

Separation of Wood and Foliage for Trees From Ground Point Clouds Using a Novel Least-Cost Path Model

Sheng Xu¹, Kai Zhou, Yuan Sun, and Ting Yun¹

Abstract—Nowadays, laser scanning technology has provided an effective and nondestructive approach to reveal the forest’s developmental process and physiological properties. For the purpose of obtaining the 3-D spatial structure and skeleton of trees, this article addresses the separation of wood and foliage from the forest using two phases. The first global phase develops a pointwise supervised learning framework to classify forest point clouds. In order to improve the classification accuracy, we design new features for the learning process, which supplements the current geometric features in terms of the topological information. The second local phase designs a new least-cost path model to further separate wood and foliage points. The separation of branch points is formulated as an energy function and optimized by the dynamic programming technique. Experiments on different plots show that points from stems and branches are detected as wood points completely and correctly. The achieved average completeness, correctness, and F_1 score of the wood and foliage separation are 91.25%, 90.34%, and 0.91, respectively, which is promising to the phenotyping study related to the organism’s physical form and structure.

Index Terms—Least-cost path model, phenotyping study, point clouds, pointwise classification, separation.

I. INTRODUCTION

PHENOTYPING study is significant to the expression of the genetic code, which is related to the organism’s physical form and structure. Different from the biological methods, laser scanning technology can provide an effective, nondestructive, and automatic approach to reveal the forest’s physiological properties by fully scanning 3-D trees.

Nowadays, laser scanner systems are widely used in 3-D vegetation information collection. Airborne laser scanning (ALS) is the main technique to capture and analyze the canopy structure of forest regions, for example, the estimation of the canopy structure and biomass based on the topographic metrics [1].

Manuscript received April 2, 2021; revised May 11, 2021 and June 5, 2021; accepted June 15, 2021. Date of publication June 18, 2021; date of current version July 14, 2021. This work was supported in part by the Natural Science Foundation of Jiangsu Province under Grant BK20200784, in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province under Grant 19KJB520010, and in part by China Postdoctoral Science Foundation under Grant 2019M661852. (*Corresponding author:* Ting Yun.)

Sheng Xu is with the College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China (e-mail: xusheng@njfu.edu.cn).

Kai Zhou, Yuan Sun, and Ting Yun are with the Co-Innovation Center for Sustainable Forestry in Southern China, Nanjing Forestry University, Nanjing 210037, China (e-mail: kaizhou@njfu.edu.cn; yuan.sun@njfu.edu.cn; njyunting@gmail.com).

Digital Object Identifier 10.1109/JSTARS.2021.3090502

Although ALS can retrieve points of a large-scale forest, the collection cost is usually high. A low-cost way to analyze tree crowns is based on the unmanned aerial vehicle point clouds [2], which collects points for accessing tree crowns at close range. Mobile laser scanning (MLS) is widely used in outdoor environment analysis, such as the plane extraction [3] and urban object classification [4]. In the field of the urban forest, researchers put their efforts to study street tree attributes, such as stem extraction and segmentation [5]. Personal laser scanning, which belongs to portable laser scanning systems, e.g., backpack laser scanning systems and handheld laser scanning systems, is flexible in data collection. For example, Xu *et al.* [6] choose the handheld laser scanning system to access more nonphotosynthetic components covered by crowns, including the second and third branching order of trees. TLS is popular in the analysis of forests, which succeeds in detecting geometric characteristics of points to identify objects [7], such as stems [8], trees [9], crowns [10], and foliage [11].

In the tree 3-D structure analysis, it is not effective to obtain plenty of nonphotosynthetic components from the forest by ALS and MLS, because branches are easily covered by foliage. By optimizing the plot size and scanner position, TLS has more probability to obtain abundant nonphotosynthetic components from crowns. Therefore, we mainly choose TLS point clouds for the wood and foliage separation in this study. Due to the fact that forest collected by TLS is usually in huge volumes of unorganized point clouds, with no topological information, and containing a certain number of outliers, noise, and overlapping points in multiple scans, the processing methods designed for ALS and MLS are often less effective in TLS. Although the combination of different platforms [12] can improve the distinguishment accuracy of point clouds, the registration of different scanner points is still a challenging task. In order to obtain the spatial structure of trees from TLS point clouds, one has to extract and separate wood and foliage from forests, which is the target of this study. Our two main contributions are 1) the calculation of topological information to supplement geometric features for classifying TLS point clouds and 2) the formulation and optimization of a least-cost path model to track branches covered by crowns.

It is worth noting that there is no existing point cloud research related to the feature calculation of surface intersection for topological information. The topological feature, which plays an important role in trunk detection as a supplement for

geometric features, is difficult to obtain from unorganized point clouds. Besides, the existing wood points detection depends on the completeness of branch points and fails to connect break branches caused by the occlusion and incompleteness, which will be solved by the proposed least-cost path model based on a branch tracking strategy.

II. RELATED WORK

Nowadays, there are lots of methods have been proposed for achieving tree structure information from TLS point clouds, e.g., the diameter at height, the under branch height, and the leaf area index. The existing methods include model-based methods, feature-based methods, and deep-learning-based methods, which will be analyzed and discussed in this section.

The model-based methods are often used for individual tree and stem extraction. With regard to the pole detection, Liang *et al.* [13] use the spatial distribution properties of points to estimate the location of stems. Their cylinder fitting works well at wood points that are from vertically standing stems, but the accuracy is decreased when stems are incomplete or occluded. Xia *et al.* [8] focus on stem detection by identifying linear structures and merging disconnected stem parts using a direction-growing algorithm. Their method does not require information about the terrain surface, but needs prior knowledge of targets for the curve model calculation. Lau *et al.* [14] investigate the branch architecture by proposing a 3-D quantitative structure model. Their model chooses the cylinder as the primitive to fit wood points in the first, second, and third branching orders, when branches are complete and fewer outliers in the primitive fitting. Zhang *et al.* [15] separate stems from the other points by calculating and modeling the normal change rate. Their method works well for those complete stems, but since the curvature of occluded regions is undefined, the extraction of discontinued stems is tackle to detect. The model-based methods work well and are easy to implement in straight trunks, because of their similar point distribution. However, those segmentation methods are easy to fail in data with occlusion and incompleteness.

The feature-based methods are often used for the wood and foliage classification. Tao *et al.* [16] introduce a geometric feature method for wood–leaf separation, e.g., circlelike shapes and line segments. The drawback lies in that their constant search radius is difficult for sparse points with little shape information. Ma *et al.* [17] propose more features to contain structural information of foliage elements based on the spatial distribution patterns and use the Gaussian mixture model to classify points. Although there is no need to prepare training samples, the point distribution of different trees is various in forest environments. Soon afterward, scholars believe that more features will bring high accuracy, Wang *et al.* [18] show that the classical machine learning methods can classify wood and foliage by using abundant geometric information. However, the feature extraction of points is sensitive to noise and density. Chen *et al.* [19] propose a point-based method for stem detection using single-scan TLS data. They design an adaptive radius for searching the eigenvalue features of points, and classify stem points by the support vector machine (SVM) directly. Although their method is easy to implement, they need to calibrate data to ensure

that the reflectance of the tree leaves and branches is smaller than that of stems. Instead of the pointwise classification, Koma *et al.* [20] conduct the region growing to obtain leaf-morphological features for the classification and obtain a high accuracy in the fixed environment by tuning optimal parameters. Li *et al.* [21] concern about the classification by the gap probabilities features resulting from leaves and woody materials. Although theoretically, the laser reflectance of wood hit and leaf hit is different, the accuracy is difficult to improve in forests due to the laser range limitations. The feature-based methods are more robust in occlusion and incompleteness. However, 3-D feature extraction in point clouds is difficult, because geometric features often fail in splitting overlapping objects.

Nowadays, more and more researchers move to deep-learning-based methods for point cloud processing. In the work of [22], they propose a voxel-based deep learning method to classify tree species. They first extract individual trees based on the density of points and, then, perform a voxel-based rasterization process to represent low-level features by the convolutional neural network (CNN) model. In the work of [23], they use the CNN model to extract the high-level representation of features. The key idea is to calculate the local geometric features, global geometric features, and full-waveform features of neighboring laser points and, then, transform those features into 2-D features as the input of a CNN model for a 3-D labeling task. In the work of [24], they propose a deep 3-D fully convolution network (FCN) to filter both stem and branch points. To train the 3-D FCN, reference stem and branch points are delineated semiautomatically. They achieve high overall accuracy (OA) of 0.94 for the stem and branch. Since their features derived from the 3-D covariance matrix are computed using the surrounding neighborhood of points, they can only capture local information resulting in requiring the application at multiple scales to describe the diversity of objects. In the work of [25], they develop a deep supervised machine learning framework for tree detection, segmentation, and stem reconstruction using the fully convolutional encoder–decoder neural network model. Their method adapts to various densities and partial occlusions in point clouds. Although deep learning-based methods achieve a high performance in the point cloud classification, the supervised learning method needs to manually segment a large number of trees for setting the training set, which is often tedious, redundant in the complex outdoor scene. Besides, they usually require a high-performance graphics processing unit for accelerating algorithms.

