

Automatic extraction of street trees' nonphotosynthetic components from MLS data

Sheng Xu^a, Shanshan Xu^{a,*}, Ning Ye^a, Fa Zhu^b

^a College of Information Science and Technology, Nanjing Forestry University, Nanjing, China

^b School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

ARTICLE INFO

Keywords:

Nonphotosynthetic components
Stem
Individual tree
MLS
Urban environment
Clustering

ABSTRACT

This paper aims to propose a cluster-based approach for trees' nonphotosynthetic components extraction from mobile LiDAR point clouds. The presented algorithm uses a bottom-up hierarchical clustering strategy to combine clusters belonging to nonphotosynthetic components. The combination process depends on the dissimilarity between two clusters. The measure in the proximity matrix calculation consists of a distance term using the Euclidean distance and a direction term based on the principal direction, respectively. The main contribution of this paper is to solve the optimization of cluster combination by minimizing the proposed energy function and to extract nonphotosynthetic components through a hierarchical clustering process automatically. Performance of the proposed nonphotosynthetic components extraction shows that we achieve the completeness of 94.0%, the correctness of 98.9% and the *F*-score of 0.96 on the experimental urban scene. Besides, we succeed to achieve promising results on the stem detection and individual tree segmentation based on the extracted nonphotosynthetic component.

1. Introduction

Roadside vegetation is an important component of the urban environment and ecosystem, especially the street tree which plays a significant role in the pollution reduction and urban landscape. The nonphotosynthetic component of an individual tree refers to its main stem and branches, which is critical to retrieve the biophysical parameters, e.g. the biomass productivity (Edson and Wing, 2011) and carbon storage (Yun et al., 2016), and monitor the tree growth, e.g. the diameter at the breast height (DBH) (Kwak et al., 2007; Yao et al., 2012) and plant density (Korpela et al., 2010). Nowadays, the light detection and ranging (LiDAR) technique succeeds to collect 3D information of objects using high-density point clouds, which provides a chance for mapping 3D tree structure accurately. The following is a brief discussion of the related work on the nonphotosynthetic component extraction, including the stem detection and individual tree segmentation, from mobile laser scanning (MLS) data and terrestrial laser scanning (TLS) data.

The stem detection refers to the extraction of the main structural axis of a tree, which includes points between the ground and the first leaf branch. Lehtomäki et al. (2010) develop a method for the extraction of pole-like objects from MLS data. Their framework requires the segmentation work to split each scan line, the clustering process to

merge the region of interest points, and the refinement step to combine clusters from the same pole and recognize each pole-like object. The problem is the low robustness to outliers, i.e. when there are scattered points around stems, trunks are difficult to be detected. Hetti Arachchige (2013) proposes a geometric-feature-based method for the tree stem segmentation from MLS data in urban environments. Their idea is to filter points from stems based on the principal direction feature which is calculated by exploring the variance of a point's neighborhood. This method does not need any priori knowledge of the tree, e.g. the size or height, and can be used for various kinds of tree structures. However, the performance of the stem growing is degraded by the break or hole on the stem caused by occlusion or data incompleteness. Liang et al. (2014) propose a method for mapping large forest plots with MLS data. They establish a local coordinate system to detect points on a vertical planar first and then use a series of 3D cylinders to describe stem sections. They achieve a high performance in the experimental forest, where stem points are from pole-like objects. Xia et al. (2015) propose a method to detect the stem of bamboos from TLS data. They succeed to distinguish neighboring stems and merge the thin structure stems from the same bamboo without a cylinder fitting process. However, a complex classification process is required before the stem detection to classify stem points. Xu et al. (2018) propose a stem detection method based on the dynamic programming technique

* Corresponding author.

E-mail address: shanshanxus@gmail.com (S. Xu).

with MLS data. The key steps include the detection process to extract candidate trees and the optimization process to achieve optimal stems. Their method works well in incomplete trees caused by the data collection and occlusion, but the detection performance highly relies on the circle fitting process, which incurs problems in the mix scene of trees and other pole-like objects.

