

THREE-DIMENSIONAL MODELING OF SHRUBS BASED ON LIDAR POINT CLOUDS

Z. Li^a, M. Li^{a,*}, M. Zhang^b

^a College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, zheng_li@nuaa.edu.cn,
minglei_li@nuaa.edu.cn

^b Co-Innovation Center for Sustainable Forestry in Southern China, College of Biology and the Environment, Nanjing Forestry
University, zhangmeng@njf.edu.cn

KEY WORDS: Point cloud, Point segmentation, Skeleton extraction, Shrub model.

ABSTRACT:

This paper proposes a method for constructing 3D models of shrubs with high accuracy from 3D point clouds acquired by terrestrial laser scanning (TLS). Since the shrub point cloud obtained by LiDAR scanning contains a large amount of redundant data, the focus of this method is to segment the branches and leaves of the shrub point cloud first, which can remove the noise and make the branch skeleton of the shrub stand out. Secondly, a triangulation network is constructed for the segmented branch points, and a minimum spanning tree (MST) is established from the triangulation network as the initial shrub skeleton of the input point cloud. Then, the redundant branches are removed by merging adjacent points and edges to simplify the initial skeleton. Finally, the 3D model of a shrub is constructed by a cylindrical fitting algorithm based on robust principal component analysis (RPCA). Experiments on different types of shrubs from different data sources demonstrate the effectiveness and robustness of the proposed method. The 3D models of shrubs can be further applied to the accurate estimation of shrub attributes and urban landscape visualization.

1. INTRODUCTION AND RELATED WORK

Shrubs usually refer to those trees that have no obvious trunks and are in a cluster state. It plays an important role in windbreak and sand fixation, urban greening, and the improvement of air quality (Davis et al., 2001; Richardson et al., 2011; Kangas and Maltamo, 2006). Constructing digital 3D models of shrubs has a wide range of applications in urban landscape design, ecological simulation, forestry management, virtual entertainment, and so on. In addition, many other applications related to ecological modeling and agroforestry management require accurate estimation of shrub parameters such as height, width, and stem diameter (Ke and Quackenbush, 2011).

The traditional method of measuring shrubs is manual field work, which is usually expensive and time-consuming (Hyypä et al., 2001). In recent years, Light Detection and Ranging (LiDAR) technology has been widely used in forestry-related analysis and research. Because LiDAR scanning can directly capture the 3D point clouds of the scenes and obtain the accurate geometric details of objects, it is widely used in tree height estimation, crown analysis, tree species classification, and many other applications (Liang et al., 2016; Aschoff and Spiecker, 2004; Astrup et al., 2014; Béland et al., 2011). In addition, we can obtain high-density point clouds by applying LiDAR technology, which provides a data basis for accurate 3D modeling of shrubs.

In order to construct accurate 3D models of shrubs, it is necessary to recover the branch shapes of shrubs. Due to the geometric characteristics of shrub branches, the common method to obtain the branch geometry is cylindrical fitting (Liu et al., 2020). Therefore, the critical task of building the 3D models of shrubs is to obtain the topology of branches.

To build the 3D model for a tree, Fu et al. (2020) first extract the initial skeleton by the octree and the level set method, then the

error position is corrected by an optimization with cylindrical prior constraint (CPC). This method has a good effect on trees, but it has the problem of skeleton distortion for shrubs. The L1 median method is widely used to extract the local center of the point cloud, and the skeleton of the point cloud can be obtained by connecting the local center points (Huang et al., 2013). However, the algorithm also has problems such as poor repeatability caused by random sampling, easy loss of details in the case of uneven density sampling, and wrong skeleton connection caused by threshold-based skeleton elongation. Lu et al. (2022) proposed an improved algorithm based on adaptive k -means clustering to deal with the drawbacks of the L1 median skeleton extraction algorithm (Lu and Fan, 2022). It effectively improves the accuracy and repeatability of the point cloud skeleton, and can achieve better extraction results. However, in the data of shrubs with poor quality, the skeleton extracted by the L1 median will have problems such as position error and skeleton crossover. Gaillard et al. (2020) generated a 3D approximate skeleton of a plant through a voxel carving algorithm, and then transformed the skeleton into a mathematical tree by comparing and evaluating the path from each leaf or stem tip to the root of the plant. Finally, they used biologically inspired features to obtain the corresponding skeleton of the input plant. This method has a good effect on the extraction of herbaceous plant skeletons, such as corn, sorghum, etc. But there are problems such as skeleton loss when extracting shrub skeletons. Zhou and Toga (1999) proposed a voxelization coding method for the 3D modeling based on the distance field. The calculation of this method is simple and insensitive to the boundary complexity, but the skeleton obtained in the discrete domain is often discontinuous. Tagliasacchi et al. (2009) proposed a method based on the rotational symmetry axis (ROSA) to extract the skeleton. For multi-branch point clouds with large data gaps, the skeleton extracted by this method can still correctly represent the topology of the original model, but the method has a significant computational complexity.