The wood and foliage unsupervised separation methods are in a pipeline of the following three steps:

- 1) feature extraction of point clouds. The most frequently used features are based on geometric primitives, including shape primitives, e.g., lines, surfaces and volumetric shapes, and structure primitives, represented by skeletons and edges [26];
- 2) classification of input points into different regions using supervised or unsupervised methods; and
- 3) refinement of results by removing false points/clusters/components/instances from the achieved positive groups.

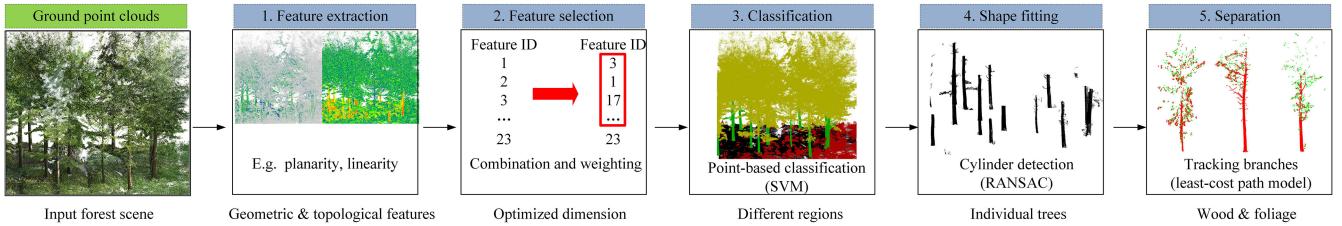


Fig. 1. Flowchart of the proposed wood and foliage separation, including feature extraction, feature selection, classification, shape fitting, and separation.

Although the aforementioned methods benefit the tree structure extraction a lot, there are still two unsettled issues in the structure extraction, including 1) how to calculate and select features of points for the high-performance pointwise classification and 2) how to separate discontinued branches, when there is little neighbor information.

This article targets the separation of wood and foliage from TLS point clouds for delineating the tree structure with the coordinate information only. In order to overcome limitations in tree structure extraction, this study proposes two phases of optimization for improving the wood and foliage separation. The first global phase intends to use a supervised learning method for classifying point clouds into four regions, including 1) crowns, 2) stems, 3) bushes, and 4) ground points. Extracted features contain both geometric and topological information. In order to improve the accuracy and efficiency, only high-performance features will be selected for the pointwise classification. The second local phase aims to fine separate wood from foliage covered by tree crowns based on our proposed least-cost path model. The energy function for building candidate paths is constrained by a data term to collect the regional branch points and a smoothness term to track the overall branch information. The path penalty is minimized by the dynamic programming technique globally.

III. METHOD

As shown in Fig. 1, our flowchart mainly contains five stages. Stage 1 is to do the feature extraction. Except for geometric features based on the eigenvalue of points, we calculate topological features. Stage 2 is to deal with feature selection, which aims to optimize the dimension of features. Stage 3 is to classify input points into different regions based on the selected features. Stage 4 is to do a simple cylinder fitting to segment individual trees using RANSAC (i.e., Random Sample Consensus). Stage 5 is to separate branches and leaves based on our proposed least-cost path model. Two key steps are the first and last stages, and the expected output is the separation of wood and foliage points.

A. Feature Extraction

The most commonly used features for the pointwise classification are based on geometric information as listed in [27], such as the sum of eigenvalues, planarity, and linearity. Those features are calculated by the combination of eigenvalues and aim to capture the shape clues of objects. However, to the best of the authors' knowledge, there is no research to calculate topological

features for the point cloud classification, which plays an important role in capturing global information of targets. Since point clouds are unorganized and containing limited information, i.e., the coordinate of x, y, and z only, the calculation of topological information is a challenging task.

Our idea is based on the knowledge that for each nonisolated 3-D point, there can be only two main conditions. 1) The point lies on a surface. 2) The point lies in the intersection of planes, i.e., contours or corners. The rest of the points are regarded as scatter points, for example, points from foliage regions. Compared with surface points and intersection points, scatter points are in a much less scale. Besides, points from surfaces, contours, and corners contain branching order information for the tree structure. Therefore, we focus on the feature extraction of points from the aforementioned two conditions. Inspired by the detection of line intersection in 2-D images [28], we know that the intersection of lines can be detected based on the gray intensity difference. Extend this conclusion to point clouds, we know that 1) it is required to define the gradient of point clouds based on the difference of neighbor points, and 2) if there is no intersection of planes around a nonisolated point p , p is on a plane; if p is from the intersection of planes, p lies in the contours or corners.

Gradient calculation. In the 2-D image, gray is used to show the brightness level of a pixel. However, our input data are limited to the coordinate information only. We try to use the number of points in a voxel to simulate the local density of points. The gradient G is the difference of the density I as shown in

$$G(x, y, z) = \sum_{\Delta x, \Delta y, \Delta z} e^{\frac{(\Delta x^2 + \Delta y^2 + \Delta z^2)}{2\sigma^2}} [I_{x+\Delta x, y+\Delta y, z+\Delta z} - I_{x, y, z}]^2 \quad (1)$$

where I is the number of points in a voxel at (x, y, z) , $(\Delta x, \Delta y, \Delta z)$ is the step size between two voxels, and σ^2 is the density variance of neighbor voxels. Based on the Taylor expansion at the first-order term, we have

$$G(x, y, z) = (\Delta x, \Delta y, \Delta z) \mathbf{M} (\Delta x, \Delta y, \Delta z)^\top \quad (2)$$

where the 3×3 symmetric matrix \mathbf{M} is

$$\mathbf{M} = \begin{bmatrix} A & A \cdot B & A \cdot C \\ A \cdot B & B & B \cdot C \\ A \cdot C & B \cdot C & C \end{bmatrix} \quad (3)$$

$$A = \left(\frac{\partial I}{\partial x} \right)^2 e^{\frac{(\Delta x^2 + \Delta y^2 + \Delta z^2)}{2\sigma^2}} \quad (4)$$

$$B = \left(\frac{\partial I}{\partial y} \right)^2 e^{\frac{(\Delta x^2 + \Delta y^2 + \Delta z^2)}{2\sigma^2}} \quad (5)$$

$$C = \left(\frac{\partial I}{\partial z} \right)^2 e^{\frac{(\Delta x^2 + \Delta y^2 + \Delta z^2)}{2\sigma^2}}. \quad (6)$$

M is a semipositive symmetric matrix, and $e^{\frac{(\Delta x^2 + \Delta y^2 + \Delta z^2)}{2\sigma^2}}$ is a window function to weight voxels. Its eigenvectors α , β , and γ are mutually orthogonal and no less than 0. G is related to the local autocorrelation function and M describes the shape at the origin [28]. M can be regarded as a rotationally invariant descriptor, which is suitable for extracting information from the unorganized point cloud.

Intersection detection. For a point p , if $\alpha \gg \beta$, and $\alpha \gg \gamma$, p tends to be on a certain plane. If $\alpha \gg 0$, $\beta \gg 0$, and $\gamma \gg 0$, p tends to be on an intersection of planes. Those can be indicated by the trace $\text{Tr}(M)$ and determinant $\text{Det}(M)$, which is the sum and the joint product of eigenvalues as calculated in (7) and (8), respectively

$$\text{Tr}(M) = A + B + C = \alpha + \beta + \gamma \quad (7)$$

$$\begin{aligned} \text{Det}(M) &= A \cdot B \cdot C + (A \cdot B) \cdot (B \cdot C) \cdot (A \cdot C) \\ &\quad + (A \cdot B) \cdot (B \cdot C) \cdot (A \cdot C) \\ &\quad - (A \cdot C) \cdot B \cdot (A \cdot C) - A \cdot (B \cdot C) \cdot (B \cdot C) \\ &\quad - (A \cdot B) \cdot (A \cdot B) \cdot C = \alpha \cdot \beta \cdot \gamma. \end{aligned} \quad (8)$$

The detection of the intersection of only two planes seems a bit complex, which requires that $\alpha \approx \beta$, and $\beta \gg \gamma$. Our detection is based on the work of [29], which aims to find the intersection of two planes for the curb extraction. The operator is shown in (9). If the result E of p is large, p tends to be from the intersection of two planes. It is worth noting that $A^2 \cdot B^2 - (A \cdot B) \cdot (A \cdot B)$ is not zero, because the derivation process is in discrete form and has been smoothed by a Gaussian filter. In the intersection detection, the derivation of the density in (6) is based on a simple detector calculated as the combination of $(-1 \ 0 \ 1)$ in the x, y, z direction as Sobel operator in the 2-D image. We do not need to calculate eigenvalues of M , i.e., α , β , and γ , which makes the intersection detection easy to implement

$$\begin{aligned} E &= \frac{A^2 \cdot B^2 - (A \cdot B) \cdot (A \cdot B)}{A + B} (A + B + C) \\ &\quad + \frac{A^2 \cdot C^2 - (A \cdot C) \cdot (A \cdot C)}{A + C} (A + B + C) \\ &\quad + \frac{B^2 \cdot C^2 - (B \cdot C) \cdot (B \cdot C)}{B + C} (A + B + C). \end{aligned} \quad (9)$$

