut (*Juglans nigra*), and elm (*Ulmus* spp.), and they are common timber species in the eastern U.S. The understory of Miller Woods is dense and relatively challenging to walk through. The data collection campaign was conducted in January, 2024 during the leaf-off season. A backpack MLS system, Hovermap ST developed by Emesent, was used for high-density point cloud collection. Hovermap ST is equipped with a 16 channels laser scanner with a range of 0.4–100 m. To obtain a high-quality point cloud without occlusion, we walked through the forest with a zigzag trajectory



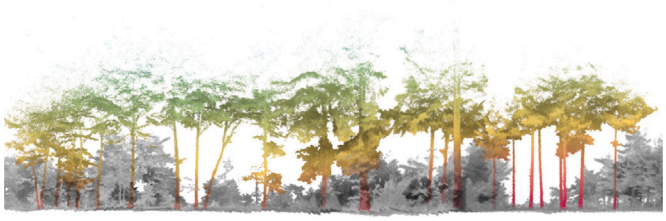
(a) Dataset I; A sample of 31 × 100 m.



(b) Dataset II; A sample of 32 × 105 m.



(c) Dataset III; A sample of 35 × 105 m.



(d) Dataset IV; A sample of 22 × 66 m.

**Fig. 1.** Four datasets used in this study. Understory points are colored by transparent gray to easy show forest structures, and tree points are colored by height.

**Table 1**
Summary of the main characteristics of datasets used in this study.

| Dataset | Country | Dominant tree species | Area ($m^2$) | # of trees | Stem density (stem/ha) | Understory complexity (0.1 m voxels/$m^2$) | Phenology | Average point density (pts/$m^2$) |
|---|---|---|---|---|---|---|---|---|
| Datasets I | US | Oak, walnut, elm | 32,708 | 1436 | 441 | 315 | Leaf-off | 11,554 |
| Datasets II | US | Maple and oak | 25,584 | 429 | 177 | 235 | Leaf-on | 25,033 |
| Datasets III | Germany | European beech | 11,632 | 410 | 349 | 130 | Leaf-off | 4312 |
| Datasets IV | Switzerland | European beech, fir | 7843 | 162 | 204 | 877 | Leaf-on | 15,777 |

**Table 2**
Technical specifications of mobile laser scanning systems used in this study.

| | Manufacturer | Model | # of channels | Maximum range (m) | Wavelength (nm) | Acquisition pattern |
|---|---|---|---|---|---|---|
| Dataset I&II | Emesent Pty Ltd | Hovermap ST | 16 | 100 | 903 | backpack |
| Dataset III | GeoSLAM | ZEB Horizon RT | 16 | 100 | 903 | handheld |
| Dataset IV | GeoSLAM | ZEB REVO RT | None | 30 | 905 | handheld |

and turned around at the edge of the forest to cover the target area. The detailed characteristics of Dataset I are shown in Table 1.

**Dataset II.** The second dataset was collected in Martell Forest (study site II), a temperate natural forest located in Tippecanoe County Indiana, USA (40°26'18"N, 87°01'50"W). The major species in Martell Forest are oak (*Quercus* spp.) and maple (*Acer* spp.), which are common timber sources in the eastern U.S. The understory is less dense than study site I, but we intentionally chose a sloped area to collect data to enrich the variety of our evaluation data. The data collection campaign was conducted in August, 2023 during the leaf-on season. Hovermap ST was also used for data collection with the same data acquisition strategy as Dataset I. The detailed characteristics of Dataset II are shown in Table 1.

**Dataset III.** Dataset III is part of a publicly available dataset (Neudam et al., 2023; Henrich et al., 2023). It was collected in a formerly managed forest located in Lübeck in Schleswig-Holstein, Germany (study site III). The major species in this forest is European beech (*Fagus sylvatica*), which is one of the three biggest timber species in Europe. Although this forest was previously a plantation, it has not been managed since around 1920. Therefore, the structure of the understory is relatively more complex than that of plantations. The data collection campaign was conducted in February, 2021 during the leaf-off season. A handheld MLS system, ZEB Horizon developed by GeoSLAM, was used for this data collection. It should be noted that the dataset has been downsampled to a 1 cm minimum point spacing when published. The detailed characteristics of Dataset III are shown in Table 1.

**Dataset IV.** Dataset IV was originally published by (Kükenbrink et al., 2022). It is a mixed temperate natural forest close to Zurich, Switzerland (study site IV). This is a natural forest with European beech (*Fagus sylvatica*), sliver fir (*Abies alba*) and Norway spruce (*Picea abies*) as the dominant species. The understory structure is the most complex among the four datasets. MLS point cloud data was collected during the leaf-on season in October, 2020. ZebRevo RT, a handheld MLS system was used for this data collection. The detailed characteristics of Dataset IV are shown in Table 1.

### 3.2. Data annotation

The objective of data annotation is to evaluate algorithms' performance. We first manually segment individual trees from the entire point cloud regardless of their sizes. After individual trees are manually segmented, we conduct the least squares circle fitting at breast height (1.3 m to 1.5 m) for each of them to obtain DBH. If the DBH is greater or equal to 12 cm, then we label it as a tree, while all trees with DBH less than 12 cm are labeled as understory. The reason for using this criterion is that trees with a DBH less than 12.7 cm are considered seedlings and saplings, which do not provide merchantable timber. For trees with DBH slightly larger than 12.7 cm (e.g., 12.7 - 13 cm), the DBH obtained by circle fitting may be slightly smaller than 12.7 cm due to the interference of point cloud noise. This could result in omissions. To avoid this problem, we apply a more conservative threshold of 12 cm, which ensures trees with DBH slightly larger than 12.7 cm are annotated. At the same time, we annotate the center of the fitted circle as the stem location and set it as the reference location. We then further annotate stem points from trees point clouds. The criteria we used to define the stem is from the bottom to the bole height (i.e., first major branching point) of a tree, a common practice for timber harvest in the US. All datasets used in this study have been annotated based on processes described above. It should be noted that we count completely scanned stems within the target area as a reference, while trees without complete scanning (outside of the target area) are not the target of our study, as stem volume cannot be accurately estimated from incomplete scanned stems. All operations related to manual annotation are performed on CloudCompare (CloudCompare, 2024), an open-source point cloud processing software.

### 3.3. Destructive field measurement

To further evaluate methods for stem reconstruction and volume estimation, we performed destructive measurements of actual stem sizes in Miller Woods (Dataset I) after data collection. We carefully selected 42 representative merchantable trees (i.e., trees that have timber market value) and worked with a professional forester to fell these trees. All felled trees were laid on the ground to conduct size measurements. For each stem, we first recorded its location and base height. Afterwards, we used calipers to measure diameters every foot (30.48 cm). Considering the differences between diameter measurements from different directions with the caliper, we measured two diameters from two perpendicular directions following the practice in Saarinen et al. (2017) to mitigate the influence of irregular stem shapes. Finally, the average value of the two diameters from perpendicular directions was calculated as the reference diameter at each measurement height. The instrumental uncertainty of the caliper we used is ± 0.127 cm.

## 4. Method

Our automatic individual stem volume estimation method is presented below: (1) understory removal; (2) individual stem detection and segmentation; (3) stem reconstruction. In addition, we introduce the baselines and metrics used for the evaluation of each task.

### 4.1. Understory removal

We adopt a 3D U-Net deep learning model to distinguish the understory and overstory trees from MLS point clouds. The 3D U-Net deep

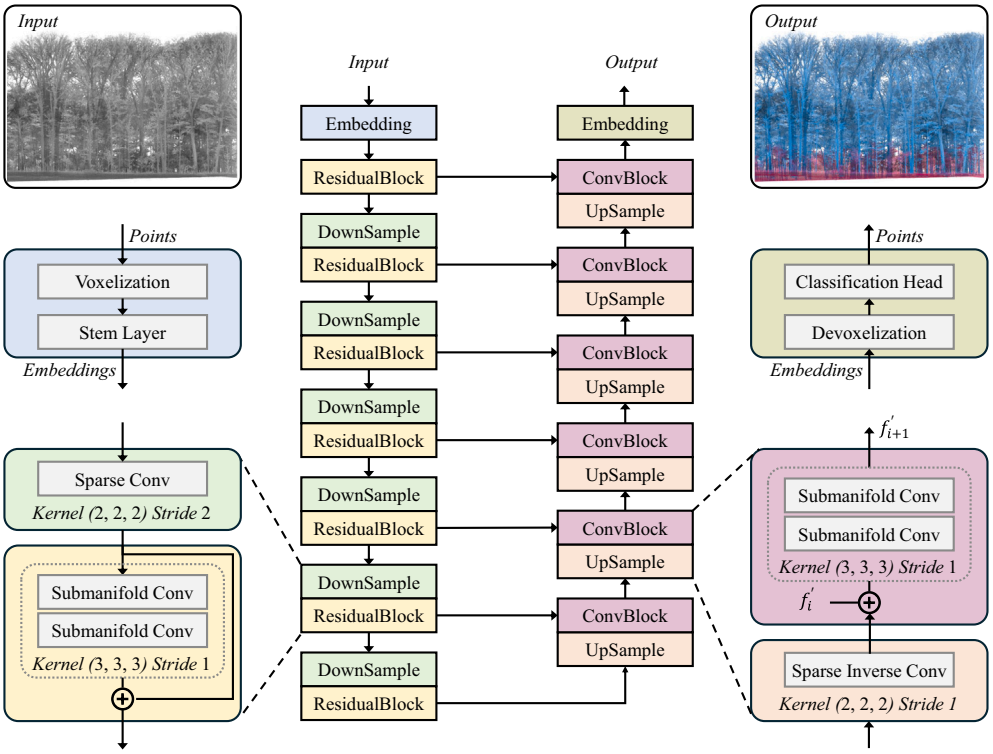



**Fig. 2.** Model architecture for understory removal. The input point cloud is processing by volixzation and a stem layer to obtain an embedding vectors. Then the data passes through the encoder with residual blocks for feature extraction with gradually reducing spatial dimensions and increasing feature channels. In the decoder, it is processed through upsampling and convolutional blocks to gradually recover the original spatial dimension. Meanwhile, skip connections are used to merge features from encoder into decoder. Finally, point-wise classification results are obtained from devoxelization and classification head.

learning model takes a forest point cloud as input and provides the point-wise classification results (i.e., understory vegetation or overstory trees) as output. The model architecture and training details are introduced below.

**Model architecture.** Fig. 2 shows details of the model architecture. It follows the previous successful design for point cloud processing (Jiang et al., 2020; Vu et al., 2022), with the encoder-decoder (U-shaped) paradigm, wherein features are progressively downsampled to capture multi-scale context and subsequently upsampled to recover spatial resolution (Ronneberger et al., 2015). The model takes the unordered point cloud as input and converts it to ordered volumetric grids. The voxelized data are first fed into a stem layer to obtain embeddings and then processed by the encoder. The encoder contains residual blocks and downsampling blocks for multi-scale feature extraction. Each residual block consists of two convolutional layers and a shortcut connection. The two convolutional layers process the input embeddings, and the shortcut connection adds the identity of the input embeddings to the convolutional output as the final output for the residual block. The role of shortcut connection is to prevent the gradient vanishing during model training (He et al., 2016). After feature extraction by the residual block, the data are downsampled using a 3D convolutional operator with a stride of 2. This operation simultaneously reduces the spatial dimension and adds 32 feature channels to the existing structure. The decoder has a symmetrical structure with the encoder. It contains upsampling blocks and convolutional blocks to recover the spatial resolution and decrease the number of feature channels. Meanwhile, skip connections are used for feature merging between residual blocks in the encoder and convolutional blocks in the decoder. After the decoder, the data are remapped to point-wise features with devoxelization and processed by a classification head to obtain point-wise classification results. We utilize the sparse convolution technique to build the entire model (Spconv, 2022). The biggest advantage of using sparse convolution is that it stores and processes only non-empty voxels, significantly reducing memory usage and enhancing computational efficiency.

**Model training.** Training a deep learning model requires a large amount of data. However, there are very few publicly available datasets designed for point cloud understory removal, especially MLS point clouds. To solve this problem, we consider fine-tuning a well-pre-trained model for our task. ScanNet is a widely used dataset in the field of 3D deep learning (Dai et al., 2017). It uses RGB-D sensors to capture 1513 scans of indoor scenes and generate voxelized point clouds with annotations of 20 semantic classes and instance labels and it has been proven to be useful for pre-training (Xie et al., 2020; Wu et al., 2023). We first train our model from scratch on the ScanNet data to obtain a pre-trained model. Subsequently, we use data from Shao et al. (2024) to fine-tune the model to conduct domain adaptation. It should be noted that all four datasets in this study are not used for model training. In both pre-training and fine-tuning stages, we use cross-entropy loss for model optimization (Mao et al., 2023) and AdamW as the optimizer (Loshchilov, 2017). In addition, the early stopping technique is used to prevent overfitting.