The individual tree segmentation refers to extract independent tree crown points from input data. Pfeifer et al. (2004) propose a method for the reconstruction of individual trees in forests from TLS data. At the beginning, they model local branches of each tree by using the circular cylinder fitting. Then, they capture the axis direction and the axis position by setting a radius for the cylinder fitting. Finally, they track cylinders along branches and reconstruct the entire tree. They achieve a high performance in obtaining individual trees in the forest, where most trees stand vertically. Pu et al. (2011) propose a framework for recognizing different objects from MLS data. It starts with an initial rough classification, including the ground surface, objects on the ground, and objects off the ground. Based on the obtained segments' features, e.g. the size, shape, orientation and topological relationships, the objects on the ground are assigned to more detailed classes such as traffic signs, trees, and building walls. They do not require samples for training in the classification and succeed to recognize points from an individual tree. However, this knowledge-based method highly relies on the parameter setting, especially the vegetation region with various point densities. Raumonon et al. (2013) propose a method for efficiently obtaining the nonphotosynthetic component of individual trees, including trunks and branches, from TLS data. Their idea is to make a flexible cylinder model to reconstruct the surface of trees, after which the branches are modeled as collections of cylinders. They perform well in the nonphotosynthetic component extraction from artificial trees, however, the validation with a large number of trunks in the real scene is unmentioned in their work. Wu et al. (2013) propose a voxel-based method to segment tree crowns from MLS data. Their voxel-based morphological model succeeds to make full use of points' spatial information in the vertical and horizontal directions. They provide an improvement over 2D neighborhood search methods in the split of overlapping tree crowns. However, they detect the multiple-stemmed street tree as several separated trees in the urban scene. Tao et al. (2015) develop a shortest-path algorithm to help segment the overlapping region of trees from MLS data. They first detect trunks based on the assumption that trunks are separated from each other, and then segment crowns based on the fact that vascular plants tend to minimize the transferring distance to the root. They achieve a high accuracy in extracting the roadside trees from both TLS data and MLS data. However, incomplete stems are potential to be missed in their trunks detection. Fan et al. (2016) present a classification method to localize urban trees with MLS data. They organize off-ground points by the voxel technique first, and then localize candidate trees by setting thresholds for the elevation. After this, they extract object features based on the geometric information to classify tree points. They achieve high accuracy in the experimental palm trees, however, the non-adaptive thresholding method is easy to cause problems in a general urban scene with different tree structures. Li et al. (2016) propose a growing strategy to segment tree crowns from MLS data. They first remove artifacts by a coarse classification process to obtain the candidate tree clusters. After that, they select tree seeds for the following trunk points growing. Finally, they propose a dual growing process to separate one tree from others by circumscribing a trunk in a growing radius and segment a crown in the constrained growing region. Their method works well in different street trees, however, their tree growing process fails to deal with the trunk curvature and incompleteness. Zhong et al. (2017) provide a top-down segmentation pipeline for the individual tree segmentation, which includes a connectivity-based spatial clustering, a stem-based segmentation, and a normalized-cut-based refinement. This algorithm performs well in the split of neighboring trees by using their modified node similarity calculation. The problem is that the

localization of stems relies on the horizontal histogram of the formulated nodes, which is easy to be affected by the point density.

Both MLS and TLS system succeed to collect the side information of objects using 3D point clouds. TLS system is flexible to collect data in different environments, such as the mountain and forest area. However, TLS data are limited to a small-scale scene, and the collection usually requires multiple scans, which brings the task of the point registration. In comparison, MLS system is easy to provide abundant side information of street trees, e.g. the trunks and branches, in a large-scale region. In MLS data collection, the mean point density along the trajectory is over 750 pts/m², which are suitable for the urban tree research. However, due to the fact that scenes in MLS data are more complicated than TLS data, e.g. much contamination generated in the collection and various artifact objects, the accuracy of the tree point extraction is far from being desired.