To better understand the proposed gradient-based features, we visualize the value of $\text{Tr}(M)$, E , and $\text{Det}(M)$ in Fig. 2. The colored points in Fig. 2(b) show that points of stems or grounds are easy to be inner a plane and detected by $\text{Tr}(M)$. Points of branch bifurcation regions are easy to be in the intersection of planes and captured by E . Points from massive foliage region are

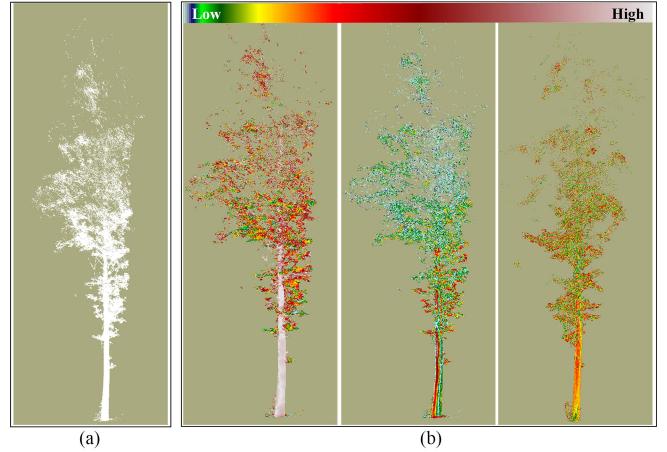


Fig. 2. Gradient-based features. (a) Input point clouds. (b) Calculation of the proposed features. From the left to the right are the detection of no plane intersection, two planes intersection (contours), and more planes intersection (contours and corners).

distinguished by $\text{Det}(M)$. The defined gradient-based features can indicate the location of a point and show the topology of neighbor planes, intersectant, or coplanar, which is promising to supplement the existing geometric features for the classification.

B. Feature Selection, Classification, and Shape Fitting

For the feature selection, we can use either supervised or unsupervised methods for optimizing the feature combination, which will be discussed in experiments based on their performance. In order to find a better feature combination, we first conduct one of the feature selection methods on the extracted features. Then, we select features according to the achieved feature ranking. Empirically, we select the top 5 to 10 features for the classification. For the point classification, this work chooses SVM to detect wood points, because SVM is easy to perform well in binary classification when feature selection has been done.

In the step of shape fitting, we first use the RANSAC method to detect poles from classification results. The axis of the cylinder is the best line fitted by stem points, and the radius is the furthest distance between the stem points and the axis line in a restricted region. Second, we have to separate trunks from other poles based on the points distribution, e.g., traffic signs and lamps. We calculate the kurtosis [9] for each nonphotosynthetic component. If the kurtosis of a nonphotosynthetic component falls in $\mu_k - 1.5\delta_k$ and $\mu_k + 1.5\delta_k$, it will be regarded as a valid component. μ_k and δ_k are, respectively, the mean and standard deviation of the kurtosis of all extracted nonphotosynthetic components.

C. Branch and Leaf Separation

In order to achieve the full structure of trees, one is required to separate wood and foliage points covered by crowns. Commonly used methods for separating wood and foliage are based on the local neighboring information of points, e.g., density and normal vectors. In our work, we formulate the separation of wood points as a global object tracking issue, which is a classical optimization

problem in computer vision. There are two issues that should be addressed in object tracking.

The first issue is the energy function formulation. We formulate the energy function into two terms. A data term to calculate the local information and a smoothness term to collect global information. The energy function is written as

$$T(P, L) = \eta \cdot D(P, L) + S(P, L) \quad (10)$$

where $P = \{p_1, p_2, p_3, \dots, p_n\}$ is the input point set, and L is the label set of points, including “1” for wood points and “0” for those foliage points. $D(P, L)$ is the data term and $S(P, L)$ is the smoothness term. η is a coefficient to balance the data term and smoothness term. The objective function is to find the optimal configuration of L to achieve the minimization as

$$\arg \min_L T(P, L). \quad (11)$$

The data term is decided by assuming that each point can be a candidate wood point. If the density ϕ of a point p is large than a cut-off value t , we think p has more possibility to be a wood point. The data term is used to constrain that more candidate points are passed through by branches as

$$D(P, L) = \sum_i^n d(p, l_p) \quad (12)$$

where

$$d(p, l_p) = \begin{cases} 0, & \phi_p > t \text{ and } l_p = 1 \\ 0, & \phi_p \leq t \text{ and } l_p = 0 \\ 1, & \text{others} \end{cases}. \quad (13)$$

It is worth noting that our data term can deal with occlusions by assigning penalties to noncandidate points. Therefore, the optimal path can retrieve those missing wood points by assigning them a large penalty.

The smoothness term is based on the principal direction of points. We assume that the branch is growing in one direction locally. If the direction of the current points is different from the previous points, there will be a penalty. The calculation of the smoothness term is shown as

$$S(P, L) = \sum_{\{p, p'\}} \sin \langle p, p' \rangle \cdot \sigma(l_p, l_{p'}) \quad (14)$$

where p' and p are spatial neighbors. If the label of p is equal to p' , there will be no penalty, otherwise, there will be a penalty $\sin \langle p, p' \rangle$ based on the principal direction of p and p' , which ranges from 0 to $\frac{\pi}{2}$. The calculation is written as if $l_p = l_{p'}$, $\sigma(l_p, l_{p'}) = 0$, and if $l_p \neq l_{p'}$, $\sigma(l_p, l_{p'}) = 1$.

There is a trick in the calculation of the data and smoothness term. If one calculates terms from the input data directly as shown in Fig. 3(a), both the density and direction results are easy to be incorrect. Because leaves are not static during the data collection. Therefore, we cluster input points based on a simple Euclidean distance clustering, and the result of the clustering is shown in Fig. 3(b) with less noise and clear branches visually. The direction of local points is based on clusters rather than the neighboring points directly using RANSAC, as shown in Fig. 3(c). Fig. 3(d) shows our target wood points and the comparison of target points and the branch direction.

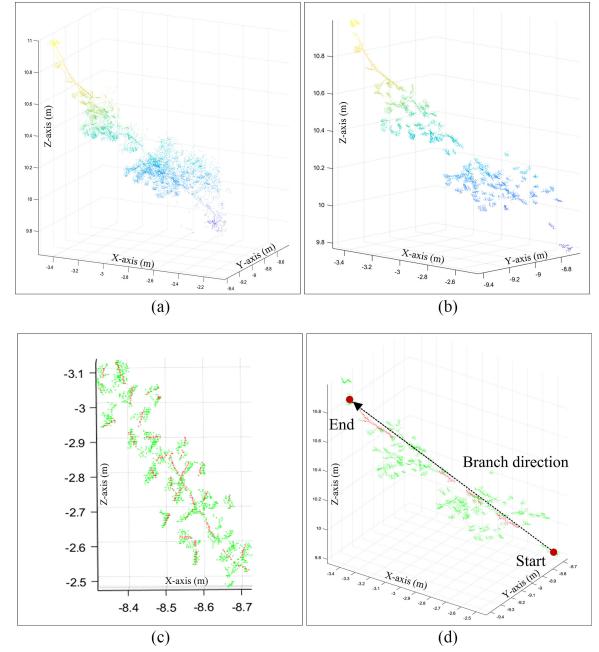


Fig. 3. Preprocessing step for the calculation of the data and smoothness terms. (a) Input point clouds. (b) Euclidean distance clustering results. (c) Local direction of points. (d) Target wood points (red).

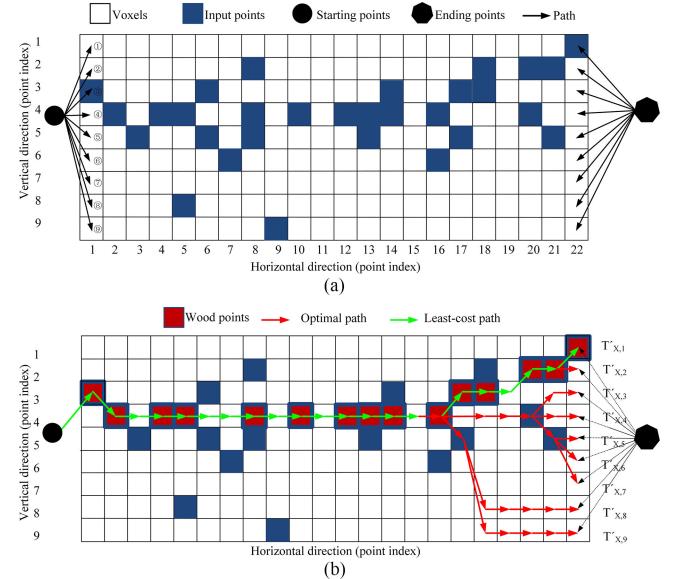


Fig. 4. Dynamic programming for path optimization. (a) Graph construction. (b) Tracking of the least-cost path.