### 4.2. Individual stem detection and segmentation

We introduce a Bidirectional Section Growing (BSG) strategy for stem detection and merchantable log segmentation in point clouds, which does not rely on point-wise geometric feature calculations or parameter tuning. This approach defines stem detection and segmentation as the task of finding the complete shape of cross section in point clouds. To achieve this, we develop the Least Squares with Similarity Optimization (LeSSO) module, which not only fits circles on primitives but also determines whether a primitive represents a complete cross section. Unlike previous methods that require complex rules to determine if a point cloud primitive belongs to a stem, BSG eliminates this dependency by

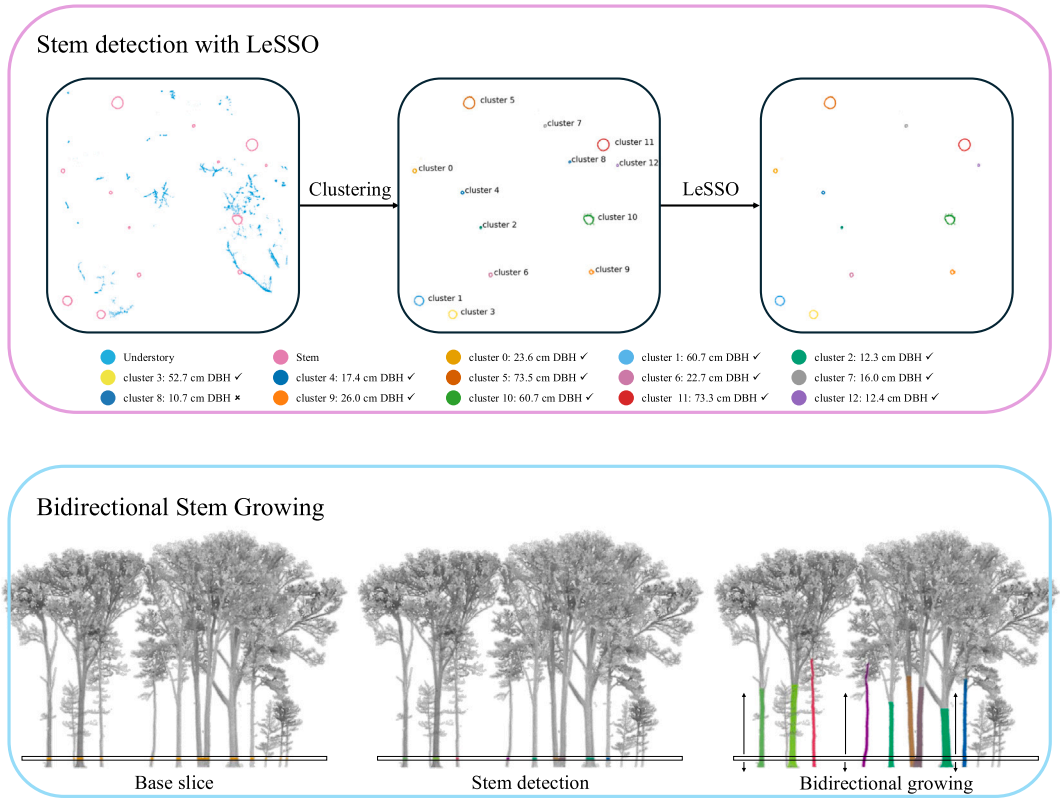



**Fig. 3.** Illustration of stem detection (top) and stem segmentation (bottom). Stem detection: we first take a base slice at the breast height and remove the understory points. Then Euclidean clustering is used to obtain individual clusters. Finally, LeSSO is used for each cluster to determine stems. Stem segmentation: for each detected stem cross section, we gradually add bounding boxes to both up and down and use LeSSO to determine cross sections. For upside growing, it stops at the first branching point.

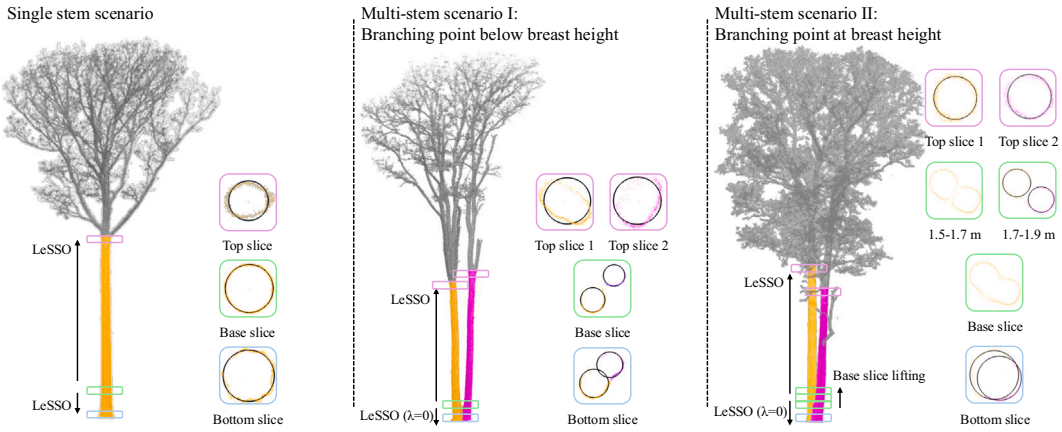



**Fig. 4.** Three different scenarios for stem growing. For single-stem scenario, we apply LeSSO for both upward and downward growing due to BSG not stopping until the lowest point. For multi-stem scenario I, cross sections of a multi-stem tree can be determined on base slice, but BSG stops during the downward growing, so we set $\lambda$ to 0 and LeSSO can determine incomplete cross sections. For multi-stem scenario II, LeSSO may miss the stem on the base slice as the bifurcation occurs at breast height with irregular shape, so we lift the base slice twice to find complete cross sections. Once stems are detected on next two slice, we process it as multi-stem scenario I.

iteratively detecting stem cross sections from breast height upwards and downwards. See Fig. 3 for a visual explanation.

**Pre-processing.** First, a point cloud is height-normalized following common practices (Cabo et al., 2018; Lin et al., 2021, 2022). The height normalization process begins with the application of the Cloth Simulation Filtering (CSF) algorithm to identify ground points (Zhang et al., 2016). Then, the Digital Terrain Model (DTM) with 0.5 m resolution is created from ground points, and interpolated to fill blank holes. The point cloud is normalized by following $z_i - z_i^{\mathrm{DTM}}$, where $z_i$ is the z-value of the $i$th point, and $z_i^{\mathrm{DTM}}$ is the z-value of the corresponding DTM grid onto which the point is projected.

**Base slice clustering.** After height normalization, we remove the understory points and take a slice at breast height (1.3 to 1.5 m) as the base slice to conduct stem detection. Euclidean Clustering is used to cluster the base slice point cloud into smaller primitives (Rusu, 2010). For each primitive, we conduct LeSSO as detailed below to detect stems at the base slice.

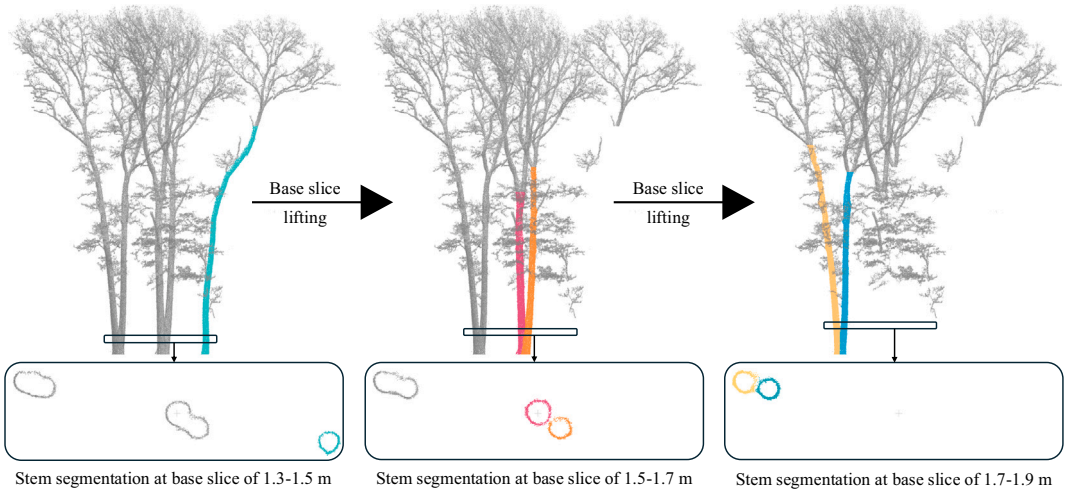



**Fig. 5.** The workflow of base slice lifting. We first perform stem segmentation based on a base slice of 1.3–1.5 m. Then we remove all the segmented stem points and lift the base slice to 1.5–1.7 m to conduct stem segmentation again. If LeSSO can determine new cross sections on this base slice, new stems are segmented following multi-stem scenario II. This process of base slice lifting is performed twice in total until 1.9 m and the whole process is automated.

**Stem detection with Least Squares with Similarity Optimization (LeSSO).** The classic least squares method performs circle fitting by minimizing the sum of squared residuals for all points:

$$\sum_{i=1}^{N} \left( \|\boldsymbol{p}_i - \boldsymbol{o}\|_2 - r \right)^2 \tag{1}$$

where $\boldsymbol{o} = (x_o, y_o)$ is the circle center, $r$ is the circle radius, $N$ is the number of points in the cluster $\boldsymbol{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$ and each $\boldsymbol{p}_i = (x_i, y_i)$ is a point in $\mathbb{R}^2$. However, even if a circle can be fitted, we still don't know whether this cluster belongs to a stem cross section. For example, an arc-shaped branch cluster can still be fitted with a circle through least squares circle fitting, but this cluster is actually not a stem. Therefore, least squares circle fitting can only find an optimal circle that crosses most points of the cluster, but cannot determine whether the cluster is a stem cross section (it is also applicable for RANSAC). In order to determine whether a cluster truly belongs to a stem cross section, we introduce a similarity optimization in the least squares process by assuming that a stem cross section has a similar shape as the fitted circle. The objective function $f$ is defined as:

$$f_{LeSSO} = \sum_{i=1}^{N} \left( \|\boldsymbol{p}_i - \boldsymbol{o}\|_2 - r \right)^2 + \lambda \sum_{\boldsymbol{c} \in \boldsymbol{C}} \min_{\boldsymbol{p}^c \in \boldsymbol{P}} \|\boldsymbol{c} - \boldsymbol{p}^c\|_2 \tag{2}$$

where $\boldsymbol{c} = (x_c, y_c)$ represents a point evenly sampled from the fitted circle $\boldsymbol{C}$, $\boldsymbol{p}^c$ represents points from the cluster $\boldsymbol{P}$, and $\lambda$ is the regularization parameter that controls the significance of the similarity in the whole objective function. The first term minimizes the sum of circle fitting residuals for all points, which is the same as in the classic circle fitting we have described above. The second term minimizes the sum of the closest distances from each $\boldsymbol{c}$ to the cluster $\boldsymbol{P}$. For each point $\boldsymbol{c}$ in the fitted circle $\boldsymbol{C}$, we find its closest point $\boldsymbol{p}^c$ in the cluster $\boldsymbol{P}$ and calculate the distance. Therefore, the sum of distances from $\boldsymbol{c}$ to $\boldsymbol{p}^c$ can be used to measure the similarity between the fitted circle and the cluster. If a cluster is a complete circle, the similarity value should be small, achieving the optimization goal; if a cluster is a semicircle or arc, the similarity value should be high, because some points on the fitted circle are far away from their nearest points; if a cluster is an irregular closed shape, such as a cross section at a branching point, the similarity value is also high, because most points fall outside the fitted circle. It should be noted that we set lambda to 1 to make sure we detect stem cross sections with complete and circular shapes. By applying LeSSO, we can detect the stems on the base slice.

**Bidirectional section growing.** After each stem section is detected at the base slice, we segment the entire stem by using our BSG strategy for each stem. We create a bounding box with a length of $L$ and a width of $W$ according to the stem section at breast height, and it shares the same center as the fitted circle from LeSSO in the stem detection. Then we vertically move this bounding box along the center and expand the length and width of the bounding box by two times simultaneously to avoid missing points. Subsequently, on the newly obtained section, we execute LeSSO to determine whether a new stem section exists. If a stem cross section can be determined, we repeat the above process, and create a new bounding box based on the newly grown cross section. For upward growing, we stop when LeSSO cannot detect the cross section, which means that it has reached the first branch; for downward growing, if LeSSO can detect the circle at the base, it means that the tree is a single-stem tree; if LeSSO stops before reaching the base, it means that there could be multiple stems sharing the same tree base (i.e., forked trees). Our solution for these multi-stem scenarios is described below.