The objective of this paper is to propose a new bottom-up hierarchical clustering algorithm to the nonphotosynthetic component extraction from MLS point clouds. The clustering process starts with a number of clusters, and then conducts a series of merging operations to combine clusters from the same nonphotosynthetic component. Two main contributions of this paper are: (1) we provide a new clustering approach to the nonphotosynthetic component extraction from MLS point clouds, and (2) we succeed to optimize the cluster combination based on minimizing the proposed energy function. Our cluster combination is automatic and is globally optimal, which provides a better extraction result than methods using the greedy strategy in the grouping of the interest region.

This paper is organized as follows. Section 2 overviews the framework of the proposed nonphotosynthetic component extraction algorithm. Section 3 focuses on the calculation of the dissimilarity between clusters. Section 4 presents the optimization of the cluster combination based on the energy function minimization. Section 5 shows experiments to evaluate the performance of the proposed extraction. The conclusions are outlined in Section 6.

2. An overview of the proposed nonphotosynthetic component extraction algorithm

The process of the nonphotosynthetic component extraction is shown in Fig. 1. The first step is the data preprocessing for the outlier removal and the ground point filtering. The outlier removal is based on the calculation of the mean μ and standard deviation σ of k -nearest neighbors (Rusu et al., 2008). In our work, k is 30 and points between $\mu - \sigma$ and $\mu + \sigma$ are regarded as valid points. To improve the efficiency of data processing, ground points are filtered before the subsequent component extraction. As mentioned in the work of Xu et al. (2018), road points are much denser than off-ground points in MLS data in the urban environment. Since road points are lower than off-ground points, peaks in the elevation histogram can be used to find the elevation threshold for the ground filtering. Points that are lower than the elevation of the achieved peak point are regarded as ground points. Details of optimizing the peak point selection for the ground point removal are shown in Xu et al. (2018).

The second step aims to extract the points from the stems or branches, and merge them into a complete nonphotosynthetic component based on a bottom-top hierarchical clustering strategy. Assume that the input point set is P and the cluster set is C . Each element in C is a cluster containing one or more points from P . The goal is to achieve that each element in C is a point cluster of the complete nonphotosynthetic component of an individual tree. Main steps of the proposed hierarchical clustering are as follows.

- (1) Start with a cluster set C and each element contains a point from P .
- (2) Formulate the proximity matrix by calculating the dissimilarity between clusters.
- (3) Solve the optimal combination solution to merge clusters and

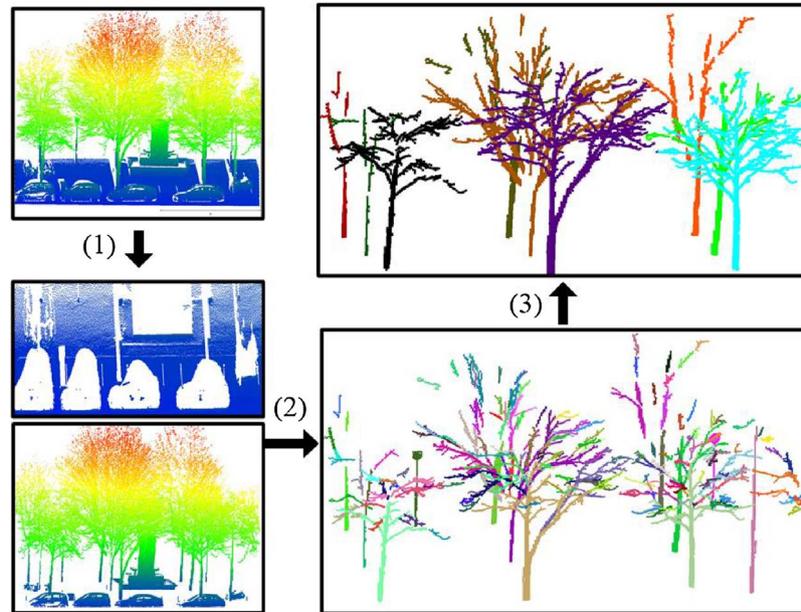


Fig. 1. Main steps of the proposed nonphotosynthetic component extraction, including (1) the preprocessing step to split the input data into ground and off-ground points, (2) the clustering step to extract and merge nonphotosynthetic component points and (3) the refinement step to remove false nonphotosynthetic components in the result.

update the set C .