The second issue of the tracking is the minimization of the energy function globally. Intuitively, the number of paths is in an exponential order with the increasing number of points in branches. Therefore, we develop a new model for the optimization of paths by the dynamic programming technique. In order to define the order of points in the path, we organize the input point clouds into voxels. To make our model easy to explain and follow, we describe the proposed model in the 2-D space, as shown in Fig. 4.

In the constructed graph, each node means a voxel, and each edge is weighted based on the data term and smoothness term. We add a virtual starting node and ending node at the left and right of the input data, respectively. A candidate path is supposed to be a full path from the starting node to the ending node. For the current node $v_{i,j}$, where i ranges from 1 to X and j ranges from 1 to Y , and its previous node is $v'_{i',j'}$. In our example, $X = 22$ and $Y = 9$. The stepsize of the node in the horizontal direction is fixed as 1 from left to right; therefore, we have $i' = i - 1$ and the length of the path is X in the horizontal direction. The weights of a path are regarded as the path's penalty. There is no smoothness penalty from the starting node to other nodes and no penalty from nodes to the ending node. This is because the starting and ending nodes are virtual and have no principal direction. In order to calculate the penalty in our graph, the energy function in (10) is rewritten as

$$\begin{aligned} T' &= \eta \cdot D'(V, L) + S'(V, L) \\ &= \eta \cdot \sum_{i=1}^X d(v_{i,j}, l_{i,j}) + \sum_{\{v,v'\}} \sin \langle v_{i,j}, v'_{i',j'} \rangle \cdot \sigma(l_{i,j}, l_{i',j'}) . \end{aligned} \quad (15)$$

We can find all paths from the starting node to the ending node, for example, there are nine different paths from the starting node to the next node as shown in Fig. 4(a); therefore, there are 9^{22} different paths in the optimization space. To find the least-cost path, we choose a dynamic programming technique to optimize paths. Assume that we have found the optimal path from the starting node to the current voxel $v_{21,j}$ as $T'_{21,j}$, the optimal path from $v_{21,j}$ to $v_{22,1}$ is obtained in

$$\begin{aligned} T'_{22,1} &= \min (T'_{21,j} + \eta \cdot d(v_{22,1}, l_{22,1}) \\ &\quad + \sin \langle v_{22,1}, v'_{21,j} \rangle \cdot \sigma(l_{22,1}, l_{21,j})) . \end{aligned} \quad (16)$$

In the calculation, we start from the starting node and we calculate $T'_{i,j}$ by column. Each $T'_{i,j}$ stands for the optimal path from the starting node to $v_{i,j}$. As shown in Fig. 4(b), for each node at the rightmost, there will be an optimal path cost from the starting node to this node $v_{X,j}$. The least-cost path of the constructed graph is decided by $\min(T_{X,j})$. In the implementation, we calculate and store the optimal path from the starting node to each node in the graph. By backtracking from the ending point, we obtain the least-cost path from the starting node to the ending node, as shown in Fig. 4(b). Although there are missing nodes in the path, we can retrieve them by minimizing the energy path, i.e., finding voxels with a large data term penalty while a small smoothness term penalty. More generally, in the 3-D space, the calculation is written as (17). In this equation, i ranges from 2 to X , j ranges from 1 to Y and k ranges from 1 to Z . X, Y, Z means the number of nodes in x, y , and z direction, respectively. The initialization of $T_{1,j,k}$ is $d(v_{1,j,k}, l_{1,j,k})$, which means there is no penalty caused by the smoothness term from the starting node to others. The tracking of paths is similar to the discussion in the case of a 2-D space

$$\begin{aligned} T'_{i,j,k} &= \min (T'_{i-1,j,k} + \eta \cdot d(v_{i,j,k}, l_{i,j,k}) \\ &\quad + \sin \langle v_{i,j,k}, v'_{i-1,j,k} \rangle \cdot \sigma(l_{i,j,k}, l_{i-1,j,k})) . \end{aligned} \quad (17)$$

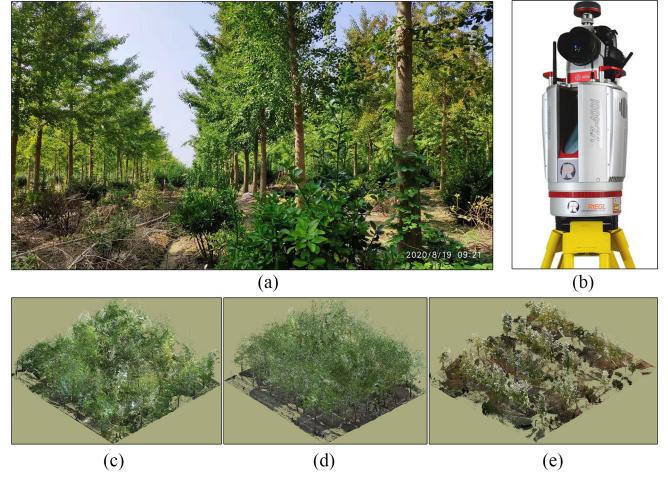


Fig. 5. Data description. (a) Image of the forest. (b) RIEGL VZ-400i scanner. (c) Point clouds of scene #1 (21 685 097 points, 65 trees). (d) Point clouds of scene #2 (21 759 017 points, 46 trees). (e) Point clouds of scene #3 (44 501 592 points, 40 trees).

IV. EXPERIMENTS AND EVALUATIONS

A. Study Area and Data Description

The site of focus is the Dongtai Forest located in the coastal Jiangsu Province of China ($120^{\circ}49'32.2''E$, $32^{\circ}52'20.6''N$). Most of the Dongtai Forest (>85%) is planted with Dawn redwood (*Metasequoia glyptostroboides*), Poplar (*Populus deltoides*), and Ginkgo (*Ginkgo biloba L.*), in various developmental stages. As shown in Fig. 5, experiment data are collected by the RIEGL VZ-400i. The scanner provides up to 1 200 000 points/s. The experimental data include three scenes as shown in Fig. 5(c)–(e). Each scene covers $30\text{ m} \times 30\text{ m}$ ground regions and was scanned by 7 to 8 scans for each forest plot on 2020/08/19. The following section focuses on fully evaluating the proposed classification and separation accuracy.

B. Results of Feature Selection and Classification

Based on the aforementioned feature extraction, we list 23 features in Table I, including commonly used geometric features based on eigenvalues (ID: 1–18) and our proposed topological related features using the defined gradient (ID: 19–23).

The evaluation of the feature selection is by the average ratio of correct classified points in four classes, namely 1) crowns, 2) stems, 3) bushes, and 4) ground points. The manually classified points of crowns, stems, bushes, and ground points are considered as the ground truth and used as a reference to evaluate the classification accuracy quantitatively. The classification accuracy is calculated based on the ratio of points classified correctly. We choose the scene #1 as the demonstration scene, which contains 21 685 097 points. As shown in the first row of Fig. 6, the ground truth of the scene #1 is demonstrated in four pictures from (a) to (d), including crowns, stems, bushes, and ground points. In the training step, we randomly choose 0.01% points of the scene #1 as training points. The rest of the input points are used for testing. Results of the geometric features (ID: 1–18) and complete features (ID: 1–23) are shown in the second

TABLE I
COMPLETE FEATURES

ID	Features	Comments
1	x	Coordinate -X
2	y	Coordinate -Y
3	z	Coordinate -Z
4	Sum	$\lambda_1 + \lambda_2 + \lambda_3$
5	Omnivariance	$(\lambda_1 \cdot \lambda_2 \cdot \lambda_3)^{\frac{1}{3}}$
6	Eigenentropy	$-\sum_{i=1}^3 \lambda_i \cdot \ln(\lambda_i)$
7	Anisotropy	$(\lambda_1 - \lambda_3)/\lambda_1$
8	Planarity	$(\lambda_2 - \lambda_3)/\lambda_1$
9	Linearity	$(\lambda_1 - \lambda_2)/\lambda_1$
10	Surface variation	$\lambda_3/(\lambda_1 + \lambda_2 + \lambda_3)$
11	Sphericity	λ_3/λ_1
12	1 st eigenvalue	λ_1
13	2 nd eigenvalue	λ_2
14	3 rd eigenvalue	λ_3
15	N_x	Normal vector in X-axis
16	N_y	Normal vector in Y-axis
17	N_z	Normal vector in Z-axis
18	NN	Number of neighbors
19	Voxel density	Number of points in a voxel volume
20	Surface density	Number of points in a voxel surface
21	Topology 1	$\text{Tr}(M)$ in Eq.(7)
22	Topology 2	$\text{Det}(M)$ in Eq.(8)
23	Topology 3	E in Eq.(9)