This paper proposes a skeleton-based method to accurately construct 3D models of shrubs from an independent shrub point cloud. In order to solve the problem of redundant data, we divide the shrub point cloud into branch points and leaf points, so that the shrub branch skeleton is highlighted. Then, the minimum spanning tree (MST) algorithm is used to extract the initial tree skeleton. Finally, skeleton simplification and cylinder fitting are used to obtain the 3D models of the shrubs.

2. METHODOLOGY

The input of the proposed method is a TLS point cloud of shrubs.

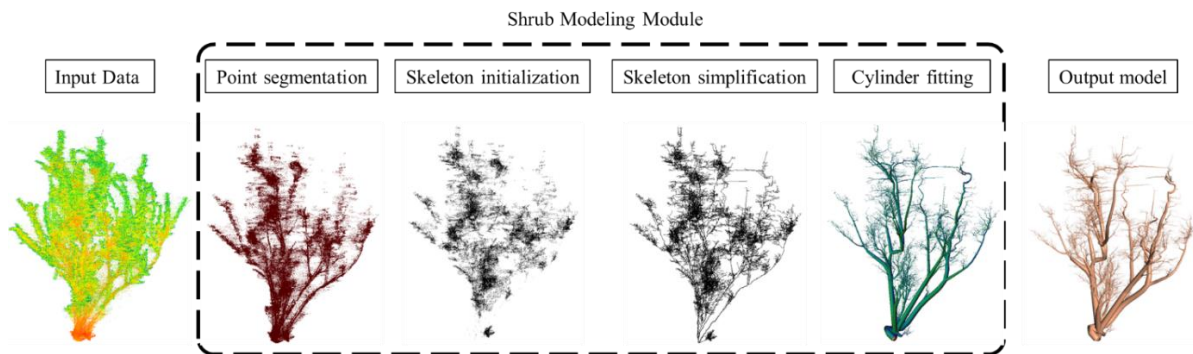


Figure 1. An overview of the proposed methodology.

2.1 Splitting up branch points and leaf points

The shrub point cloud obtained by LiDAR scanning is scattered and contains a large amount of redundant data. Splitting up the branch points from leaves can filter out outliers and eliminate the influence of leaf points, thereby reducing the number of points. Besides, the segmentation process consolidates the branch skeleton of shrubs, which helps to accelerate the calculation speed of subsequent construction models.

2.1.1 Segmentation according to the intensity value: The intensity value obtained by LiDAR scanning is the echo intensity of digital representation, which is proportional to the number of photons hitting the detector (Core and Sterzai, 2006). The intensity values of branch points and leaf points are different due to their different physical reflectivity. In order to adapt to different types of shrub point clouds, we use an adaptive intensity threshold to segment the branch points and leaf points. First, a spatial index K -D tree is established for the input point cloud. Based on the Random Sample Consensus (RANSAC) algorithm, n points are randomly selected as seed points. Second, the local point set is sampled in a spherical space using the seed points as the center and γ as the radius. Then, the density of each sampling point set is calculated. According to the shrub point cloud data used in this paper, n is set to 1000 and γ is set to 0.03.