- (4) Repeat (2) and (3) until C converges, i.e. the number of clusters in C is stable.

Keys in the above hierarchical clustering process are the step (2) and (3) which will be fully discussed in Sections 3 and 4, respectively.

The third step is the refinement process, which aims to (1) address the over-clustering, i.e. the nonphotosynthetic component of an individual tree is grouped into several clusters, (2) address the under-clustering, i.e. a single cluster contains points from more than one individual tree and (3) remove false nonphotosynthetic components, e.g. the streetlights and traffic signs. The refinement is based on the rule that each individual tree has only one main root, and details are as follows.

First, we subtract the spatially closest ground point for each off-ground point achieved in the preprocessing step to obtain its elevation. Then, we perform the Euclidean distance clustering to group low points (< 1 m) as root clusters for the following refinement. (1) Separated tree branch clusters, which are the main reason of appearing the over-clustering, will be combined in the refinement as shown in Fig. 2(a). We detect clusters that are 3 m above the ground region and merge them with their closest root cluster. (2) A group containing more than one root cluster will be divided into more clusters as shown in Fig. 2(b). We divide the group into two different clusters by assigning points to their closest root cluster using the minimum distance rule. (3) Pole-like

objects are easy to be wrongly detected as trunks in the clustering as shown in Fig. 2(c). The distinguish between trees and other pole-like objects is based on the points' distribution in the vertical direction. We calculate the vertical kurtosis (Zhong et al., 2017) for each individual tree's nonphotosynthetic component to remove outlier components. If the kurtosis of a nonphotosynthetic component cluster falls in $\mu_k - 1.5\sigma_k$ and $\mu_k + 1.5\sigma_k$, it will be regarded as a valid component. μ_k and σ_k are the mean and standard deviation of the kurtosis of all extracted nonphotosynthetic components.

3. The calculation of the dissimilarity between clusters

In a general hierarchical clustering approach, we are required defining a symmetric matrix called the proximity matrix PM to calculate the dissimilarity between clusters. The (i, j) th element of the matrix, i.e. $PM(c_i, c_j)$, measures the dissimilarity between the cluster c_i and c_j in C . The clustering principle is that two clusters with a low dissimilarity are preferred to be combined into one cluster. In our work, the calculation of the dissimilarity is based on a distance term $dis(p_i, p_j, c_i, c_j)$ and a direction term $dir(p_i, p_j, c_i, c_j)$. The distance term $dis(p_i, p_j, c_i, c_j)$, which is defined as the measure of the distances between the point p_i and p_j , is formed as

$$dis(p_i, p_j, c_i, c_j) = \|p_i - p_j\|, \quad (1)$$

where $\|p_i - p_j\|$ is the Euclidean distances between the point p_i and p_j .

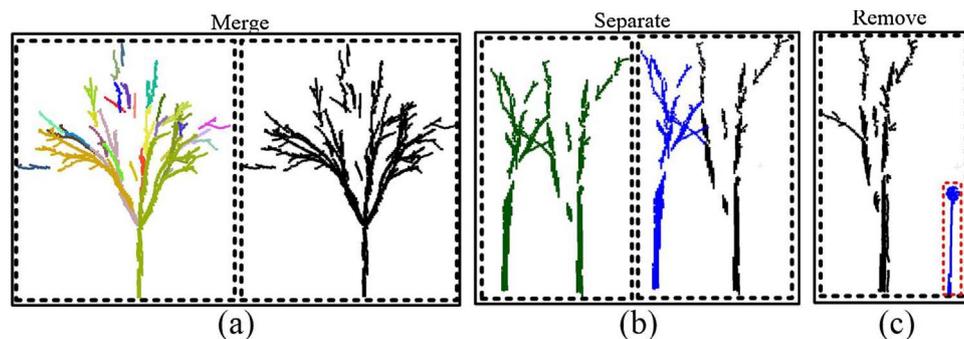


Fig. 2. Refinement process. (a) Separated tree branch clusters are merged with their spatially closest root cluster. (b) A nonphotosynthetic component containing two root clusters is separated into two components. (c) The false component is removed based on the filtering of its vertical kurtosis.