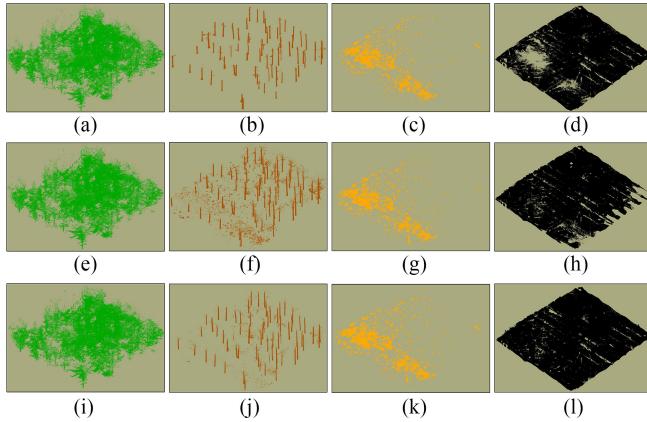


Fig. 6. Classification of the scene #1. (a)–(d) Ground truth by manual. (e)–(h) Classification results by the geometric features only. (i)–(l) Classification results by the complete features.

and third rows of Fig. 6, respectively. As shown in Fig. 6(f) and (j), stems are detected more accurately by adding the proposed topological features. We achieve fewer misclassification points and more complete trunks. The quantitative comparison of the classification accuracy between the geometric features and complete features is shown in Fig. 7. The improvement of the classification accuracy of each class is 0.31%, 3.72%, 0.52%, and 1.20%, respectively. Compared with the classification by geometric features only, we achieve an average improvement of 1.52% in four classes.

To further evaluate our proposed features, we choose both unsupervised [30], [31] and supervised methods [32], [33] for the feature selection to show the weight of each feature in the classification. In terms of the unsupervised feature selection, LS [30] define Laplacian scores to evaluate the importance of a feature by its power of locality preserving. Local Learning-based Clustering Feature Selection (LLCFS) [31] focuses on

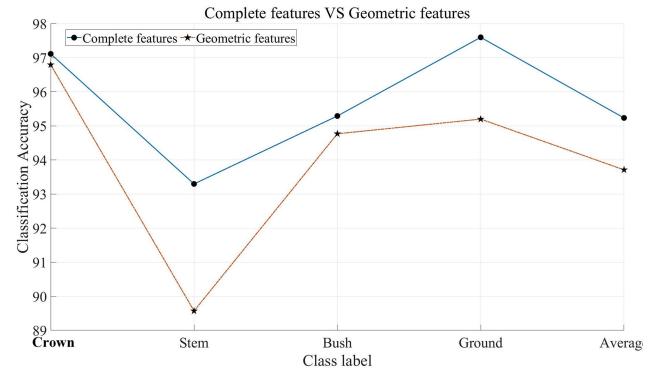


Fig. 7. Comparison of the classification accuracy quantitatively.

the relevance of each feature or kernel for the clustering. For the supervised feature selection, ReliefF [32] is a feature weight algorithm, which gives a higher weight to all features with a high correlation with the classification. Infinite latent feature selection (ILFS) [33] aims to be robust to data by performing the ranking step while considering all the possible subsets of features. We choose the aforementioned four different feature selection methods to evaluate the classification. For each feature selection method, we sort all 23 features based on their importance. The corresponding classification accuracy is demonstrated in Fig. 8.

In Fig. 8, we know that for each feature selection method, we only need a part of the listed features to achieve good accuracy in the classification. The achieved average classification accuracy on four classes is 95.24% by using the complete features. We set the cut-off threshold accuracy for obtaining the optimized features as 95%. The selected features of each method are shown in Table II. In this table, the “optimized features” demonstrates the selected features. “Per.” shows the percentage of the selected topological features in the optimized features. “Red.” illustrates the reduction of the feature dimension. “Acc.” means the classification accuracy based on the selected features.

In our experiment, the best performance is achieved by ReliefF. The feature dimension is optimized from 23 to 11. Compared with the choice of all features, the classification accuracy is improved from 95.24 to 95.35%. From the feature selection results, we find following three key points in the point cloud classification.

- 1) More features do not mean higher accuracy in the classification. As shown in Fig. 8, the accuracy is reduced slightly when we add more features for the classification.
- 2) Combination of optimal features is various. This means it is the combination of features rather than the simple integration of features that plays a key role in the classification.
- 3) There is not too much accuracy difference between the supervised and unsupervised feature selection methods in point cloud classification.

We choose the feature selection result by ReliefF, including 3, 1, 17, 2, 22, 12, 6, 4, 9, 23, for the generalization ability discussion. Experimental scenes include the scene #1 (21 685 097 points, 65 trees), #2 (21 759 017 points, 46 trees), and #3 (44 501 592 points, 40 trees). The training samples

TABLE II
COMPARISON OF FEATURE SELECTION METHODS

Method	Optimized features	Per.	Red.	Acc.
LS	20 19 18 22 1 3 21 2 17 15 8 7	80%	48%	95.13%
LLCFS	1 2 22 20 3 21 15 16 17 9 19 8	80%	48%	95.33%
ReliefF	3 1 17 2 22 12 21 6 4 9 23	60%	53%	95.35%
ILFS	3 8 7 10 11 14 23 1 17 5 13 22 18 19 20 12 15 4 9 21	100%	14%	95.26%

TABLE III
GENERALIZATION ABILITY (%)

Scene Index	#1	#2	#3	#1,#2	#1,#3	#2,#3	#1,#2,#3
#1	95.35	82.92	82.44	87.55	87.70	76.25	82.57
#2	85.08	95.42	83.86	87.69	74.41	88.20	79.20
#3	80.02	82.77	95.48	76.20	83.51	86.36	80.15
#1,#2	91.81	91.95	83.71	91.44	86.10	86.18	87.27
#1#3	95.23	90.03	95.47	91.36	95.24	91.35	91.71
#2,#3	86.70	94.90	94.93	88.00	87.79	95.00	89.02
#1,#2,#3	92.00	91.73	92.04	91.61	92.18	92.18	91.76

are obtained by choosing 0.01% points from the input scene randomly. The classification accuracy is shown in Table III. The first column “Scene Index” indicates the training scene. Each row illustrates the classification accuracy on each testing data. For example, the first row shows the classification accuracy on the scene #1, scene #2, scene #3, scene #1+#2, scene #1+#3, scene #2+#3, and scene #1+#2+#3, when the training data are on the scene #1. The diagonal direction of Table III shows the classification accuracy when the training and testing scenes are the same, which is easy to be high.

The accuracy based on each training scene is illustrated in Fig. 9. The horizontal axis is the training data. The vertical axis is the average classification accuracy on all other scenes, including the scene #1, scene #2, scene #3, scene #1+#2, scene #1+#3, scene #2+#3, and scene #1+#2+#3. To show our superiority, we demonstrate the classification accuracy based on the complete features and geometric features in Fig. 9. The classification accuracy based on the selected optimal features is close to the complete features and higher than the geometric features in most cases. Know that there are only 11 elements in our optimal features, which is 53% of the complete features. The classification results of the scenes #1, #2, and #3 are shown in Fig. 10 by using 0.01% points from scene #1 as the training samples.

In terms of the stem-based evaluation, a stem is regarded as classified correctly if more than 80% points of this stem are assigned with the correct label. For the chosen three scenes, we detect all stems (65+46+40) from the input TLS point clouds, which is important to the following branch tracking for wood and foliage separation.