**Multi-stem scenarios.** Our method may encounter two kinds of multi-stem scenarios: 1) downward growing of BSG stops before reaching the base; 2) multi-stem cross sections occur at the base slice so LeSSO cannot determine stems. For the first case, we aim to split the shared tree base into multiples and each of them belongs to a specific stem side. We adapt $\lambda$ to 0 to continue executing downward growing. As $\lambda$ is set to 0, incomplete cross sections can also be detected. We only recognize points around the fitted circle as stem points. In this way, tree base points belonging to a certain stem side can be segmented. For the second case, LeSSO may struggle to detect multi-stem cross sections when bifurcation occurs at breast height because multi-stem cross sections sometimes exhibit low similarity scores with the fitted circle. To address this issue, we filter out the segmented stem points after stem segmentation and keep the remaining points that potentially contain multi-stem trees, then lift the base slice on the remaining points and re-apply LeSSO stem detection on new base slices. Base slice lifting is conducted twice, the first lifting is to change the base slice from 1.3–1.5 m to 1.5 to 1.7 m, and the second lifting is to change the base slice from 1.5–1.7 m to 1.7–1.9 m. The reason for elevating the base slice is to cross the branching point to obtain more regular cross sections so that LeSSO can determine them. If stems

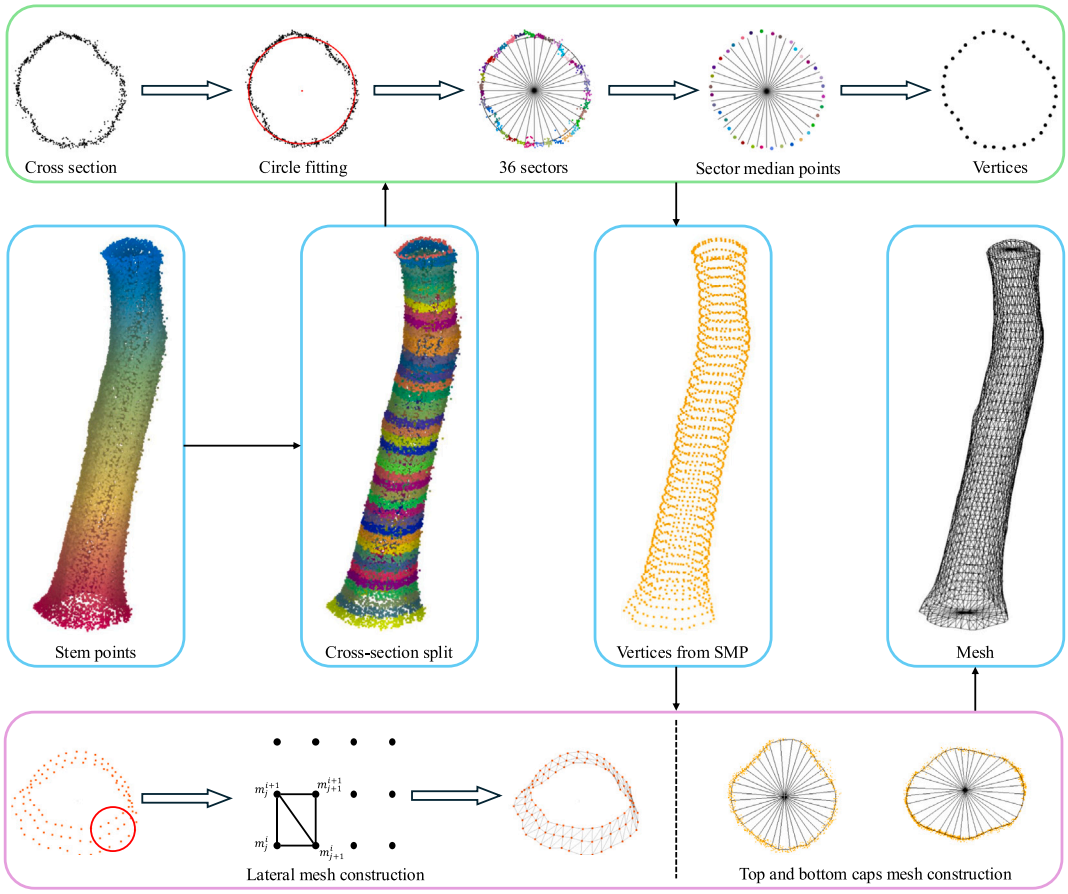



**Fig. 6.** Sector Median Points (SMP) for stem reconstruction. A stem point cloud is first split into multiple horizontal slices. For each slice, median points from 36 sectors are obtained as vertices for mesh model reconstruction. Then, triangles are constructed on vertices with lateral order. Finally, bottom and top caps are constructed to make the model watertight to be calculated volume.

are identified, the BSG method is subsequently employed for stem segmentation. Fig. 4 shows different multi-stem scenarios and Fig. 5 shows the workflow of base slice lifting.

### 4.3. Stem reconstruction

We propose a new Sector Median Points (SMP) method to reconstruct the shape of the cross section from MLS point clouds. We set this task to progressively fit the shape of each cross section, ultimately reconstructing the entire stem, as illustrated in Fig. 6 for the reconstruction process.

**Cross section split.** Starting from the lowest point, we denote a height $h_i$ every 10 cm upwards. For each $h_i$, we take a 20 cm slice by extending 10 cm above and below $h_i$. For the bottom slice $h_1$, which is the lowest one, we take 10 cm points above $h_1$ since there is no point lower than the bottom slice $h_1$. For the top slice, we take the points 10 cm below it. In this way, each $h_i$ has a 20 cm thick slice except for the bottom and top slices which have 10 cm thick slices.

**Cross section reconstruction.** For each slice, a circle is first fitted using least squares circle fitting. This circle is then evenly divided into 36 sectors based on the circle's center. For $j$ th sector $S_j$, all points that fall within this sector are considered, regardless of whether they are inside or outside the initially fitted circle. Then we calculate the median $x$ and $y$ of points within the $j$ th sector and use $h_i$ as the $z$ value to create a 3D median point $\boldsymbol{m}_j^i = (x_{S_j}, y_{S_j}, h_i)$ for each sector $S_j$ at the height $h_i$. If there is no point in a specific sector, we use the median point of the fitted circle falling in this sector as the median values of $x$ and $y$. For the cross section at height $h_i$, 36 median points $\boldsymbol{m}_j^i$ are derived for mesh reconstruction in total.

**Mesh construction.** All median points reconstructed from the last step are considered as the vertices of a stem mesh model. The vertices are denoted as $\boldsymbol{m}_j^i$, which represent the calculated median points for the $j$ th sector median point at the height $h_i$. We construct triangles in the lateral order of the sectors for each layer from bottom to top. For any median point $\boldsymbol{m}_j^i$, we take $\boldsymbol{m}_j^i$ itself and its neighbor $\boldsymbol{m}_{j+1}^i$ as the base angles of the triangle, and the point $\boldsymbol{m}_j^{i+1}$ in the $i+1$ layer at the height $h_{i+1}$ is counted as the vertex angle, to construct a triangle. At the same time, $\boldsymbol{m}_j^{i+1}$ and $\boldsymbol{m}_{j+1}^{i+1}$ in the $i+1$ layer are used as the base angles of the adjacent triangles, and $\boldsymbol{m}_{j+1}^i$ is used as the vertex angle of the adjacent triangle. In this order, it produces continuous triangles in the lateral direction around the stem by connecting end to end. We also construct bottom and top caps to make the mesh model watertight. The fitted circle centers for the $i$ and $i+1$ layers are denoted as $\boldsymbol{o}^i$ and $\boldsymbol{o}^{i+1}$, which are used as the vertex angles of the triangles for the bottom and top caps. Then, $\boldsymbol{m}_j^i$ and $\boldsymbol{m}_{j+1}^i$ at the bottom cap, and $\boldsymbol{m}_j^{i+1}$ and $\boldsymbol{m}_{j+1}^{i+1}$ at the top cap are used as the base angles to form a triangle specifically. In this way, the top and bottom caps and the horizontal mesh can form a watertight model, so the volume can be directly calculated using surface integrals.

### 4.4. Evaluation

We evaluate our proposed method from three perspectives: 1) understory removal; 2) individual stem detection and segmentation; and 3) stem reconstruction. For each task, we carefully selected several representative methods for comparison. Models and metrics used in the

evaluation process are detailed below. All baselines and experiments are executed on a computer with an Intel i7-11700 CPU, 64 GB RAM, and an NVIDIA RTX A5000 GPU with 24 GB.

#### 4.4.1. Understory removal

The removal of understory points can be viewed as a binary segmentation task that classifies the points in the forest into understory vegetation and overstory trees. Therefore, we use precision, recall, F-score, and IoU as metrics to evaluate understory removal. However, to our knowledge, there is no method or model specifically developed for understory removal. Therefore, we use a popular point cloud classification model, CANUPO as a baseline to compare with our model (Brodu and Lague, 2012). The core idea of CANUPO is to characterize the local dimensionality properties at different scales for each point and classify them using a linear probabilistic classifier. We use the CANUPO implementation (Brodu and Lague, 2012) in CloudCompare (CloudCompare, 2024). One should note that we use manually annotated datasets described in Section 3.2 as a reference.

#### 4.4.2. Individual stem detection and segmentation

We evaluate stem detection and stem segmentation separately in this part. Stem detection focuses on the accuracy of stem localization, while stem segmentation focuses of the accuracy on stem point recognition from the bottom to the first major branch.

**Stem detection.** For stem detection, we choose TreeLS, FORTLS, and 3DFin as baselines to assess. TreeLS and FORTLS are two R packages developed for forest inventory with TLS or MLS point clouds, and 3DFin is an open-source tool for forest inventory that can also be used on TLS or MLS point clouds. In addition, to evaluate the effectiveness of understory removal and its impact on different algorithms, we conduct stem detection experiments using point clouds before and after understory removal. The former is the original point cloud, while the latter is the point cloud processed by our understory removal 3D UNet. Completeness, omission error, commission error and F-score are used as evaluation metrics following previous research (Yin and Wang, 2016; Liang et al., 2018a). Similar to understory removal, reference data are from manually labeled datasets. As all models produce point-wise stem detection results, we set True Positives (TP) as detected stem points spatially containing the stem location of reference, i.e., the stem location of reference data falls into the bounding box of detected stem points. False Positives (FP) are set as detected points that don't contain any stem location of reference data, and False Negatives (FN) are set as reference stem locations that are not contained by any detected stems. Other than comparing detections with annotations in point clouds, we also use 42 field recorded stem locations described in Section 3.3, as independent reference data, to evaluate detection performance at the individual stem level. In this experiment, we set reference locations that spatially match with detected stems as TP, and reference locations that do not spatially match with any detected stems as FN. To determine TP or FN, we calculate the distance between each reference stem and its nearest detected stem in the point cloud. If the distance is small enough, we set it as TP; if there is no close enough detected stem, we set it as FN. Following Hyyppä et al. (2020b); Kükenbrink et al. (2025), the threshold is set as 0.5 m. The minimum distance between these trees is 7.5 m, so the 0.5 m threshold would not affect the matching results. TP, FN, completeness and omission are reported for stem detection before/after understory removal, and baselines are used as described above.

**Stem segmentation.** In terms of stem segmentation, although there are many excellent stem segmentation approaches, they do not focus on first-branch stem segmentation, so they cannot be directly compared with our approach. ForestSPG, a deep learning model developed for first-branch stem segmentation, is used as the baseline for comparison (Shao et al., 2024). Precision, recall, F-score, and IoU are used for stem segmentation because it is a binary segmentation task. For fair comparison, we use default parameters for all baselines.

#### 4.4.3. Stem reconstruction

**Reconstruction quality**. Three reconstruction methods are selected to compare with our stem reconstruction algorithm. They are Poisson surface reconstruction, TreeQSM and TreeQSM.Triangulation. Poisson reconstruction is a classic surface reconstruction algorithm, and TreeQSM is the most commonly used model-based method to reconstruct individual trees to estimate volume or biomass. Note that TreeQSM.Triangulation is an option of TreeQSM that reconstructs the shape of buttress roots. The objective of stem reconstruction is to recover the real shape of a stem as much as possible; therefore, it is necessary to measure the similarity between the original point cloud and the reconstructed stem. Here we use Chamfer distance (CD) and Point-to-Mesh (P2M) distance to evaluate the reconstruction quality. Chamfer distance computes the average distance from each point in one point set to its closest point in the other point set; P2M distance is a sum of Point-to-Face (P2F) distance and Face-to-Point (F2P) distance, where P2F is the average value of the squared distance of each point in the point cloud to the closest triangular face in the mesh and F2P is the averaged value of the squared distance of each triangular face in the mesh to the closest point in the point cloud. Their joint use is a common practice to robustly validate the quality of reconstruction (Zong et al., 2023; Hao et al., 2024; Chen et al., 2024). This experiment is conducted on all annotated stem point clouds across the four datasets.