The entire density interval $[\rho_{min}, \rho_{max}]$ is quartered. The points in the spherical sampling space with densities greater than $\rho_{3/4}$ are defined as branch points, and the points with densities less than $\rho_{1/4}$ are defined as leaf points. The calculation of $\rho_{3/4}$ and $\rho_{1/4}$ is shown in Formula (1).

$$\begin{cases} \rho_{3/4} = \rho_{max} - \frac{\rho_{max} - \rho_{min}}{4} \\ \rho_{1/4} = \rho_{min} + \frac{\rho_{max} - \rho_{min}}{4} \end{cases} \quad (1)$$

The original scanning point cloud contains the main branches and leaf structures of shrubs, but it is usually surrounded by strong noise and a large number of outliers. The output of the method is a 3D model. The first step is to segment the branch points and the leaf points. Secondly, the MST algorithm is used to extract the initial skeleton. The next key step is skeleton simplification. The initial skeleton is simplified by merging adjacent points. Finally, the simplified skeleton is smoothed using the Hermite curve, and the radius of each branch is calculated using the least square method. Figure 1 shows an overview of the module design of the proposed methodology.

According to RANSAC theory, the sample points can approximately represent the intensity distribution of the original point cloud. The intensity values of the branch and leaf points obtained by the statistics are analyzed, and the intensity curves are fitted respectively. The intersection of the curves is used as the intensity threshold I_t to segment the global point cloud, and the initial branch points and leaf points are obtained. As shown in Figure 2, the intersection of the intensity curves of the branch points and the leaf points is used as the intensity threshold I_t .

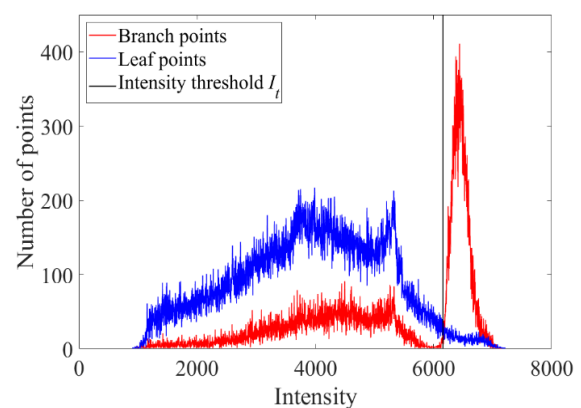


Figure 2. Intensity curves.

2.1.2 Refine the Segmentation based on neighborhoods: Due to the overlap of intensity values, some points will be divided into wrong categories. The neighborhood classification method is used to find the misclassified leaf points in the branch points according to the spatial geometric information, so as to obtain the final branch points. The distribution of branch points is more regular compared with leaf points. Therefore, it can be inferred that the spatial distribution of branch points is also compact and dense, while the misclassified leaf points are sparse and discrete in the initial branch points. First, a K -D tree is constructed for the

initial branch points to facilitate the search of adjacent points. Secondly, according to the formula (2), the average distance d_a between each point p in the initial branch points and its K nearest neighbor points is calculated.

$$d_a = \frac{\sum_{n=1}^K d_n}{K} \quad (2)$$

where d_n is the distance from each adjacent point to the target point. Empirically, the number of neighborhood points K is set to 8. At last, we set a distance threshold T to judge the class of the target points. If the d_a value of the target point is less than T , the point is classified as the final branch point. Otherwise, the point is classified as leaf point.

2.2 Skeleton initialization

In the shrub point cloud, adjacent points are likely to belong to the same branch. Based on this observation, we use an MST algorithm to extract the initial skeleton from the extracted point cloud of branches. Delaunay triangulation is first performed on the point cloud to form a triangulation network (Zhou et al., 2002). Moreover, Delaunay triangulation helps to complete missing regions or incomplete branches, which ensures the robustness of the proposed method to input point clouds with poor data quality. After obtaining the Delaunay triangulation, the length of each edge of a triangle is used as a weight to judge all edges. We use the Dijkstra shortest path algorithm to calculate an MST from the triangulation network as the initial skeleton. Figure 3 shows the initial skeleton extracted from the input points using the shortest path algorithm.