C. Results of Separation

Our quantitative separation evaluation requires the calculation of true positive (TP), false negative (FN), and false positive (FP). TP means that wood points are detected correctly from the input. FN means that wood points are wrongly detected as foliage points. FP means that foliage points are wrongly detected as wood points. Similarly, wood points separated manually are considered as the ground truth and used as a reference to evaluate

TABLE IV
SEPARATION ACCURACY ANALYSIS

Method	Target	Evaluation	p	r	f
[34]	Wood	Point-based	94.56	66.83	0.78
[35]	Wood	Point-based	81.42	86.67	0.84
[15]	Wood	Point-based	82.18	89.14	0.86
[36]	Wood	Point-based	91.87	84.27	0.88
Ours	Wood	Point-based	91.25	90.34	0.91

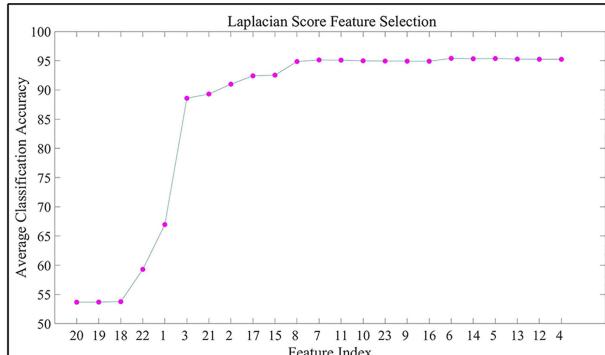
the separation quantitatively. To evaluate extraction results, we calculate the correctness r , completeness p , and F_1 score f as

$$r = \frac{TP}{TP + FP}, p = \frac{TP}{TP + FN}, f = 2 \cdot \frac{r \cdot p}{r + p}. \quad (18)$$

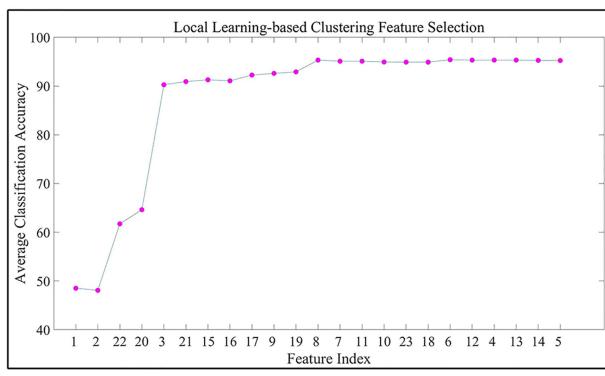
Our separation results are shown in Fig. 11. Fig. 11(a)–(c) shows example results from the scenes #1, #2, and #3, respectively. In each figure, the first column shows input tree point clouds and the achieved stem points (brown) from the classification. The second column shows the ground truth of wood points separated manually. The third column shows the separation results based on the proposed least-cost path model by using the data term only. The last column demonstrates the separation results based on both the data term and the smoothness term. As shown in the ablation analysis, the smoothness term based on the tracking information of branches helps separate lots of foliage from branches. There are still misclassified wood points in our results as shown in the last subfigure in Fig. 11(c). Because the density of points from foliage regions is close to trunks in this case. Besides, if the weather is windy, branches keep swaying, our accuracy will be reduced unavoidably.

In order to show the quantitative accuracy improvement, we choose four methods of wood and foliage separation for the comparison, including [15], [34]–[36]. In our experiment, we have 151 Ginkgo trees scanned by RIEGL VZ-400. Segmentation performances are shown in Table IV. The second column “Target” shows the evaluated region, and “Wood” means all wood points from trees. Our evaluation is based on the correctly segmented wood points, i.e., point-based. The last three columns show the accuracy of completeness, correctness, and F_1 score, respectively. Point-based accuracy shows that we have achieved the best balance of segmentation completeness and correctness.

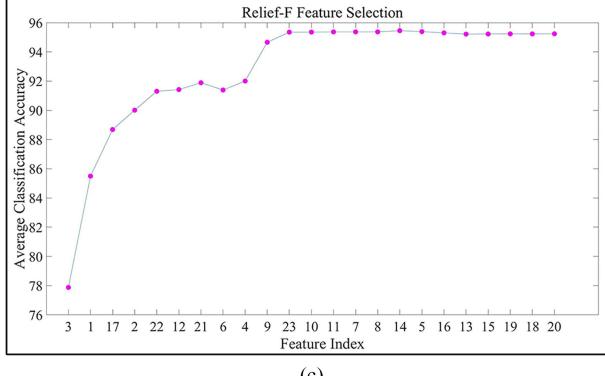
It is worth noting that the chosen compared methods are not open source and it is difficult to reimplement them optimally due to the parameter setting and programming skills. We figure out those algorithms on MATLAB R2019a based on their published algorithm description. According to the work in [34], we organize point clouds in voxels to generate clusters through the point density algorithm DBSCAN (i.e., Density-Based Spatial Clustering of Applications with Noise), and then, we merge



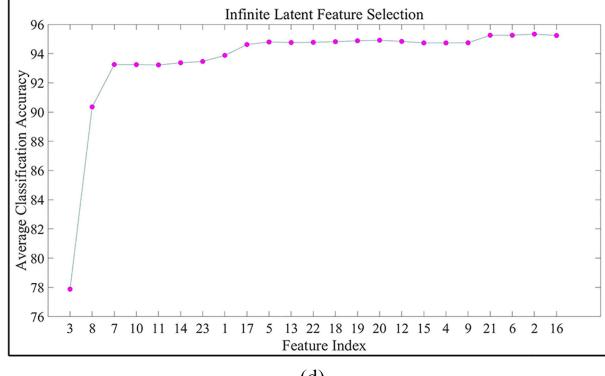
(a)



(b)



(c)



(d)

Fig. 8. Classification accuracy analysis based on the importance of each feature. (a) Laplacian score feature selection. (b) LLCFS. (c) Relief-F feature selection. (d) ILFS.

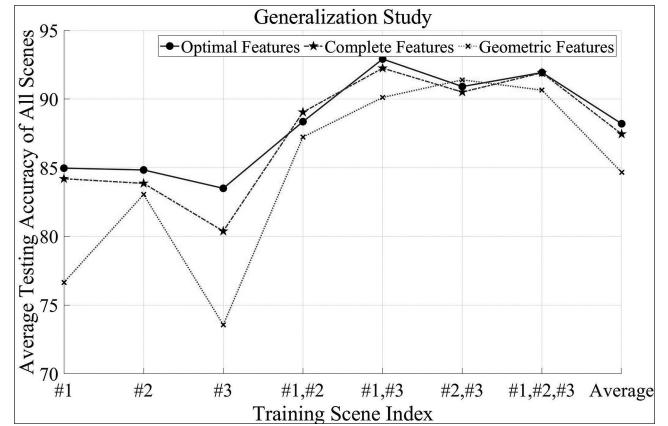


Fig. 9. Generalization ability study.

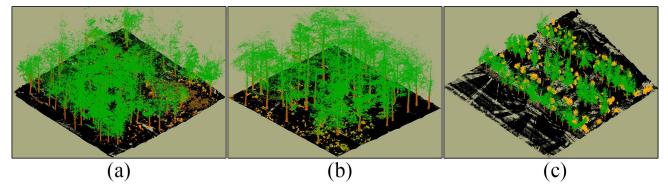


Fig. 10. Classification performance on the scene #1, #2, and #3.

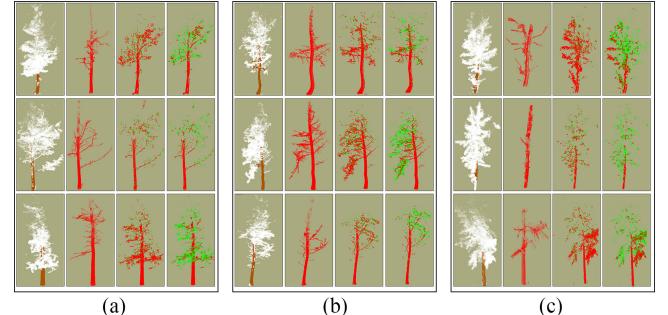


Fig. 11. Separation results of wood points from individual trees. (a) Scene #1. (b) Scene #2. (c) Scene #3.

neighboring clusters containing fewer points (<100). An example segmentation result [34] is shown in Fig. 12(b), which shows that it is not robust to dense foliage regions. According to [35], we extract eigenvalue features from points and use the machine learning model based on the random forest (RF) algorithm (200 trees) to classify points. An example segmentation result [35] is shown in Fig. 12 (c), which misses wood points covered by crowns. According to the work in [15], we use the normal change rate (≤ 0.1) to thin branches and, then, calculate geometric features for filtering stem points using height-to-width ratio. An example segmentation result [15] is shown in Fig. 12(d), which works well in vertically standing stems. According to the work in [36], we first use K-means to classify point clouds into different regions and, then, perform the RANSAC algorithm to obtain cylindrical segments as wood points. An example segmentation result [36] is shown in Fig. 12(e), which misclassifies foliage points from cylinder volumes as wood points. The ground truth is shown in Fig. 12(a), and our result is shown in Fig. 12(f).