**Section-wise diameter estimation.** To verify whether the reconstructed stem can accurately represent the actual stem shape, we also compare it with the field measurements collected from 42 trees. Our assumption is that if diameters derived from a reconstructed stem are close to field measured diameters at the same heights, then it can recover the shape of the whole stem as much as possible so that it can be used for accurate volume derivation. Therefore, we estimate diameters from each reconstructed tree model at heights that are the same as the field measurements. Diameters are derived by the least squares circle fitting on the corresponding cross sections of 12.7 cm thickness (6.35 cm up and down). Original stem points, Poisson surface reconstruction, TreeQSM, TreeQSM.Triangulation and SMP are used for comparisons. In terms of evaluation metrics, we use MAE, MSE, RMSE, $R^2$ and bias to indicate accuracy. MAE reflects the average magnitude of errors, MSE emphasizes the impact of large deviations, RMSE provides a unit-consistent summary of overall accuracy, $R^2$ quantifies the goodness of fit between estimated and reference values, and bias indicates systematic overestimation or underestimation.

**Volume estimation.** We compare the performance of volume estimation in reconstructed stems from different methods. The way to obtain reference data for the volume of each tree follows Saarinen et al. (2017) and Hyyppä et al. (2020b). First, a stem curve is derived based on field derived section-wise measurements, and then a volume value is calculated based on this curve as the reference data. Original stem points, Poisson surface reconstruction, TreeQSM, TreeQSM.Triangulation and SMP are used to estimate volume and these volume estimates are compared with the reference data. For original stem points and Poisson surface reconstruction, we derive stem curves and calculate volume as their estimation, as volume cannot be directly obtained from them. For TreeQSM, TreeQSM.Triangulation and SMP, we use their mesh volume as estimations. MAE, MSE, RMSE, R2 and bias are used to evaluate performance.

## 5. Results

### 5.1. Understory removal

Our proposed model effectively distinguishes understory vegetation and overstory trees as illustrated in Fig. 7 and Table 3. Visually, Fig. 7 reveals that our model effectively distinguishes understory and overstory points, whereas CANUPO struggles with accurate recognition, particularly in regions with dense and complex understory. Quantitatively, our deep learning model outperforms CANUPO across all datasets and metrics. Overall, our method achieves a precision of 94.2%, recall of 94.8%,

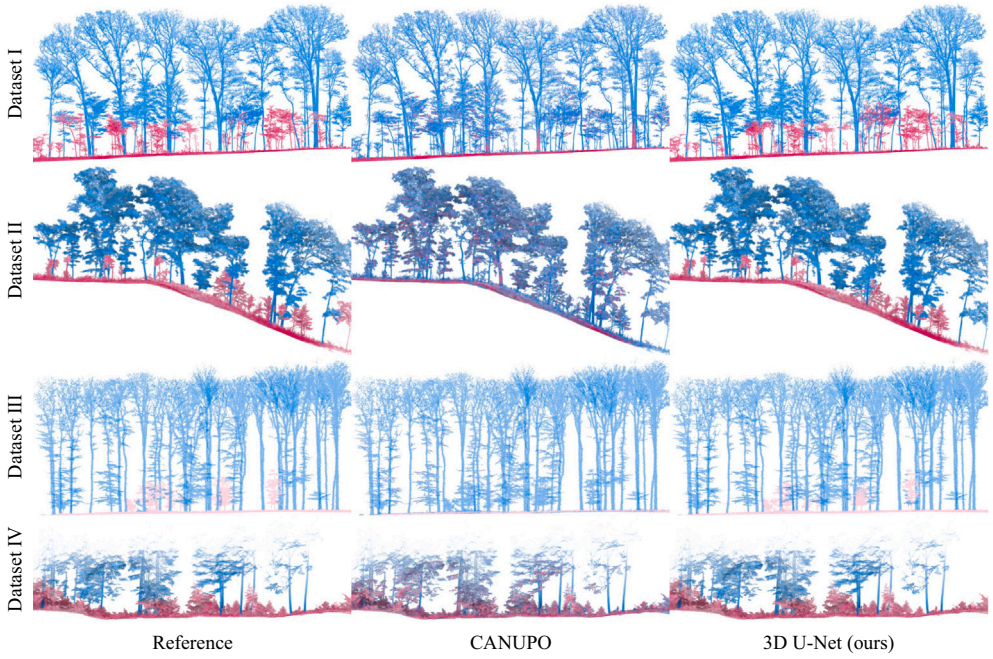



**Fig. 7.** Understory removal visualization of reference, CANUPO, and our 3D U-Net. 3D U-Net successfully distinguish understory and overstory points, while CANUPO hardly deal with complex forest structures.

**Table 3**
Understory removal results with CANUPO and our 3D U-Net. Our model achieves better precision, recall, F-score and IoU than the feature-based CANUPO.

| | | Precision | Recall | F-score | IoU |
|---|---|---|---|---|---|
| Dataset I | CANUPO | 77.7 | 76.5 | 74.5 | 59.4 |
| | Ours | **92.6** | **98.7** | **95.4** | **91.3** |
| Dataset II | CANUPO | 70.7 | 70.7 | 70.7 | 54.9 |
| | Ours | **99.5** | **99.4** | **99.4** | **98.9** |
| Dataset III | CANUPO | 90.5 | 86.5 | 88.3 | 80.0 |
| | Ours | **98.0** | **98.2** | **98.1** | **96.3** |
| Dataset IV | CANUPO | 62.4 | 72.3 | 63.5 | 50.2 |
| | Ours | **91.1** | **91.8** | **91.0** | **83.5** |
| Overall | CANUPO | 75.5 | 75.4 | 74.2 | 59.0 |
| | Ours | **94.2** | **94.8** | **94.3** | **89.2** |

F-score of 94.3%, and an IoU of 89.2%. These results highlight a consistent and substantial improvement over CANUPO, which reports an overall precision of 75.5%, recall of 75.4%, F-score of 74.2%, and IoU of 59.0%. Both methods perform best on Dataset III, which had the simplest understory vegetation structure. Our model achieves an IoU of 96.3%, while CANUPO reached 80.0%. This suggests that simpler understory patterns are more easily identified, even by the traditional feature-based approach. The most challenging dataset is Dataset IV, characterized by the highest understory vegetation complexity. IoUs drop to 50.2% for CANUPO and 83.5% for our method. Despite the reduced performance, our model maintains a significant margin over CANUPO, reflecting its robustness in complex scenarios. In summary, CANUPO's performance degrades significantly as the understory complexity increases, whereas our model maintains high consistency in precision, recall, F-score, and IoU across all datasets.

### *5.2. Individual stem detection and segmentation*

**Stem detection.** Table 4 summarizes the performance of four different stem detection methods on four datasets, both before and after understory removal. Overall, LeSSO achieves the highest completeness (i.e., rate of successful stem detection) (74.6% pre-removal, 98.7% post-removal) and best F-score (85.2%/99.4%). In contrast, TreeLS shows the largest omission error (82.2%/79.2%) and FORTLS obtains the highest commission error (25.1%/15.9%). Dataset III, which contains the simplest understory, yields relatively high performance for all methods, with completeness values above 45.9% and F-scores exceeding 62.3% after understory removal. However, for the other three datasets with more complex understory, TreeLS and FORTLS do not achieve satisfactory results even on post understroy removal datasets, registering overall completeness of 20.8% and 36.0%, respectively, and F-scores of only 33.9% and 50.4%. It's worth noting that LeSSO acquired very

**Table 4**
Stem detection results (before/after understory removal) using different methods. LeSSO obtains highest detection accuracy, and understory removal improves most methods' performance.

| | | Completeness | Omission | Commission | F-score |
|---|---|---|---|---|---|
| Dataset I | TreeLS | 16.7/17.9 | 83.3/82.1 | 13.7/6.9 | 28.0/30.0 |
| | FORTLS | 31.5/32.0 | 68.5/68.0 | 20.0/19.7 | 45.3/45.8 |
| | 3DFin | 53.6/63.2 | 46.4/36.8 | 11.4/**0.0** | 66.8/77.4 |
| | LeSSO (Ours) | 73.3/**99.0** | 26.7/**1.0** | 0.9/**0.0** | 84.2/**99.5** |
| Dataset II | TreeLS | 2.8/2.8 | 97.2/97.2 | 14.3/20.0 | 5.4/5.4 |
| | FORTLS | 22.8/29.8 | 77.2/70.2 | 21.0/14.7 | 35.4/44.2 |
| | 3DFin | 52.0/56.2 | 48.0/43.8 | 28.8/9.4 | 60.1/69.4 |
| | LeSSO (Ours) | 58.0/**96.7** | 42.0/**3.3** | 0.8/**0.0** | 73.2/**98.3** |
| Dataset III | TreeLS | 36.8/49.3 | 63.2/50.7 | 14.7/6.0 | 51.4/64.6 |
| | FORTLS | 41.0/45.9 | 59.0/54.1 | 3.4/3.1 | 57.5/62.3 |
| | 3DFin | 68.8/70.7 | 31.2/29.3 | 7.2/1.4 | 79.0/82.4 |
| | LeSSO (Ours) | 98.8/**100.0** | 1.2/**0.0** | 0.0/**0.0** | 99.4/**100.0** |
| Dataset IV | TreeLS | 22.2/19.1 | 77.8/80.9 | 33.3/31.1 | 33.3/30.0 |
| | FORTLS | 52.5/63.0 | 47.5/37.0 | 59.5/19.7 | 45.7/70.6 |
| | 3DFin | 86.4/90.1 | 13.6/9.9 | 51.4/**0.0** | 62.2/94.8 |
| | LeSSO (Ours) | 69.1/**98.6** | 30.9/**1.4** | 1.8/**0.0** | 81.2/**99.3** |
| Overall | TreeLS | 17.8/20.8 | 82.2/79.2 | 12.7/8.8 | 29.4/33.9 |
| | FORTLS | 33.0/36.0 | 67.0/64.0 | 25.1/15.9 | 45.8/50.4 |
| | 3DFin | 58.1/65.0 | 41.9/35.0 | 20.3/1.8 | 67.2/78.2 |
| | LeSSO (Ours) | 74.6/**98.7** | 25.4/**1.3** | 0.8/**0.0** | 85.2/**99.4** |

**Table 5**
Stem detection results of 42 field recorded tree locations (before/after understory removal). LeSSO obtains the highest detection accuracy, and understory removal improves most methods' performance.

| | TP | FN | Completeness | Omission |
|---|---|---|---|---|
| TreeLS | 22/24 | 20/18 | 52.4/57.1 | 47.6/42.9 |
| FORTLS | 24/24 | 18/18 | 57.1/57.1 | 42.9/42.9 |
| 3DFin | 40/**42** | 2/**0** | 95.2/**100** | 4.8/**0** |
| LeSSO (Ours) | 39/**42** | 3/**0** | 92.7/**100** | 7.1/**0** |

**Table 6**
Stem segmentation performance from ForestSPG and BSG. BSG shows significant improvement compared to ForestSPG.

| | | Precision | Recall | F-score | IoU |
|---|---|---|---|---|---|
| Dataset I | ForestSPG | 82.4 | 74.2 | 78.1 | 64.1 |
| | BSG (ours) | **94.5** | **98.7** | **96.6** | **93.3** |
| Dataset II | ForestSPG | **94.3** | 58.7 | 72.4 | 56.7 |
| | BSG (ours) | 92.3 | **94.3** | **93.3** | **87.5** |
| Dataset III | ForestSPG | **95.4** | 71.5 | 81.7 | 69.1 |
| | BSG (ours) | 93.0 | **100.0** | **96.4** | **93.0** |
| Dataset IV | ForestSPG | 96.2 | 69.0 | 90.4 | 67.2 |
| | BSG (ours) | **99.5** | **99.0** | **99.3** | **98.5** |
| Overall | ForestSPG | 89.2 | 66.6 | 76.2 | 61.6 |
| | BSG (ours) | **94.3** | **96.9** | **95.6** | **91.5** |

low commission error across all datasets, meaning almost all stems detected belong to the TP group without any FP. Regarding sensitivity to understory removal, LeSSO benefits the most, increasing its completeness by more than 20% (from 74.6% to 98.7%) and its F-score by over 14% (from 85.2% to 99.4%). 3DFin also improves its F-score by 11% after understory removal, whereas TreeLS and FORTLS show no clear improvement. Table 5 presents the performance of these four stem detection methods on the independent reference data, i.e., field recorded stem locations. LeSSO and 3DFin detect all stems after understory removal. Additionally, all methods except FORTLS show improved completeness following understory removal.