When calculating the MST, the root point of the point cloud needs to be defined first, and the root point is used as the initial point of the MST. Due to the shape characteristics of shrubs, it is hard to distinguish the root point within the point cloud, so the lowest point in the data cannot be directly used as the root point. We assume that the roots of shrubs are located in the bottom center of the entire shrub. Therefore, this paper solves this problem by deliberately selecting the points of the shrub point cloud data center. Firstly, the size of the box in the 3D space of the whole point cloud is calculated. The points in the central part of the box are extracted, and then the lowest point in this part is found as the root point.

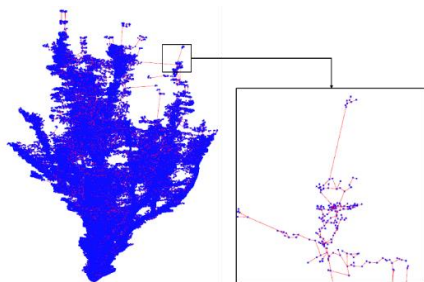


Figure 3. Initial skeleton.

2.3 Skeleton simplification

The initial skeleton has a large number of redundant points and edges. It is possible to delete a large number of redundant edges to further simplify the shrub skeleton without affecting the reconstruction accuracy of the model. Assigning weight values to each point and each edge in the initial skeleton will help to further simplify the skeleton. In this paper, the weight value is assigned to each point according to the length of the subtree. The

subtree of a point is defined as a collection consisting of itself and its offspring points and edges. Therefore, the weight value w_i of a point p_i is calculated as the sum of the lengths of all edges in its subtree. The weight value of each branch is the average of the weights of its two points. In this way, the points and edges of the top area of the shrub have consistent low weight values. The trunk branches near the bottom area of the shrub will obtain larger weight values, while the small branches near the trunk will obtain very small weight values. Such characteristics can help us to remove the small noisy branches lying on the main branches, while retaining the small branches in the upper part of the shrub.

After simplifying the initial skeleton by the weight value, there are still some redundant edges and points. This paper simplifies these edges and points by detecting the proximity between adjacent points. The similarity β is defined to describe the closeness between adjacent points. There are two possible cases where the current point has one child point or multiple child points.

For points with only one child point, the skeleton simplification problem is transformed into a line simplification problem. In this paper, if the distance between the current point and the line segment formed by its parent point and child point is closer, the importance of the current point is lower. Therefore, the similarity β is defined as:

$$\beta = d \left(\frac{w_c}{w_{\max}} \right)^{1.1} \quad (3)$$

where d represents the distance between the current point and line segment formed by its parent point and child point, w_c is the weight of the current point, w_{\max} is the maximum weight of all points. Setting the exponential rate to 1.1 is to adjust the adaptability of the similarity. If the similarity β is less than the given threshold σ , it is considered that the current point is less important and can be deleted from the skeleton.

For points with more than 2 child points, we compare each pair of child points iteratively. Every time, we determine whether a pair of child points need to be merged by a similarity score β . If merging is necessary, we will include the merged point in the set and delete the original two. The similarity β of any two child points is calculated by:

$$\beta = \min \left(l_1 \sin \theta \left(\frac{w_c}{w_{\max}} \right)^{1.1}, l_2 \sin \theta \left(\frac{w_c}{w_{\max}} \right)^{1.1} \right) \quad (4)$$

where l_1 and l_2 represents the lengths between the current point and its two child points, θ is the angle between the two edges of the current point and two child points. It should be noted that the similarity calculated from different directions is different, so we choose the minimum value to judge the closeness between adjacent child points. By calculating the similarity β of each two child points, we can get a set of similarity $\{\beta_i\}$.

If the minimum similarity β_{\min} in the $\{\beta_i\}$ is less than the given threshold σ , the two child points corresponding to β_{\min} are needed to be merged. We use formula (5) to merge these two child points into a new child point to complete the simplification. The merged new child point position is the weighted average of the original two child points.

$$p_{\text{new}} = \frac{p_1 w_1 + p_2 w_2}{w_1 + w_2} \quad (5)$$

where \mathbf{p}_{new} represents the position of the new child point, \mathbf{p}_1 and \mathbf{p}_2 are the positions of the original two child points, and w_1 and w_2 are the weight values of the original two child points, respectively. Figure 4 shows the simplification process.