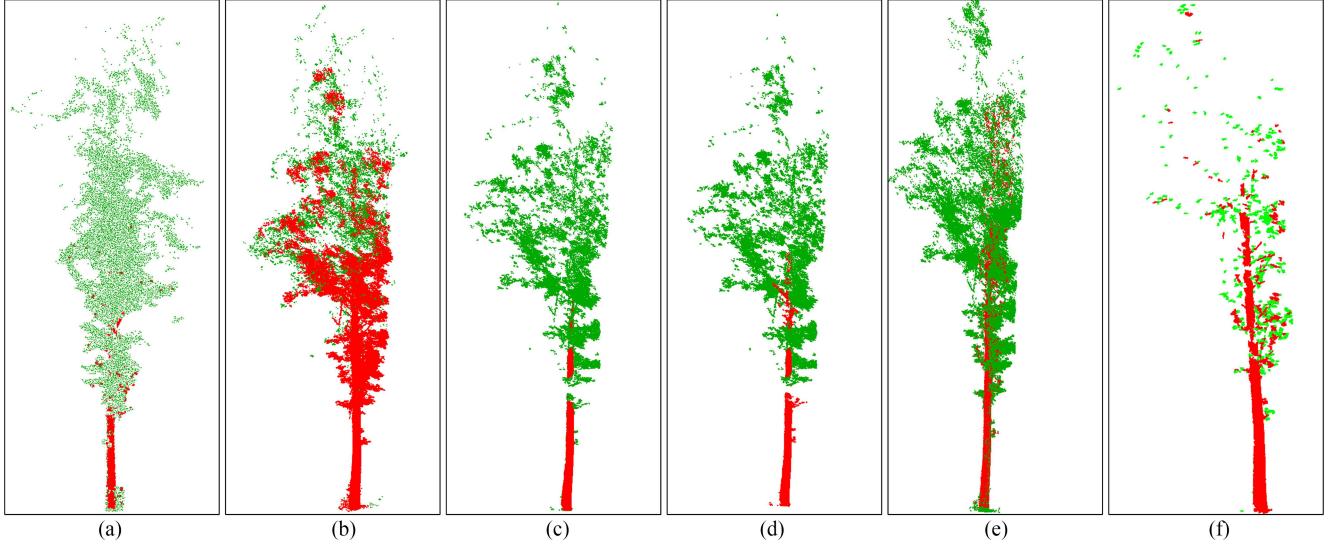


Fig. 12. Comparison of segmentation results. (a) Ground truth. (b) Result from [34]. (c) Results from [35]. (d) Result from [15]. (e) Result from [36]. (f) Our result.

achieving the most of wood points, including points from the second branching order of trees.

V. DISCUSSION

In terms of the complexity, we mainly discuss the time complexity for the first and last steps in the proposed framework. In the feature extraction, we need to do the voxelization for the gradient calculation by splitting the input data into $1 \times 1 \times 1$ cm cubes and assign points to their spatial closest cube centers at the complexity of $\mathcal{O}(N \cdot c)$, where c is the number of cubes and N is the number of points. Second, we are required to calculate $\text{Tr}(\mathbf{M})$, $\text{Det}(\mathbf{M})$, and E for each point at the complexity of $\mathcal{O}(N)$. In the branch and leaf separation, the dynamic programming, including the backtracking process for the refinement, is at the complexity of $\mathcal{O}(X \cdot Y \cdot Z)$, which depends on the size of input data. Our complexity is less than $\mathcal{O}(N^3)$, which is the common complexity in other methods. Experiments were done on a Windows 10 Enterprise 64-b, Intel Core i7-6900 k, 3.20 GHz processor with 64 GB of RAM and computations were carried on MATLAB R2019a. For the scene #1, the calculation of topological features is finished within 1.2 s. The feature selection (1% points for the training) based on ReliefF is finished within 178.9 s. The classification based on SVM is finished within 82.7 and 4.7 s in terms of the training and testing process, respectively. The shape fitting (90% confidence in RANSAC), including the kurtosis calculation, is finished within 61.5 s. The branch and leaf separation is finished within 116.63 s.

In the parameter setting, the size of voxels is $1 \times 1 \times 1$ cm, and the step size for the gradient calculation is $\Delta x = \Delta y = \Delta z = 1$. The density variance σ is set as 1. Theoretically, the parameter setting depends on the point density. However, since the plot size and position have been chosen to split the scene equally, and the distance between trees and the scanner is limited to 5 m, we fix the voxel size as $1 \times 1 \times 1$ cm for different scenes in the separation. In the calculation, the radius of neighbor points is set

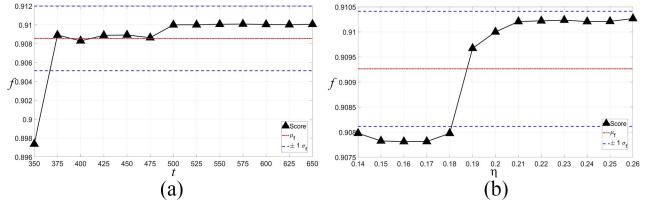


Fig. 13. Sensitivity analysis of the proposed least-cost path model.

at 0.2 m. The number of neighbors for the Euclidean distance clustering is 30 and the minimum distance between two clusters is 0.01 m. Two key parameters in the least-cost path model are t and η , which are set as 500 and 0.2, respectively.

For the purpose of the parameter sensitivity analysis in the proposed least-cost path model, we range all parameters from -30 to 30% with respect to the suggested values. The analysis is conducted by floating one parameter and fixing the rest of the parameters. The accuracy of the aforementioned scenes is shown in Fig. 13 using different parameters. Fig. 13 shows that we can achieve a high F_1 score at different parameter settings according to the suggested value. The mean F_1 score μ_f of different t and η is above 0.90 in the parameter sensitivity analysis.

In terms of extensibility, we test the proposed method on different tree species collected by MLS and HLS. In the data of MLS, points are from the open dataset located in Paris, France [37]. In order to contain more nonphotosynthetic components, we limit the distance between the road and trees to 20 m. The input data describe an urban forest scene containing a mix of trees. In the data of HLS, points are collected at Nanjing, China with GeoSLAM Horizon system, which collects points at the rate of 300 000 points/s. Separation results are shown in Fig. 14, and the proposed method separates wood and foliage effectively in different scanner points and tree species.

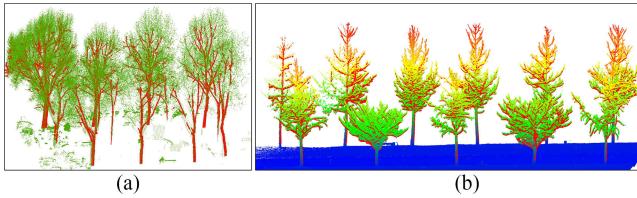


Fig. 14. Separation results of wood points from various trees and scanners. (a) Results of urban forest collected by MLS. (b) Results of street trees collected by HLS.

VI. CONCLUSION

This article introduces a method of foliage and wood separation from ground point clouds, which shows that the terrestrial laser scanning technique helps capture the 3-D spatial structure of trees effectively. Our achieved classification results can benefit the retrieval of biomass information. We have addressed three key issues in the point cloud separation. First, in the pointwise classification, although the most commonly used features based on the geometric information have provided the linearity and planarity of objects, the topological information is missing, which has been supplemented in this study. Second, the existing point cloud features are redundant, which have been optimized at less classification accuracy loss in the feature selection. Third, the proposed least-cost path model improves the extraction of wood points by tracking the branch information globally rather than using the local density information only. Experimental results show that we have achieved completeness, correctness, and F_1 score of 91.25%, 90.34%, and 0.91, respectively, which means a good balance between completeness and correctness in the wood point separation. Besides, we achieve higher completeness than the existing methods with the help of the proposed least-cost path model to connect branches, which shows promising to the phenotyping study in terms of the growing structure.

Although we have achieved high performance in experimental scenes, the proposed framework has two main limitations to be tackled in future work. First one is not able to segment large discontinued branches correctly, which will be solved by an adaptive cylinder fitting method in a local coordinate system. Second lies in the nonautomatic branch growing process, which will be addressed by adding prior knowledge for the starting and ending points initialization.

ACKNOWLEDGMENT

The authors sincerely thank Academician Fuliang Cao for his constructive comments and suggestions.