**Stem segmentation.** Our BSG method also achieves better stem segmentation results comparing to other existing methods as illustrated in Fig. 8 and Table 6. In Fig. 8, we compare the reference data (left column), ForestSPG results (middle column), and BSG results (right column) for each of the four datasets. Overall, BSG shows more complete stem segmentation, while ForestSPG often misclassifies points in the middle section of a stem. When small branches or foliage are attached to the stem, ForestSPG tends to either mislabel these regions as stem or exclude them entirely, whereas BSG more robustly preserves and correctly labels these small branches as non-stem. BSG also handles multi-stem cases better by distinguishing two stems sharing the same base. Fig. 9 shows a leanning stem case, BSG can segment leanning stem points to to the first-branch point, while ForestSPG is not able to recognize leaning stems. In some cases, BSG's segmentation extends

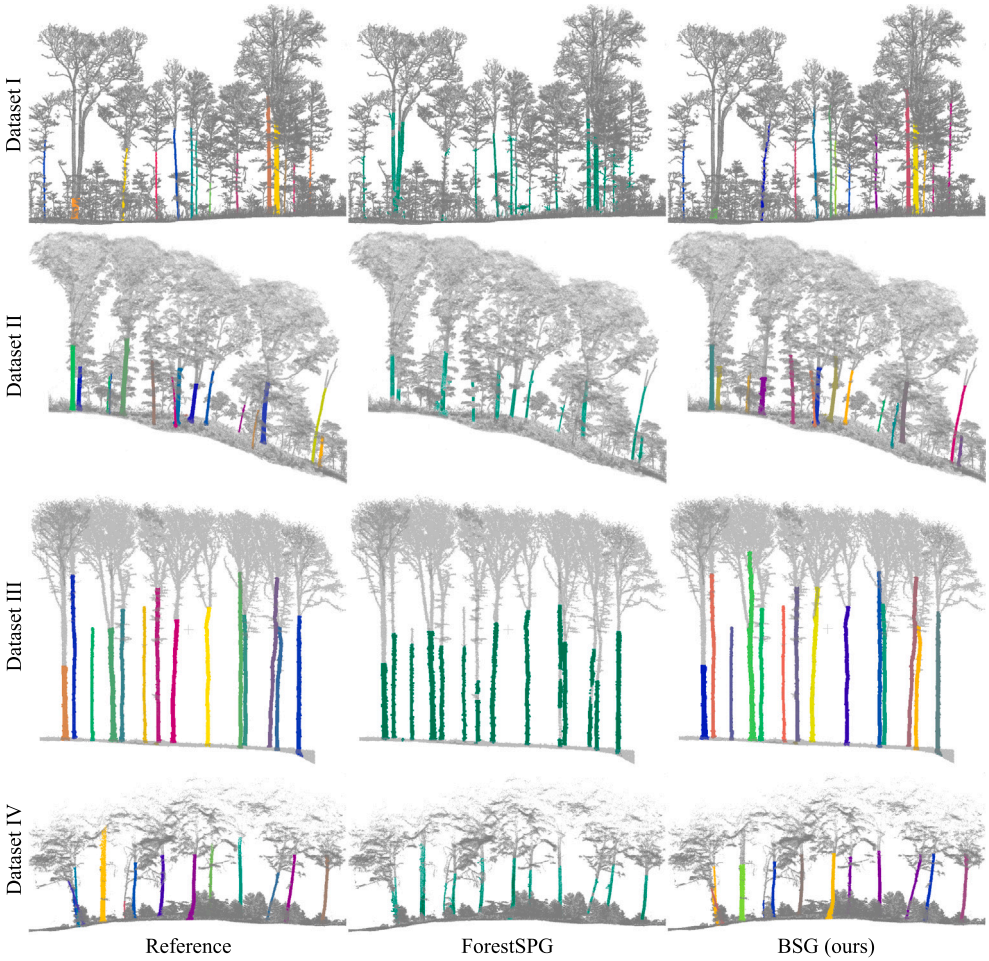



**Fig. 8.** Stem segmentation visualization of reference, ForestSPG, and BSG. BSG is able to segment more complete stems than ForestSPG.

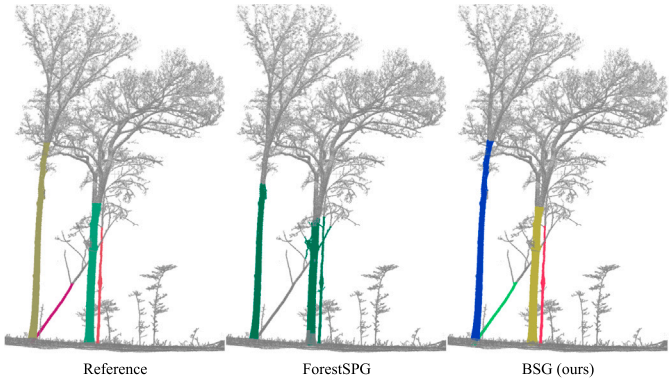



**Fig. 9.** Leaning stem segmentation with different methods. From left to right: reference, ForestSPG, and BSG (ours). The proposed BSG can successfully segment the leaning stem while ForestSPG cannot.

beyond the first bifurcation, continuing up to the treetop. Another distinction is that ForestSPG provides only binary segmentation (stem vs. non-stem), whereas BSG assigns separate instance labels for each stem. Table 6 reports quantitative comparisons. Across all datasets, BSG outperforms ForestSPG on every metric, achieving an overall precision of 94.3%, recall of 96.9%, F-score of 95.6%, and IoU of 91.5%. ForestSPG's performance is notably lower than other methods (89.2% precision, 66.6% recall, 76.2% F-score, 61.6% IoU). Examining the results by each dataset, we see that ForestSPG's precision exceeds its recall in each case, indicating that many stem points are missed. This is consistent with Fig. 8, where ForestSPG overlooks part of the stems. In contrast, BSG consistently maintains recall rates above 94.3% for all datasets, underscoring its robustness in identifying stem points even under conditions with complex understory.

### 5.3. Stem reconstruction

#### 5.3.1. Reconstruction quality

Overall, our proposed SMP method demonstrates the most accurate alignment with the point cloud among all methods used, exhibiting minimal artifacts (Fig. 10). In contrast, TreeQSM produces the poorest results showing that its discontinuous section-wise cylinder-based reconstruction approach fails to accurately capture the true shape of the stems. TreeQSM.Triangulation can produce continuous surfaces, but they are not smooth on the noisy side. The Poisson reconstruction method exhibits relatively stable performance; however, its reliance on normal estimation results in stem surfaces with excessive small bumps. Additionally, it is worth noting that, unlike SMP, the reconstructed meshes produced by the three baseline methods are not fully closed, meaning they are not watertight. This limitation necessitates further processing to estimate the volume of the mesh, which introduces potential errors. The results of the quantitative reconstruction are presented in Table 7. SMP overall outperforms baseline methods, achieving a

**Table 7**
Reconstruction quality comparison between different methods. SMP and Poisson reconstruction achieve lower P2M distance and Chamfer distance than TreeQSM and TreeQSM.Triangulation.

| | | P2M ($m^2$) | CD (m) |
|---|---|---|---|
| Dataset I | Poisson Reconstruction | **0.0002** | 0.0931 |
| | TreeQSM | 0.0021 | 0.2170 |
| | TreeQSM.Triangulation | 0.0003 | **0.0444** |
| | SMP (Ours) | **0.0002** | 0.0454 |
| Dataset II | Poisson Reconstruction | 0.0035 | 0.0761 |
| | TreeQSM | 0.0008 | 0.1768 |
| | TreeQSM.Triangulation | 0.0005 | 0.0451 |
| | SMP (Ours) | **0.0003** | **0.0448** |
| Dataset III | Poisson Reconstruction | 0.0005 | 0.1053 |
| | TreeQSM | 0.0009 | 0.1754 |
| | TreeQSM.Triangulation | 0.0004 | **0.0467** |
| | SMP (Ours) | **0.0003** | 0.0488 |
| Dataset IV | Poisson Reconstruction | 0.0022 | 0.0714 |
| | TreeQSM | 0.0014 | 0.3620 |
| | TreeQSM.Triangulation | 0.0042 | 0.0670 |
| | SMP (Ours) | **0.0005** | **0.0512** |
| Overall | Poisson Reconstruction | 0.0013 | 0.0839 |
| | TreeQSM | 0.0015 | 0.2328 |
| | TreeQSM.Triangulation | 0.0015 | 0.0508 |
| | SMP (Ours) | **0.0004** | **0.0476** |

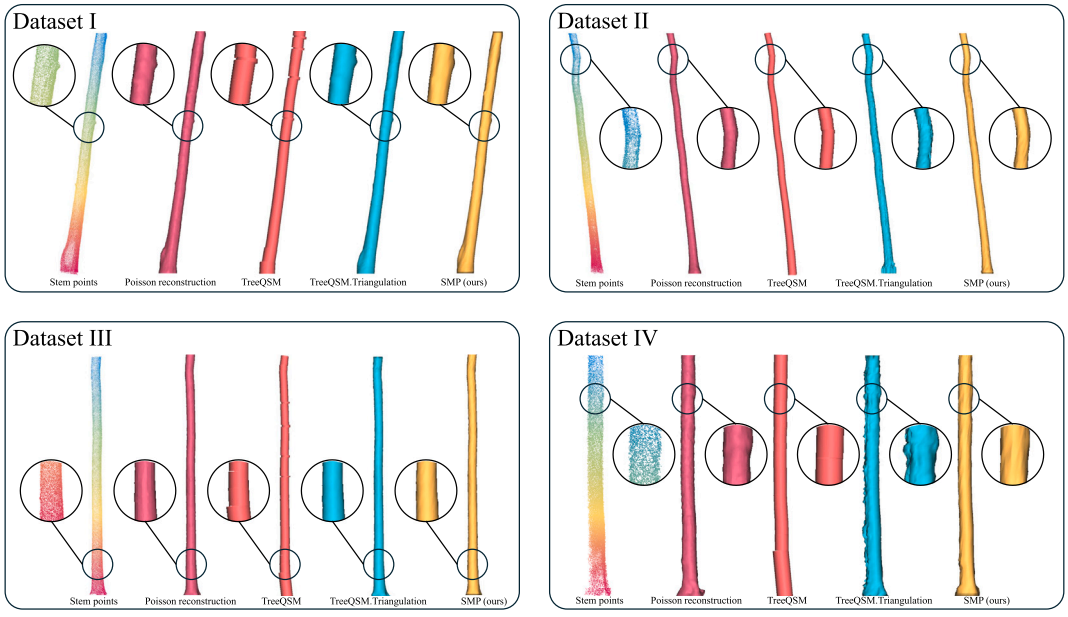



**Fig. 10.** Visual comparison of stem reconstruction between different methods. Poission reconstruction, TreeQSM.Triangulation, and SMP produce continuous mesh surfaces. TreeQSM.Triangulation introduces surface artifacts on the side with noise points, whereas occasionally there are gaps between cylinders that make up TreeQSM models.