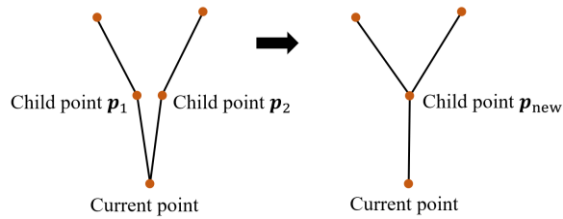


Figure 4. Point simplification by merging similar child points.

2.4 Cylinder fitting

The simplified skeleton can already represent the topological structure of the shrub. On this basis, this paper smooths the shrub skeleton by Hermite curve fitting. The method makes the models of branches more in line with the bending characteristics of shrub branches. Then, the cylinder fitting method is used to accurately simulate the geometry of the branches. In practice, we use two different means to adjust the cylinders for the branches, which are divided into two types, namely trunk branches and branch tips.

If the branches don't contain the leaf points in the MST, these branches are not belonging to branch tips, and they usually have high point density. We use a robust principal component analysis (RPCA)-based method to obtain accurate branch geometries (Nurunnabi et al., 2019). Firstly, the points near each branch are segmented and identified. Then, the cylinder surface of each branch is fitted according to the corresponding branch points to approximate the branch geometry. Figure 5 illustrates the cylinder fitting process.

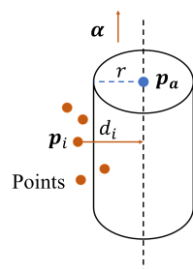


Figure 5. Parameters to be solved in cylindrical fitting.

The input data for the cylinder fitting is a set of coordinates $\{\mathbf{p}_i\}$ of the divided branch points. The point set is selected by calculating the distance to the line segment in the skeleton. We choose the points within a distance of 3.5 times the weight of the line segment as the branch points. The parameters to be solved include the axial direction vector α of the cylinder, the position \mathbf{p}_a of the endpoint on the axis, and the radius r of the cylinder. In this paper, principal component analysis (PCA) is used to calculate the axial direction vector α and the position \mathbf{p}_a of the endpoint on the axis. To solve radius r , we design an objective function, which minimizes the sum of the distance from the branch points to the initial cylinder.

$$r = \operatorname{argmin} \left(\sum_{i=1}^n |d_i - r| \right) \quad (6)$$

where $|d_i - r|$ represents the distance from the point P_i to the cylindrical surface. The Levenberg-Marquardt (LM) algorithm is used to solve the nonlinear least squares problem (Marquardt, 1963). With the help of the LM algorithm, the initial cylinder radius r can be calculated.

In order to further improve the quality of the solution, we introduce weights for each point to adjust r . The weight λ_i of each branch point is defined as follows:

$$\lambda_i = 1 - \frac{|d_i - r|}{d_{\max}} \quad (7)$$

where d_{\max} denotes the maximum distance among all distances between the points of the specific branch and its initial cylinder. By formula (7), the weights of all the points of the branch are normalized. Then, the objective function is updated to:

$$r_{\text{new}} = \operatorname{argmin} \left(\sum_{i=1}^n \lambda_i |d_i - r_{\text{new}}| \right) \quad (8)$$

Once again, we use the LM algorithm to solve formula (8) and get the adjusted r . Using the calculated \mathbf{p}_a , α , and r , we can generate the cylinder surface for each branch.

On the other hand, the points near the top of the shrub are usually noisy, and it is difficult to perform accurate cylindrical fitting. We treat these branches as branch tips and calculate their radius by using the weights of the branches:

$$r_i = \bar{r} \left(\frac{w_i}{\bar{w}} \right) \quad (9)$$

where r_i represents the radius of the i -th branch, and \bar{r} represents the average radius of all trunk branches, which are calculated by (8). w_i represents the weight value of the i -th branch, and \bar{w} represents the average of the weight values of all trunk branches. As described in section 2.3, the weight value of a trunk branch is the average of the point weights of its two end points in the MST.