REFERENCES

- [1] L. Cao *et al.*, “Estimating canopy structure and biomass in bamboo forests using airborne LiDAR data,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 148, pp. 114–129, 2019.
- [2] W. Yan, H. Guan, L. Cao, Y. Yu, S. Gao, and J. Lu, “An automated hierarchical approach for three-dimensional segmentation of single trees using UAV LiDAR data,” *Remote Sens.*, vol. 10, no. 12, 2018, Art. no. 1999.
- [3] S. Xu, R. Wang, H. Wang, and R. Yang, “Plane segmentation based on the optimal-vector-field in LiDAR point clouds,” *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: [10.1109/TPAMI.2020.2994935](https://doi.org/10.1109/TPAMI.2020.2994935).
- [4] B. Yang, Z. Dong, G. Zhao, and W. Dai, “Hierarchical extraction of urban objects from mobile laser scanning data,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 99, pp. 45–57, 2015.
- [5] Q. Li, P. Yuan, X. Liu, and H. Zhou, “Street tree segmentation from mobile laser scanning data,” *Int. J. Remote Sens.*, vol. 41, no. 18, pp. 7145–7162, 2020.
- [6] S. Xu, X. Sun, J. Yun, and H. Wang, “A new clustering-based framework to the stem estimation and growth fitting of street trees from mobile laser scanning data,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3240–3250, Jan. 2020.
- [7] Z. Dong *et al.*, “Registration of large-scale terrestrial laser scanner point clouds: A review and benchmark,” *ISPRS J. Photogrammetry Remote Sens.*, vol. 163, pp. 327–342, 2020.
- [8] S. Xia, C. Wang, F. Pan, X. Xi, H. Zeng, and H. Liu, “Detecting stems in dense and homogeneous forest using single-scan TLS,” *Forests*, vol. 6, no. 11, pp. 3923–3945, 2015.
- [9] L. Zhong, L. Cheng, H. Xu, Y. Wu, Y. Chen, and M. Li, “Segmentation of individual trees from TLS and MLS data,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 10, no. 2, pp. 774–787, Feb. 2017.
- [10] C. Hu, S. Pan, H. Zhang, and P. Li, “Trunk model establishment and parameter estimation for a single tree using multistation terrestrial laser scanning,” *IEEE Access*, vol. 8, pp. 102 263–102277, 2020.
- [11] K. Tan, W. Zhang, Z. Dong, X. Cheng, and X. Cheng, “Leaf and wood separation for individual trees using the intensity and density data of terrestrial laser scanners,” *IEEE Trans. Geosci. Remote Sens.*, to be published, doi: [10.1109/TGRS.2020.3032167](https://doi.org/10.1109/TGRS.2020.3032167).
- [12] T. Yun *et al.*, “Simulation of multi-platform LiDAR for assessing total leaf area in tree crowns,” *Agricultural Forest Meteorol.*, vol. 276, 2019, Art. no. 107610.
- [13] X. Liang, P. Litkey, J. Hyppä, H. Kaartinen, M. Vastaranta, and M. Holopainen, “Automatic stem mapping using single-scan terrestrial laser scanning,” *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 2, pp. 661–670, Feb 2012.
- [14] A. Lau *et al.*, “Quantifying branch architecture of tropical trees using terrestrial LiDAR and 3D modelling,” *Trees*, vol. 32, no. 5, pp. 1219–1231, 2018.
- [15] W. Zhang *et al.*, “A novel approach for the detection of standing tree stems from plot-level terrestrial laser scanning data,” *Remote Sens.*, vol. 11, no. 2, 2019, Art. no. 211.
- [16] S. Tao, Q. Guo, S. Xu, Y. Su, Y. Li, and F. Wu, “A geometric method for wood-leaf separation using terrestrial and simulated LiDAR data,” *Photogrammetric Eng. Remote Sens.*, vol. 81, no. 10, pp. 767–776, 2015.
- [17] L. Ma, G. Zheng, J. U. H. Etel, L. M. Moskal, W. He, and H. Huang, “Improved salient feature-based approach for automatically separating photosynthetic and nonphotosynthetic components within terrestrial LiDAR point cloud data of forest canopies,” *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 679–696, Feb. 2016.
- [18] D. Wang, M. Hollaus, and N. Pfeifer, “Feasibility of machine learning methods for separating wood and leaf points from terrestrial laser scanning data,” *ISPRS Ann. Photogrammetry, Remote Sens., Spatial Inf. Sci.*, vol. 4, pp. 157–164, 2017.
- [19] M. Chen, Y. Wan, M. Wang, and J. Xu, “Automatic stem detection in terrestrial laser scanning data with distance-adaptive search radius,” *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2968–2979, May 2018.
- [20] Z. Koma, M. Rutzinger, and M. Bremer, “Automated segmentation of leaves from deciduous trees in terrestrial laser scanning point clouds,” *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 9, pp. 1456–1460, Sep. 2018.
- [21] Z. Li, A. Strahler, C. Schaaf, D. Jupp, M. Olofsson, and P. Olofsson, “Seasonal change of leaf and woody area profiles in a midlatitude deciduous forest canopy from classified dual-wavelength terrestrial LiDAR point clouds,” *Agricultural Forest Meteorol.*, vol. 262, pp. 279–297, 2018.
- [22] X. Zou, M. Cheng, C. Wang, Y. Xia, and J. Li, “Tree classification in complex forest point clouds based on deep learning,” *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 12, pp. 2360–2364, Dec. 2017.
- [23] Z. Yang, W. Jiang, B. Xu, Q. Zhu, S. Jiang, and W. Huang, “A convolutional neural network-based 3D semantic labeling method for ALS point clouds,” *Remote Sens.*, vol. 9, no. 9, 2017, Art. no. 936.
- [24] Z. Xi, C. Hopkinson, and L. Chasmer, “Filtering stems and branches from terrestrial laser scanning point clouds using deep 3D fully convolutional networks,” *Remote Sens.*, vol. 10, no. 8, 2018, Art. no. 1215.
- [25] L. Windrim and M. Bryson, “Detection, segmentation, and model fitting of individual tree stems from airborne laser scanning of forests using deep learning,” *Remote Sens.*, vol. 12, no. 9, 2020, Art. no. 1469.
- [26] S. Xia, D. Chen, R. Wang, J. Li, and X. Zhang, “Geometric primitives in LiDAR point clouds: A review,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 685–707, Jan. 2020.

- [27] T. Hackel, J. D. Wegner, and K. Schindler, "Contour detection in unstructured 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1610–1618.
- [28] C. G. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, pp. 147–151.
- [29] S. Xu, R. Wang, and H. Zheng, "Road curb extraction from mobile LiDAR point clouds," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 996–1009, Feb. 2017.
- [30] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," *Adv. Neural Inf. Process. Syst.*, vol. 18, pp. 507–514, 2005.
- [31] H. Zeng and Y.-M. Cheung, "Feature selection and kernel learning for local learning-based clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1532–1547, Aug. 2011.
- [32] N. Spolaôr, E. A. Cherman, M. C. Monard, and H. D. Lee, "Relieff for multi-label feature selection," in *Proc. IEEE Braz. Conf. Intell. Syst.*, 2013, pp. 6–11.
- [33] G. Roffo, S. Melzi, U. Castellani, and A. Vinciarelli, "Infinite latent feature selection: A probabilistic latent graph-based ranking approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1398–1406.
- [34] R. Ferrara, S. G. Virdis, A. Ventura, T. Ghisu, P. Duce, and G. Pellizzaro, "An automated approach for wood-leaf separation from terrestrial LiDAR point clouds using the density based clustering algorithm DBSCAN," *Agricultural Forest Meteorol.*, vol. 262, pp. 434–444, 2018.
- [35] S. M. K. Moorthy, Y. Bao, K. Calders, S. A. Schnitzer, and H. Verbeeck, "Semi-automatic extraction of liana stems from terrestrial LiDAR point clouds of tropical rainforests," *ISPRS J. Photogrammetry Remote Sens.*, vol. 154, pp. 114–126, 2019.
- [36] Z. Su, S. Li, H. Liu, and Y. Liu, "Extracting wood point cloud of individual trees based on geometric features," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1294–1298, Aug. 2019.
- [37] B. Vallet, M. Brédif, A. Serna, B. Marcotegui, and N. Paparoditis, "TerraMobilis/iQmulus urban point cloud analysis benchmark," *Comput. Graph.*, vol. 49, pp. 126–133, 2015.



Sheng Xu received the B.Eng. degree in computer science and technology from Nanjing Forestry University, Nanjing, China, in 2010, and the Ph.D. degree in digital image systems from the University of Calgary, Calgary, AB, Canada, in 2018.

In 2018, he joined the College of Information Science and Technology, Nanjing Forestry University, where he is currently an Associate Professor. His research interests include mobile mapping, vegetation mapping, and computer vision.



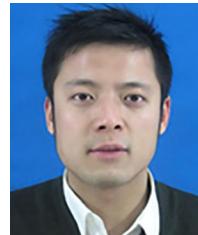
Kai Zhou received the B.S. degree in remote sensing science and technology from the Nanjing University of Information Science and Technology, Nanjing, China, in 2010, and the Ph.D. degree in Agricultural Information from Nanjing Agricultural University, Nanjing, in 2018.

His research interests include nitrogen status diagnosis of trees with spectroscopy, forest volume estimation, and precision silviculture.



Yuan Sun received the B.S. degree in forestry and the M.S. degree in forest management from Nanjing Forestry University, Nanjing, China, in 2000 and 2006, respectively, and the Ph.D. degree in forest biometry from the Faculty of Forest and Environmental Sciences, Freiburg University, Freiburg im Breisgau, Germany, in 2011.

In late 2011, she joined the Faculty of Forestry, Nanjing Forestry University, and was appointed as an Associate Professor of Forest Management. Her research interests include forest mensuration, forest assessment, and sustainable forest management.



Ting Yun received the bachelor's degree in computer science, the master's degree in power engineering, and the Ph.D. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 2002, 2004 and 2008 respectively.

He is currently a Professor with the School of Computer Science, Nanjing Forestry University, Nanjing. He is also a Postdoctoral Fellow with the School of Informatics and Computing, Southeast University, Nanjing. His research interests include computer vision, computer graphics, and LiDAR survey for forest

applications.