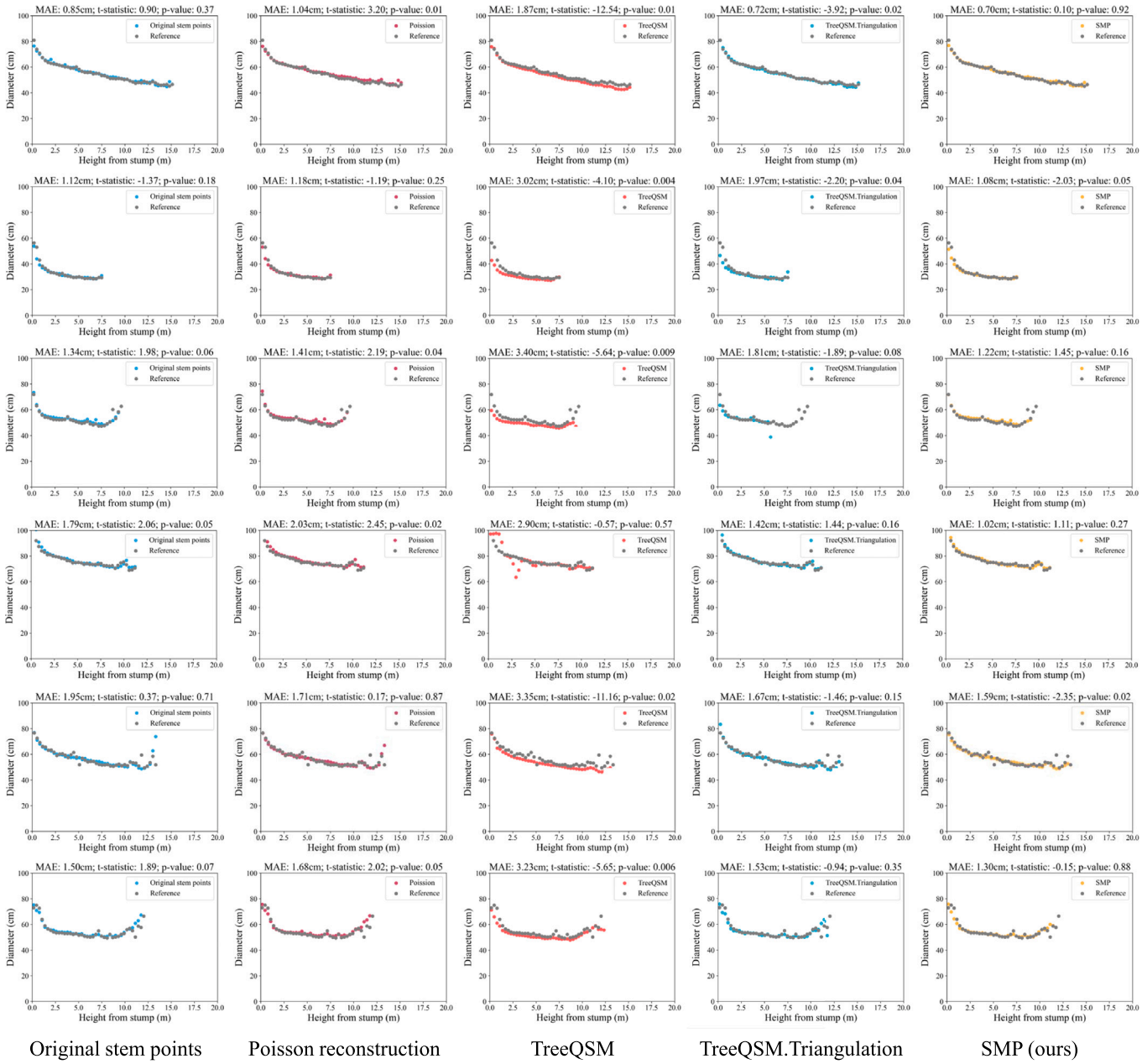



**Fig. 11.** Section-wise comparison between estimated diameters and manual measurements. Each panel shows diameter versus height from stump for a single stem. MAE is reported to show the difference, paired-t and p-value are reported to show the statistical difference. SMP shows closest agreement with manual measurements. Among the six samples showed with five different methods, 11 out of 30 show statistically significant differences ($p < 0.05$).

mean P2M distance of 0.0004 $m^2$ and a mean Chamfer distance of 0.0476 m. TreeQSM.Triangulation produces the second best results in terms of Chamfer distance, and performs better than SMP on Dataset I and Dataset III. In comparison, the Poisson reconstruction exhibits weaker performance, with a mean P2M distance of 0.0013 $m^2$ and a mean Chamfer distance of 0.0839 m. In summary, the proposed SMP method demonstrates similar or better performance compared to other representative methods, both visually and quantitatively.

### 5.3.2. Section-wise diameter estimation

We report section-wise diameter measurements obtained in the field versus those derived from various reconstruction methods. Fig. 11 shows six examples in terms of the relationship between diameters and height. From top to bottom: a regular shaped stem, a short stem, a short stem with various defects, a stem with a knob, a stem with large forks at the top, and a stem with a large base and large forks. Raw points, Poisson reconstruction and SMP closely match the section-wise manually measured diameters. While TreeQSM.Triangulation captures the overall trends, it occasionally produces outliers. Additionally, although TreeQSM follows the general trend of manual measurements, it sometimes doesn't fit the subtle change in the diameters versus heights trend. Fig. 12 shows scatter plots for the relationship between all reference diameters and estimated diameters, and Table 8 presents MAE, MSE, RMSE, $R^2$, and bias in detail. Among all evaluated methods, SMP achieves the highest accuracy, with the lowest MAE (1.58 cm), MSE (6.41 cm), RMSE (2.27 cm), and best $R^2$ (0.96) and bias ($-0.27$ cm). Raw point clouds, the second-best performer, exhibit slightly higher

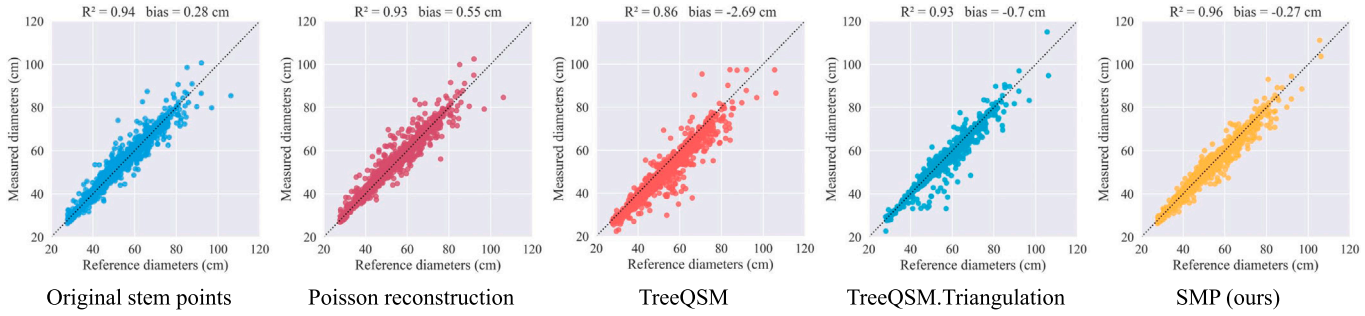



**Fig. 12.** Scatter plots of section-wise diameter measurements for all 42 trees. From left to right: original stem points, poisson reconstruction, TreeQSM, TreeQSM.Triangulation, and our SMP. The black dashed line is the 1:1 reference line. SMP achieves highest $R^2$ and best bias.

**Table 8**
Comparison of section-wise diameter measurements using different methods on 42 trees with destructive measurements. SMP obtains the best results compared to manually measured data in terms of MAE, MSE, RMSE bias and $R^2$.

| | MAE (cm) | MSE ($cm^2$) | RMSE (cm) | $R^2$ | bias (cm) |
|---|---|---|---|---|---|
| Original stem points | 1.75 | 8.86 | 2.64 | 0.94 | 0.28 |
| Poisson Reconstruction | 1.88 | 9.42 | 2.80 | 0.93 | 0.55 |
| TreeQSM | 3.09 | 18.72 | 3.99 | 0.86 | −2.69 |
| TreeQSM.Triangulation | 1.87 | 11.36 | 2.92 | 0.93 | −0.70 |
| SMP (Ours) | **1.58** | **6.41** | **2.27** | **0.96** | **−0.27** |

**Table 9**
Comparison of stem volume estimation using different methods on 42 trees with destructive measurements. SMP obtains the best results compared to manually measured data in terms of MAE, MSE, RMSE bias and $R^2$.

| | MAE ($m^3$) | MSE ($m^6$) | RMSE ($m^3$) | $R^2$ | bias ($m^3$) |
|---|---|---|---|---|---|
| Original stem points | 0.22 | 0.18 | 0.43 | 0.85 | 0.11 |
| Poisson Reconstruction | 0.19 | 0.08 | 0.28 | 0.93 | 0.05 |
| TreeQSM | 0.35 | 0.20 | 0.45 | 0.84 | 0.21 |
| TreeQSM.Triangulation | **0.18** | 0.06 | 0.24 | 0.95 | 0.06 |
| SMP (Ours) | **0.18** | **0.03** | **0.18** | **0.97** | **−0.04** |

errors, with an MAE of 1.75 cm, MSE of 8.86 cm, RMSE of 2.64 cm, $R^2$ of 0.94 and bias of 0.28 cm. In contrast, Poisson reconstruction and TreeQSM.Triangulation yield less accurate results, with MAEs of 1.88 cm and 1.87 cm, respectively and TreeQSM demonstrates relatively larger errors with an MAE of 3.09 cm and $R^2$ of 0.86.

### 5.3.3. Volume estimation

Volume estimation results for 42 stems are presented in Fig. 13 and Table 9. Fig. 13 shows scatter plots for the relationship between reference volume and derived volume, and Table 9 shows different metrics. SMP achieves the best $R^2$ of 0.97, bias of 0.18 $m^3$, MAE of 0.18 $m^3$, MSE of 0.03, and RMSE of 0.18 $m^3$. Overall, SMP and TreeQSM.Triangulation have accurate volume estimation with fewer outliers, while the other three have more outliers. Poisson reconstruction takes third place with $R^2$ of 0.93, while raw points and TreeQSM obtain $R^2$ less than 0.9, with 0.85 and 0.84.

## 6. Discussion

Obtaining spatially explicit individual stem volume information of temperate forests has both ecological and economic significance. However, substantial challenges remain when automating this process with MLS systems, especially in natural forests with complex understory and irregular shapes of stems. To tackle these challenges, we introduce a three-stage method by removing understory points to mitigate the influence of the complex forest structures, identifying stem cross sections to detect and segment individual stems, and reconstructing 3D stem shapes. The effectiveness of our method has been demonstrated by comparing it with other representative methods in four datasets. To our knowledge, this is the first work using MLS point clouds to automatically estimate individual stem volume at the stand-level in temperate natural forests with complex understory.

### 6.1. Effects of understory removal

Numerous studies have demonstrated that understory impacts the performance of stem detection (Ryding et al., 2015; Liang et al., 2018a,b; Hyyppä et al., 2020a,c, 2021). Although many methods, such as shape-based and feature-based approaches, have been developed for stem detection, the complexity of understory vegetation often hinders shape fitting or feature extraction, leading to omissions in stem detection (Muhojoki et al., 2024). To address this challenge, we leverage the power of deep learning to directly remove understory vegetation,

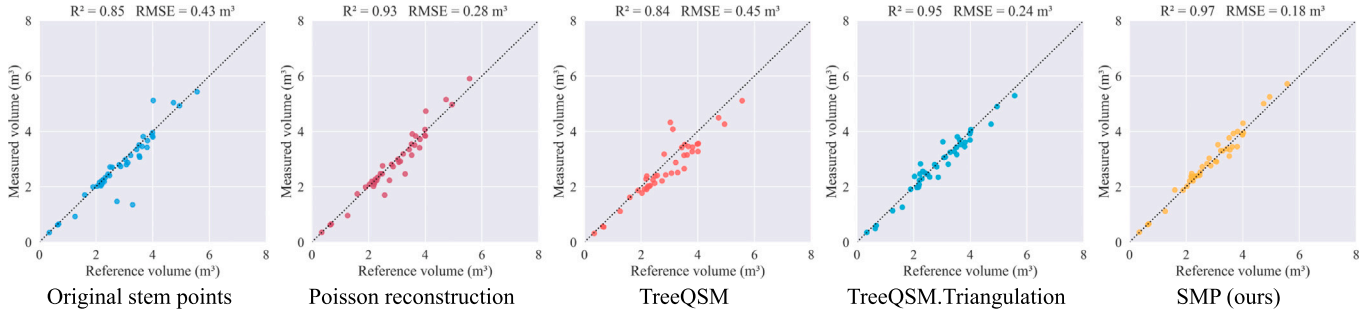



**Fig. 13.** Scatter plots of volume estimation for all 42 reference trees. From left to right: point cloud, poission reconstruction, TreeQSM, TreeQSM.Triangulation, and our SMP. The black dashed line is the 1:1 reference line. SMP achieves highest $R^2$ and best bias.

thereby mitigating its influence and improving stem detection accuracy. Experimental results show that our 3D U-Net model distinguishes understory points from overstory trees, outperforming commonly used machine learning methods such as CANUPO. In addition, we report the specific impact of understory removal on stem detection. Our experiments reveal that the understory removal improves all methods' stem detection accuracy across all datasets. Beyond enhancing stem detection, the results of understory removal also provide future opportunities for more detailed analysis of understory seedlings and saplings, broadening the potential applications of this approach.

### 6.2. Detection and segmentation of merchantable log

In temperate natural forests, multi-stemmed trees commonly exist (Tanentzap et al., 2012), yet previous studies have largely overlooked this case. Our method finds stem cross sections on the base slice at breast height, effectively identifying complete cross sections even in multi-stem cases. For highly irregular multi-stem cross sections at the base slice, we address this challenge by lifting the base slice to the higher levels. Regarding stem segmentation, the stem from bottom to the first branching point is the primary source of timber that accounts for the majority of a tree's biomass in temperate forests. Current approaches either rely on a predefined fixed strip height (Brodu and Lague, 2012) or segment the entire stem without considering the first branching height (Heinzel and Huber, 2016; Zhang et al., 2019), missing critical ecological and economic information. The most comparable method to ours is ForestSPG, which utilizes a deep learning model to segment merchantable logs (Shao et al., 2024). However, it does not account for multi-stem trees or assign instance labels to individual stems. In contrast, our method addresses the multi-stem scenario prevalent in temperate natural forests and accurately segments stems from bottom to the first major branch, assigning instance labels to each merchantable stem. With our approach, spatially explicit stand-level information on merchantable logs can be obtained directly from MLS point clouds, offering valuable insights for both ecological assessment and economic evaluation. However, it should be acknowledged that our algorithm is only tested on fully scanned stems with DBH larger than 12 cm as we focus on merchantable log, while it would fail for incomplete scanned stems, and its performance on small trees is yet to be tested. Meanwhile, the effectiveness of our algorithm in the case of tree trunks with holes also needs further evaluation. In terms of computational efficiency, our approach is not fast enough on very large point clouds due to frequent points query and cross sections detection. For comparison, when running Dataset I (32708 $m^2$ with 400 million points) with 3DFin (CloudCompare plugin) and our algorithm (understory removal and stem segmentation) on the same computer configuration (Intel i7-11700 with 64GB RAM), our algorithm (5.8 hours) takes almost twice as long as 3DFin (2.5 hours).