3. RESULTS

3.1 Experimental results

All point clouds in our experiment are acquired with a TLS device. The scanning equipment is Leica BLK360. The scanning process is performed independently for an individual shrub, and the shrub point cloud data is manually extracted after the scanning. Figure 6 shows the reconstruction results of 2 sets of data. The bottom part of the original point cloud of Shrub 1 is missing, and the points of Shrub 2 have a higher density and noise level. The models reconstructed from these two data sets show that the proposed method can handle shrubs with different shapes and structures.

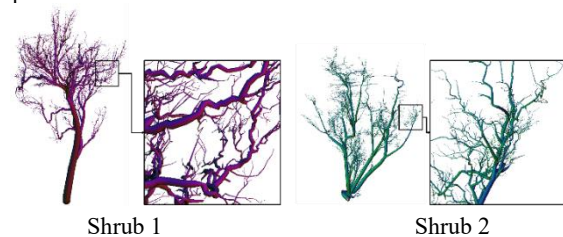


Figure 6. 3D models of shrubs.

In this paper, the accuracy of the modeling results is quantified by calculating the average distance from the input shrub point cloud data to the generated model (P2M distance) (Du et al., 2019). The shortest distance from each point in the original shrub point cloud to the model surface is calculated, and the original point cloud is colored according to the distances, as shown in Figure 7. The closer the color to blue, the shorter the distance; and vice versa. Table 1 lists the statistics corresponding to Figure 7. We can observe that the points located in the main branch are usually closer to the model, and the points near the tip of the branch usually have larger errors. The results demonstrate that the proposed method can generate high-precision main branch structures of shrubs. However, as the points near the tip of the branches become sparser, the point density is not sufficient to reliably reconstruct such branches.

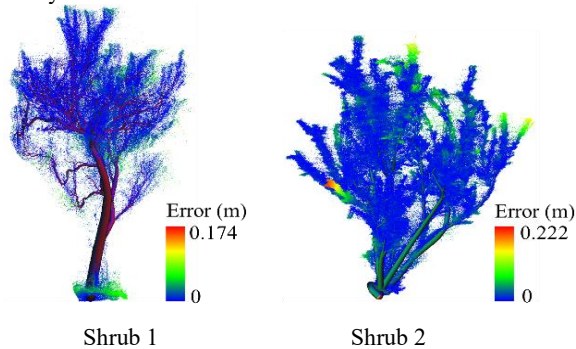


Figure 7. Geometric error heat map.

Number	Total Points	Complexity	P2M distance (cm)
Shrub 1	272925	Easy	0.98
Shrub 2	1012927	Difficult	1.18

Table 1. Shrub data statistics related to Fig.7.

3.2 Parameter tuning

In our methodology, distance threshold T and similarity threshold σ are introduced in point segmentation and skeleton simplification. This section will discuss the influence of different parameter settings on the modeling results. On this basis, we select the threshold that is most suitable for our method.

The distance threshold T controls the degree of outlier filtering in the density-based branch separation. According to the experimental data, we test the threshold T from 0.005 to 0.009, and the results are shown in Figure 8. Through experiments, many branch points are misclassified into leaf points when the threshold is too small, resulting in the incomplete skeleton of the final branch points. When the threshold is too large, the wrong leaf points contained in the initial branch points are not classified. Therefore, we choose 0.007 as the segmentation threshold.

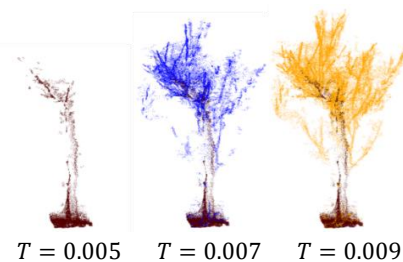


Figure 8. Segmentation results using different segmentation thresholds.

The similarity threshold σ controls the similarity of points in the skeleton simplification process. According to the experimental data, we test the threshold σ from 1.0 to 2.0, and the results are shown in Figure 9. The degree of skeleton merging at the branches is low when the simplified threshold is small, which does not meet the characteristics of branch growth; there will be excessive simplification when the simplification threshold is large. Therefore, through experiments, we choose 1.5 as the simplification threshold.

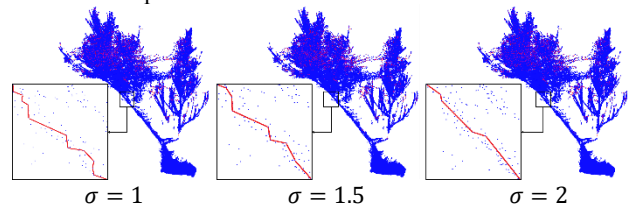


Figure 9. Skeletonization results using different similarity thresholds.

3.3 Comparisons

This section compares the proposed method with two other methods proposed for tree model reconstruction. Figure 10 shows the modeling results compared with *PypeTree* (Delagrangé et al., 2014) and *AdTree* (Du et al., 2019). *PypeTree* is a tool for reconstructing tree model from point clouds that were acquired with TLS. It introduces the idea of using semi-supervised adjustment tools to further improve reconstruction accuracy. *AdTree* is a tool based on extracting skeleton to build a tree model. It has a good modeling effect for trees.

Table 2 lists the accuracy parameters corresponding to Figure 10. It can be seen that the result of *PypeTree* only provides a rough description of the shrub topology, and does not restore the branch geometry. *AdTree* can extract the topological structure and branch geometry of shrubs. However, the modeling results of *AdTree* are very dependent on data quality. For point cloud data with gaps and large amounts of noise, the modeling results are seriously distorted. The model of shrub 2 generated by *AdTree* has higher accuracy because it contains a large number of redundant branches filling the entire data space, as shown in Figure 11. Although the *AdTree* points can get a small P2M distance, each point participating in the calculation did not obtain the true distance. On the contrary, the proposed method can construct models with higher topological and geometric accuracy for shrub data sets.

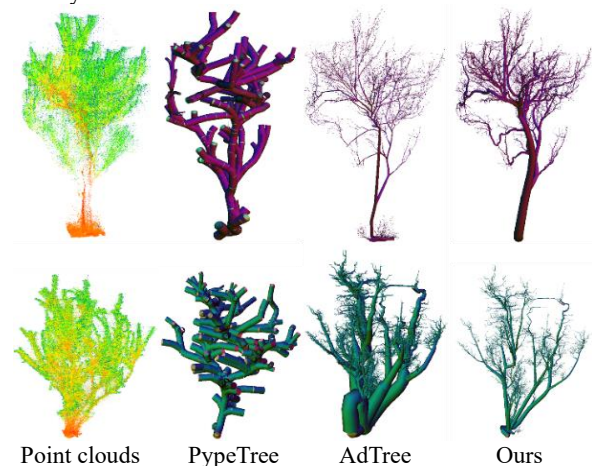


Figure 10. 3D models of shrubs constructed by different methods.

P2M distance (cm)	PypeTree	AdTree	Ours
Shrub 1	1.53	1.13	0.98
Shrub 2	2.32	1.07	1.18

Table 2. The accuracy of shrub models with different methods corresponding to Fig.10.

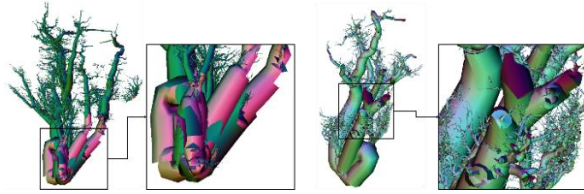


Figure 11. Redundant branches generated by AdTree.

4. CONCLUSION

This paper proposes a method for automatically reconstructing 3D models of shrubs from LiDAR point clouds. A novelty of the method proposed is that the input shrub point cloud is first divided into branch points and leaf points, so as to obtain branch point cloud with obvious branch skeleton, which helps to improve the accuracy of shrub skeleton extraction. In addition, the cylindrical fitting method based on RPCA can better fit the geometric structure of shrub branches. Experiments show that the proposed method is robust when dealing with shrub point clouds of various types and sizes.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (No. 42271343) and the National Natural Science Foundation of China (No. 42177387), and supported by the Fund of State Key Laboratory of Remote Sensing Information and Image Analysis Technology of Beijing Research Institute of Uranium Geology (6142A01210403).