### 6.3. Shape reconstruction for stem volume

In this study, we derive stem volume by reconstructing accurate 3D shapes of stems. Traditionally, stem volume is estimated by measuring DBH and tree height, combined with species-specific allometric equations (West, 2009). However, this approach is inherently limited, as the volume derived from allometric equations is an estimate based on relatively simple equations rather than the shape of stems. Given the variability in tree shapes and resultant allometric relationships (i.e., approximation of volume based on DBH and height), especially in temperate forests, such estimations can have high uncertainties (Vorster et al., 2020). Advancements in laser scanning technology overcome these limitations by capturing complete 3D structural information of stems for volume derivation. Existing methods typically approximate the stem as a series of cylinders or use stem curves for volume calculation (Hackenberg et al., 2015b; Pitkänen et al., 2021; Saarinen et al., 2017), and most of them use TLS data. Sorokina et al. (2026) found that noise in point clouds affects the accuracy and logical consistency of TreeQSM, which means TreeQSM could produce relatively lower performance on MLS data with noise. Our method reconstructs the shape of each cross section and generates 3D stem models to derive their volume for MLS data. More importantly, obtaining the median point for each section can offset the effect of noise, which is especially suited for MLS point clouds. Experimental results demonstrate that our approach achieves the closest agreement with destructive measurements and most accurately represents the original stem shapes. Compared to other methods, such as Poisson surface reconstruction and TreeQSM, our SMP approach accurately captures the true shape of the cross section while accounting for noise in MLS point clouds, making it better suited for stem reconstruction. Another advantage is that the 3D model reconstructed by SMP is guaranteed to be watertight, so the stem volume can be calculated directly. In contrast, even if other reconstruction methods can obtain accurate shapes, the stem volume can only be derived by constructing a stem curve or integrating cross sections. In terms of the generalizability of SMP, we have tested its effectiveness on the four datasets that have different point densities, forest types, and are from different mobile laser scanners, but users may need to adjust parameters if their data characteristics are too different from the data we have tested. With our method, not only can stem volume be precisely calculated, but the reconstructed geometry can also be utilized to estimate timber volume. This advancement enhances both the ecological and economic assessments of forest resources.

### 6.4. Limitations and future work

There are several limitations in this study. First, understory removal relies on GPU-based deep learning, which could limit its accessibility for users with restricted computational resources. And the computational efficiency of our method is suboptimal. Compared to popular forest inventory tools such as 3DFin, our approach takes two to three times longer (see details in Section 6.2) due to frequent point queries and circle-fitting operations. Second, the BSG method occasionally segments tree stems above the first branch. As illustrated in Fig. 8, the upward growing of the fourth tree (from left to right) in Dataset III does not terminate at the first branching point, leading to higher stem segmentation. We believe this occurs because one of the branches at the bifurcation is too thin, allowing LeSSO to identify a cross section despite the branching. Third, the use of an average value of two caliper measurements is based on the assumption that cross sections are relatively circular, but it may not represent the true diameter of less circular shapes of cross sections. A tape measurement is likely to provide a closer diameter estimate; however, it is impractical to put a tape around the trunk given the felled trees are lying on the ground and are too heavy to lift or rotate. Finally, the reference data of stem volume in this research were derived based on the stem curve, rather than physical measurements of true stem volume. Therefore, this could introduce uncertainty. Consequently, our results should be interpreted as a comparison between LiDAR derived and field data derived estimations, rather than a strict validation against a physical ground truth (e.g., volume measurement using water displacement).

Future efforts will aim to address these limitations. Additionally, we plan to explore timber yield prediction and more detailed reconstruction of branch shapes within the tree structure.

## 7. Conclusion

We introduced a novel method for automated estimation of individual stem volume, designed for stand-level MLS point clouds collected in natural temperate forests with complex understory structure. MLS has great potential to be used for virtual 3D forest representation, but the automation of individual stem detection, segmentation, and volume estimation is limited by complex understory structure and irregular stem shape. Our method incorporates three key modules that include a deep learning model for understory removal, a parameter-free algorithm BSG for stem detection and segmentation, and an SMP approach for stem

shape reconstruction. In the evaluation of four point cloud datasets from different temperate forests, our approach achieved the best performance compared with other representative approaches in terms of understory removal, stem detection and segmentation, and stem reconstruction. More importantly, reconstructed stem models obtained the closest estimation values compared to the destructive field measurement. We believe our method can provide a useful framework for analyzing large-scale MLS point clouds, thereby advancing intelligent 3D forest analytics.

## CRediT authorship contribution statement

**Jinyuan Shao:** Writing – review & editing, Writing – original draft, Visualization, Validation, Project administration, Methodology, Data curation, Conceptualization. **Dennis Heejoon Choi:** Writing – review & editing, Writing – original draft, Validation. **Jidong Liu:** Writing – review & editing, Writing – original draft, Validation. **Xiangxi Tian:** Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Bina Thapa:** Writing – review & editing, Validation. **Seunghyeon Lee:** Writing – review & editing, Validation. **Ayman Habib:** Writing – review & editing, Writing – original draft, Resources. **Songlin Fei:** Writing – review & editing, Writing – original draft, Resources, Project administration, Funding acquisition, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgment

This work was partially supported by the National Institute of Food and Agriculture, United States [grant number 2023-68012-38992].

## Appendix A. Supplementary data

Supplementary data for this article can be found online at doi:10.1016/j.rse.2026.115246.

## References

Abegg, M., Kükenbrink, D., Zell, J., Schaepman, M.E., Morsdorf, F., 2017. Terrestrial laser scanning for forest inventories—tree diameter distribution and scanner location impact on occlusion. Forests 8, 184.

Bauwens, S., Bartholomeus, H., Calders, K., Lejeune, P., 2016. Forest inventory with terrestrial lidar: a comparison of static and hand-held mobile laser scanning. Forests 7, 127.

Béland, M., Widlowski, J.-L., Fournier, R.A., Côté, J.-F., Verstraete, M.M., 2011. Estimating leaf area distribution in savanna trees from terrestrial lidar measurements. Agric. For. Meteorol. 151, 1252–1266.

Bornand, A., Rehush, N., Morsdorf, F., Thürig, E., Abegg, M., 2023. Individual tree volume estimation with terrestrial laser scanning: evaluating reconstructive and allometric approaches. Agric. For. Meteorol. 341, 109654.

Brede, B., Calders, K., Lau, A., Raumonen, P., Bartholomeus, H.M., Herold, M., Kooistra, L., 2019. Non-destructive tree volume estimation through quantitative structure modelling: comparing UAV laser scanning with terrestrial lidar. Remote Sens. Environ. 233, 111355.

Brodu, N., Lague, D., 2012. 3D terrestrial lidar data classification of complex natural scenes using a multi-scale dimensionality criterion: applications in geomorphology. ISPRS J. Photogramm. Remote Sens. 68, 121–134.

Brolly, G., Király, G., Lehtomäki, M., Liang, X., 2021. Voxel-based automatic tree detection and parameter retrieval from terrestrial laser scans for plot-wise forest inventory. Remote Sens. 13, 542.

Burt, A., Disney, M., Calders, K., 2019. Extracting individual trees from lidar point clouds using treeseg. Methods Ecol. Evol. 10, 438–445.

Cabo, C., Ordóñez, C., López-Sánchez, C.A., Armesto, J., 2018. Automatic dendrometry: tree detection, tree height and diameter estimation using terrestrial laser scanning. Int. J. Appl. Earth Obs. Geoinf. 69, 164–174.

Chen, M., Liu, X., Pan, J., Mu, F., Zhao, L., 2023. STEM detection from terrestrial laser scanning data with features selected via STEM-based evaluation. Forests 14, 2035.

Chen, M., Wan, Y., Wang, M., Xu, J., 2018. Automatic STEM detection in terrestrial laser scanning data with distance-adaptive search radius. IEEE Trans. Geosci. Remote Sens. 56, 2968–2979.

Chen, Y., He, T., Huang, D., Ye, W., Chen, S., Tang, J., Chen, X., Cai, Z., Yang, L., Yu, G., et al., Meshanything: Artist-created mesh generation with autoregressive transformers, arXiv preprint arXiv:2406.10163, 2024.

CloudCompare, 2024. Cloudcompare. http://www.cloudcompare.org/.

de Conto, T., Olofsson, K., Görgens, E.B., Rodriguez, L.C.E., Almeida, G., 2017. Performance of STEM denoising and STEM modelling algorithms on single tree point clouds from terrestrial laser scanning. Comput. Electron. Agric. 143, 165–176.

Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M., 2017. Scannet: richly-annotated 3D reconstructions of indoor scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5828–5839.

Dassot, M., Constant, T., Fournier, M., 2011. The use of terrestrial lidar technology in forest science: application fields, benefits and challenges. Ann. For. Sci. 68, 959–974.

Demol, M., Calders, K., Verbeeck, H., Gielen, B., 2021. Forest above-ground volume assessments with terrestrial laser scanning: a ground-truth validation experiment in temperate, managed forests. Ann. Bot. 128, 805–819.

Hackenberg, J., Spiecker, H., Calders, K., Disney, M., Raumonen, P., 2015a. Simpletree—an efficient open source tool to build tree models from TLS clouds. Forests 6, 4245–4294.

Hackenberg, J., Wassenberg, M., Spiecker, H., Sun, D., 2015b. Non destructive method for biomass prediction combining TLS derived tree volume and wood density. Forests 6, 1274–1300.

Hao, Z., Romero, D.W., Lin, T.-Y., Liu, M.-Y., Meshtron: High-fidelity, artist-like 3D mesh generation at scale, arXiv preprint arXiv:2412.09548, 2024.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Heinzel, J., Huber, M.O., 2016. Detecting tree stems from volumetric TLS data in forest environments with rich understory. Remote Sens. 9, 9.

Henning, J.G., Radtke, P.J., 2006. Detailed STEM measurements of standing trees from ground-based scanning lidar. Forest Sci. 52, 67–80.

Henrich, J., van Delden, J., Seidel, D., Kneib, T., Ecker, A.S., 2024. TreeLearn: a deep learning method for segmenting individual trees from ground-based LiDAR forest point clouds . Ecol. Inform. 84, 102888.

Hetti Arachchige, N., 2013. Automatic tree STEM detection–a geometric feature based approach for mls point clouds. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. 2, 109–114.

Hyyppä, E., Hyyppä, J., Hakala, T., Kukko, A., Wulder, M.A., White, J.C., Pyörälä, J., Yu, X., Wang, Y., Virtanen, J.-P., et al., 2020a. Under-canopy UAV laser scanning for accurate forest field measurements. ISPRS J. Photogramm. Remote Sens. 164, 41–60.

Hyyppä, E., Kukko, A., Kaijaluoto, R., White, J.C., Wulder, M.A., Pyörälä, J., Liang, X., Yu, X., Wang, Y., Kaartinen, H., et al., 2020b. Accurate derivation of STEM curve and volume using backpack mobile laser scanning. ISPRS J. Photogramm. Remote Sens. 161, 246–262.

Hyyppä, E., Yu, X., Kaartinen, H., Hakala, T., Kukko, A., Vastaranta, M., Hyyppä, J., 2020c. Comparison of backpack, handheld, under-canopy UAV, and above-canopy UAV laser scanning for field reference data collection in boreal forests. Remote Sens. 12, 3327.

Hyyppä, J., Yu, X., Hakala, T., Kaartinen, H., Kukko, A., Hyyti, H., Muhojoki, J., Hyyppä, E., 2021. Under-canopy UAV laser scanning providing canopy height and STEM volume accurately. Forests 12, 856.

Jiang, L., Zhao, H., Shi, S., Liu, S., Fu, C.-W., Jia, J., 2020. Pointgroup: dual-set point grouping for 3D instance segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4867–4876.