REFERENCES

Aschoff, T., Spiecker, H., 2004: Algorithms for the automatic detection of trees in laser scanner data. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36, 71-74.

Astrup, R., Ducey, M.J., Granhus, A., Ritter, T., von Lüpke, N., 2014: Approaches for estimating stand-level volume using terrestrial laser scanning in a single-scan mode. *Canadian Journal of Forest Research*, 44(6), 666-676.

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. *Agricultural and Forest Meteorology*, 151, 1252-1266.

Core, F., Sterzai, P., 2006: Radiometric correction in laser scanning. *International Journal of Remote Sensing*, 27(15), 3097-3104.

Davis, L.S., Johnson, K.N., Bettinger, P.S., Howard, T.E., 2001: *Forest management: To sustain ecological, economic, and social values*. Boston: McGraw Hill.

Delagrangé, S., Jauvin, C., Rochon, P., 2014: PypeTree: A tool for reconstructing tree perennial tissues from point clouds. *Sensors*, 14, 4271-4289.

Du, S., Lindenbergh, R., Ledoux, H., Stoter, J., Nan, L., 2019: AdTree: Accurate, Detailed, and Automatic Modelling of Laser-Scanned Trees. *Remote sensing*, 11, 2074.

Fu, L., Liu, J., Zhou, J., Zhang, M., Lin, Y., 2020: Tree Skeletonization for Raw Point Cloud Exploiting Cylindrical Shape Prior. *IEEE Access*, 8, 27327-27341.

Gaillard, M., Miao, C., Schnable, J., Benes, B., 2020: Sorghum segmentation by skeleton extraction. *Computer Vision - ECCV 2020 Workshops*, 296-311.

Huang, H., Wu, S., Cohen-Or, D., Gong, M.L., Zhang, H., Li, G., Chen, B., 2013: L1-Medial Skeleton of Point Cloud. *ACM Transactions on Graphics*, 32(4): 1-10.

Hyypä, J., Kelle, O., Lehtikoinen, M., Inkinen, M., 2001: A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners. *IEEE Transactions on Geoscience and Remote Sensing*, 39(5), 969-975.

Kangas, A.S., Maltamo, M., 2006: *Forest Inventory: Methodology and Applications*. Springer.

Ke, Y., Quackenbush, L.J., 2011: A review of methods for automatic individual tree-crown detection and delineation from passive remote sensing. *International Journal of Remote Sensing*, 32(17/18), 4725-4747.

Liang, X., Kankare, V., Hyypä, J., Wang, Y., Kukko, A., Haggrén, H., Yu, X., Kaartinen, H., Jaakkola, A., Guan, F., 2016: Terrestrial laser scanning in forest inventories. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115, 63-77.

Liu, Z., Zhang, Q., Wang, P., Li, Z., Wang, H., 2020: Automated Classification of Stems and Leaves of Potted Plants Based on Point Cloud Data. *Biosystems Engineering*, 200, 215-230.

Lu, B., Fan, X., 2022: Research on 3D Point Cloud Skeleton Extraction Based on Improved Adaptive k-means Clustering. *Acta Automatica Sinica*, 48(8), 1994-2006.

Marquardt, D.W., 1963: An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2), 431-441.

Nurunnabi, A., Sadahiro, Y., Lindenbergh, R., Belton, D., 2019: Robust cylinder fitting in laser scanning point cloud data. *Measurement*, 138, 632-651.

Richardson, D.M., Rejmánek, M., 2011: Trees and shrubs as invasive alien species - a global review. *Diversity and Distributions*, 17, 788-809.

Tagliasacchi, A., Zhang, H., Cohen-Or, D., 2009: Curve skeleton extraction from incomplete point cloud. *ACM Transactions on Graphics*, 28(3), 71.

Zhou, H., Shenoy, N., Nicholls, W., 2002: Efficient minimum spanning tree construction without Delaunay triangulation. *Information Processing Letters*, 81, 271-276.

Zhou, Y., Toga, A.W., 1999: Efficient Skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3), 196-209.