Kükenbrink, D., Marty, M., Bösch, R., Ginzler, C., 2022. Benchmarking laser scanning and terrestrial photogrammetry to extract forest inventory parameters in a complex temperate forest. Int. J. Appl. Earth Obs. Geoinf. 113, 102999.

Kükenbrink, D., Marty, M., Rehush, N., Abegg, M., Ginzler, C., 2025. Evaluating the potential of handheld mobile laser scanning for an operational inclusion in a national forest inventory–a Swiss case study. Remote Sens. Environ. 321, 114685.

Liang, X., Hyyppä, J., Kaartinen, H., Lehtomäki, M., Pyörälä, J., Pfeifer, N., Holopainen, M., Brolly, G., Francesco, P., Hackenberg, J., et al., 2018a. International benchmarking of terrestrial laser scanning approaches for forest inventories. ISPRS J. Photogramm. Remote Sens. 144, 137–179.

Liang, X., Kankare, V., Hyyppä, J., Wang, Y., Kukko, A., Haggrén, H., Yu, X., Kaartinen, H., Jaakkola, A., Guan, F., et al., 2016. Terrestrial laser scanning in forest inventories. ISPRS J. Photogramm. Remote Sens. 115, 63–77.

Liang, X., Kankare, V., Yu, X., Hyyppä, J., Holopainen, M., 2013. Automated STEM curve measurement using terrestrial laser scanning. IEEE Trans. Geosci. Remote Sens. 52, 1739–1748.

Liang, X., Kukko, A., Hyyppä, J., Lehtomäki, M., Pyörälä, J., Yu, X., Kaartinen, H., Jaakkola, A., Wang, Y., 2018b. In-situ measurements from mobile platforms: an emerging approach to address the old challenges associated with forest inventories. ISPRS J. Photogramm. Remote Sens. 143, 97–107.

Liang, X., Litkey, P., Hyyppa, J., Kaartinen, H., Vastaranta, M., Holopainen, M., 2011. Automatic STEM mapping using single-scan terrestrial laser scanning. IEEE Trans. Geosci. Remote Sens. 50, 661–670.

Lin, Y.-C., Liu, J., Fei, S., Habib, A., 2021. Leaf-off and leaf-on UAV lidar surveys for single-tree inventory in forest plantations. Drones 5, 115.

Lin, Y.-C., Shao, J., Shin, S.-Y., Saka, Z., Joseph, M., Manish, R., Fei, S., Habib, A., 2022. Comparative analysis of multi-platform, multi-resolution, multi-temporal lidar data for forest inventory. Remote Sens. 14, 649.

Loshchilov, I., Decoupled weight decay regularization, arXiv preprint arXiv:1711.05101, 2017.

Maas, H.-G., Bienert, A., Scheller, S., Keane, E., 2008. Automatic forest inventory parameter determination from terrestrial laser scanner data. Int. J. Remote Sens. 29, 1579–1593.

Mao, A., Mohri, M., Zhong, Y., 2023. Cross-entropy loss functions: theoretical analysis and applications. In: International Conference on Machine Learning. PMLR, pp. 23803–23828.
Masuda, H., Hiraoka, Y., Saito, K., Eto, S., Matsushita, M., Takahashi, M., 2021. Efficient calculation method for tree STEM traits from large-scale point clouds of forest stands. Remote Sens. 13, 2476.
Moorthy, S.M.K., Bao, Y., Calders, K., Schnitzer, S.A., Verbeeck, H., 2019. Semi-automatic extraction of liana stems from terrestrial lidar point clouds of tropical rainforests. ISPRS J. Photogramm. Remote Sens. 154, 114–126.
Muhojoki, J., Tavi, D., Hyyppä, E., Lehtomäki, M., Faitli, T., Kaartinen, H., Kukko, A., Hakala, T., Hyyppä, J., 2024. Benchmarking under-and above-canopy laser scanning solutions for deriving STEM curve and volume in easy and difficult boreal forest conditions. Remote Sens. 16, 1721.
Neudam, L.C., Fuchs, J.M., Mjema, E., Johannmeier, A., Ammer, C., Annighöfer, P., Paul, C., Seidel, D., 2023. Simulation of silvicultural treatments based on real 3D forest data from mobile laser scanning point clouds. Trees For. People 11, 100372.
Newnham, G.J., Armston, J.D., Calders, K., Disney, M.I., Lovell, J.L., Schaaf, C.B., Strahler, A.H., Danson, F.M., 2015. Terrestrial laser scanning for plot-scale forest measurement. Curr. For. Rep. 1, 239–251.
de Paula Pires, R., Olofsson, K., Persson, H.J., Lindberg, E., Holmgren, J., 2022. Individual tree detection and estimation of STEM attributes with mobile laser scanning along boreal forest roads. ISPRS J. Photogramm. Remote Sens. 187, 211–224.
Pitkänen, T.P., Raumonen, P., Kangas, A., 2019. Measuring STEM diameters with TLS in boreal forests by complementary fitting procedure. ISPRS J. Photogramm. Remote Sens. 147, 294–306.
Pitkänen, T.P., Raumonen, P., Liang, X., Lehtomäki, M., Kangas, A., 2021. Improving tls-based STEM volume estimates by field measurements. Comput. Electron. Agric. 180, 105882.
Pueschel, P., Newnham, G., Rock, G., Udelhoven, T., Werner, W., Hill, J., 2013. The influence of scan mode and circle fitting on tree STEM detection, STEM diameter and volume extraction from terrestrial laser scans. ISPRS J. Photogramm. Remote Sens. 77, 44–56.
Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., Lewis, P., 2013. Fast automatic precision tree models from terrestrial laser scanner data. Remote Sens. 5, 491–520.
Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. Springer, pp. 234–241.
Rusu, R.B., 2010. Semantic 3D object maps for everyday manipulation in human living environments. KI-Künstl. Intell. 24, 345–348.
Ryding, J., Williams, E., Smith, M.J., Eichhorn, M.P., 2015. Assessing handheld mobile laser scanners for forest surveys. Remote Sens. 7, 1095–1111.
Saarinen, N., Kankare, V., Pyörälä, J., Yrttimaa, T., Liang, X., Wulder, M.A., Holopainen, M., Hyyppä, J., Vastaranta, M., 2019. Assessing the effects of sample size on parametrizing a taper curve equation and the resultant stem-volume estimates. Forests 10, 848.
Saarinen, N., Kankare, V., Vastaranta, M., Luoma, V., Pyörälä, J., Tanhuanpää, T., Liang, X., Kaartinen, H., Kukko, A., Jaakkola, A., et al., 2017. Feasibility of terrestrial laser scanning for collecting STEM volume information from single trees. ISPRS J. Photogramm. Remote Sens. 123, 140–158.
Shao, J., Cheng, Y.-T., Koshan, Y., Manish, R., Habib, A., Fei, S., 2023. Radiometric and geometric approach for major woody parts segmentation in forest lidar point clouds. In: IGARSS 2023-2023 IEEE International Geoscience and Remote Sensing Symposium. IEEE, pp. 6220–6223.
Shao, J., Lin, Y.-C., Wingren, C., Shin, S.-Y., Fei, W., Carpenter, J., Habib, A., Fei, S., 2024. Large-scale inventory in natural forests with mobile lidar point clouds. Sci. Remote Sens. 10, 100168.
Sorokina, H.E., Campos, M., Raumonen, P., Shcherbacheva, A., Echriti, R., Hyyppä, J., Puttonen, E., Wang, Y., 2026. Boundary conditions for studying branch-scale tree growth strategies using tree quantitative structure model time series. Remote Sens. Environ. 333, 115105.
Spconv, 2022. Spconv: spatially sparse convolution library. https://github.com/traveller59/spconv.
Tan, K., Zhang, W., Dong, Z., Cheng, X., Cheng, X., 2020. Leaf and wood separation for individual trees using the intensity and density data of terrestrial laser scanners. IEEE Trans. Geosci. Remote Sens. 59, 7038–7050.
Gonzalez de Tanago, J., Lau, A., Bartholomeus, H., Herold, M., Avitabile, V., Raumonen, P., Martius, C., Goodman, R.C., Disney, M., Manuri, S., et al., 2018. Estimation of above-ground biomass of large tropical trees with terrestrial lidar. Methods Ecol. Evol. 9, 223–234.
Tanentzap, A.J., Mountford, E.P., Cooke, A.S., Coomes, D.A., 2012. The more stems the merrier: advantages of multi-stemmed architecture for the demography of understorey trees in a temperate broadleaf woodland. J. Ecol. 100, 171–183.
Tansey, K., Selmes, N., Anstee, A., Tate, N.J., Denniss, A., 2009. Estimating tree and stand variables in a corsican pine woodland from terrestrial laser scanner data. Int. J. Remote Sens. 30, 5195–5209.
Tao, S., Labrière, N., Calders, K., Fischer, F.J., Rau, E.-P., Plaisance, L., Chave, J., 2021. Mapping tropical forest trees across large areas with lightweight cost-effective terrestrial laser scanning. Ann. For. Sci. 78, 103.
Thies, M., Pfeifer, N., Winterhalder, D., Gorte, B.G.H., 2004. Three-dimensional reconstruction of stems for assessment of taper, sweep and lean based on laser scanning of standing trees. Scand. J. For. Res. 19, 571–581.
Vandendaele, B., Martin-Ducup, O., Fournier, R.A., Pelletier, G., 2024. Evaluation of mobile laser scanning acquisition scenarios for automated wood volume estimation in a temperate hardwood forest using quantitative structural models. Can. J. For. Res. 54 (7), 774–792.
Vandendaele, B., Martin-Ducup, O., Fournier, R.A., Pelletier, G., Lejeune, P., 2022. Mobile laser scanning for estimating tree structural attributes in a temperate hardwood forest. Remote Sens. 14, 4522.
Vorster, A.G., Evangelista, P.H., Stovall, A.E.L., Ex, S., 2020. Variability and uncertainty in forest biomass estimates from the tree to landscape scale: the role of allometric equations. Carbon Bal. Manag. 15, 1–20.
Vu, T., Kim, K., Luu, T.M., Nguyen, T., Yoo, C.D., 2022. Softgroup for 3D instance segmentation on point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2708–2717.
Wang, D., Hollaus, M., Puttonen, E., Pfeifer, N., 2016. Automatic and self-adaptive STEM reconstruction in landslide-affected forests. Remote Sens. 8, 974.
West, P.W., 2015. Tree and Forest Measurement, third ed. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-319-14708-6
Wu, X., Wen, X., Liu, X., Zhao, H., 2023. Masked scene contrast: a scalable framework for unsupervised 3D representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9415–9424.
Xia, S., Chen, D., Peethambaran, J., Wang, P., Xu, S., 2021. Point cloud inversion: a novel approach for the localization of trees in forests from TLS data. Remote Sens. 13, 338.
Xie, S., Gu, J., Guo, D., Qi, C.R., Guibas, L., Litany, O., 2020. Pointcontrast: unsupervised pre-training for 3D point cloud understanding. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16. Springer, pp. 574–591.
Yin, D., Wang, L., 2016. How to assess the accuracy of the individual tree-based forest inventory derived from remotely sensed data: a review. Int. J. Remote Sens. 37, 4521–4553.
Yrttimaa, T., Saarinen, N., Kankare, V., Hynynen, J., Huuskonen, S., Holopainen, M., Hyyppä, J., Vastaranta, M., 2020. Performance of terrestrial laser scanning to characterize managed Scots pine (pinus sylvestris l.) stands is dependent on forest structural variation. ISPRS J. Photogramm. Remote Sens. 168, 277–287.
Yu, X., Liang, X., Hyyppä, J., Kankare, V., Vastaranta, M., Holopainen, M., 2013. STEM biomass estimation based on STEM reconstruction from terrestrial laser scanning point clouds. Remote Sens. Lett. 4, 344–353.
Zhang, J., Wang, J., Dong, P., Ma, W., Liu, Y., Liu, Q., Zhang, Z., 2022. Tree STEM extraction from TLS point-cloud data of natural forests based on geometric features and dbscan. Geocarto Int. 37, 10392–10406.
Zhang, W., Qi, J., Wan, P., Wang, H., Xie, D., Wang, X., Yan, G., 2016. An easy-to-use airborne lidar data filtering method based on cloth simulation. Remote Sensing 8, 501.
Zhang, W., Wan, P., Wang, T., Cai, S., Chen, Y., Jin, X., Yan, G., 2019. A novel approach for the detection of standing tree stems from plot-level terrestrial laser scanning data. Remote Sens. 11, 211.
Zhong, L., Cheng, L., Xu, H., Wu, Y., Chen, Y., Li, M., 2016. Segmentation of individual trees from TLS and Mls data. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 10, 774–787.
Zong, C., Xu, J., Song, J., Chen, S., Xin, S., Wang, W., Tu, C., 2023. P2m: a fast solver for querying distance from point to mesh surface. ACM Trans. Graph. 42, 1–13.