Our second term is based on the principal direction, which is estimated by the singular value decomposition (SVD) method. The principal direction at a point p_i is estimated from its k -nearest neighbors. From SVD we have

$$\mathbf{D}_{k \times 3} = \mathbf{U}_{k \times k} \cdot \mathbf{S}_{k \times 3} \cdot \mathbf{V}_{3 \times 3}^T, \quad (2)$$

where \mathbf{D} is the input matrix formulated by k -nearest neighbors of p_i and will be decomposed into the matrices \mathbf{U} , \mathbf{S} and \mathbf{V} . For a point p_i , denote the first, second and third column vector of \mathbf{V} as $\mathbf{V}_1^{p_i}$, $\mathbf{V}_2^{p_i}$ and $\mathbf{V}_3^{p_i}$, respectively. Denote the diagonal elements in \mathbf{S} as $S_{(1,1)}^{p_i}$, $S_{(2,2)}^{p_i}$ and $S_{(3,3)}^{p_i}$, respectively, which are regarded as eigenvalues of $\mathbf{D}\mathbf{D}^T$ in the decomposition. The principal component $\mathbf{V}_1^{p_i}$ corresponding to the largest eigenvalue $S_{(1,1)}^{p_i}$ will be chosen as the principal direction at the point p_i . The direction term $\text{dir}(p_i, p_j, c_i, c_j)$, which is defined as the measure of the direction differences at the point p_i and p_j , is formed as

$$\text{dir}(p_i, p_j, c_i, c_j) = \frac{1 - \cos(\langle \mathbf{V}_1^{p_i}, \mathbf{V}_1^{p_j} \rangle)}{S_i + S_j}, \quad (3)$$

where

$$S_i = \frac{S_{(1,1)}^{p_i}}{S_{(1,1)}^{p_i} + S_{(2,2)}^{p_i} + S_{(3,3)}^{p_i}}, \quad S_j = \frac{S_{(1,1)}^{p_j}}{S_{(1,1)}^{p_j} + S_{(2,2)}^{p_j} + S_{(3,3)}^{p_j}}.$$

The numerator in Eq. (3) aims to evaluate the principal direction differences at the point p_i and p_j . The denominator in Eq. (3) aims to evaluate the weight of the principal direction in the decomposed vector \mathbf{V} . If the point is from the nonphotosynthetic component, V_i will be consistent with the direction of the tree growth and usually large than 0.9. If two clusters share the similar principal direction, the direction term will be small, which means the combination of these two clusters.

According to our priori knowledge, the distances between the street trees and the sensor range from 5 m to 30 m in MLS data. For an individual tree, the density of points from the nonphotosynthetic component is higher than leaves. In order to separate the nonphotosynthetic component and leaves, we add a coefficient to the formulation of the proximity matrix \mathbf{PM} to balance the distance and direction term as

$$\mathbf{PM}(c_i, c_j) = \lambda \cdot \text{dis}(p_i, p_j, c_i, c_j) + (1 - \lambda) \cdot \text{dir}(p_i, p_j, c_i, c_j), \quad (4)$$

where

$$\lambda = \frac{\sum_{p_k \in k(p_i)} e^{-\|p_i - p_k\|}}{k}$$

and

$$\{p_i, p_j\} = \text{argmin}_{\{p_i, p_j\}} \|p_i - p_j\|: p_i \in c_i, p_j \in c_j.$$

$k(p_i)$ is the set of k -nearest neighbor points of p_i . The coefficient λ ranges from 0 to 1, which depends on the point's neighborhood density. In our work, the dissimilarity between two clusters is small if they are spatially close and they have the consistent principal direction. When both two clusters are from nonphotosynthetic components, the dissimilarity is dominated by the distance term, otherwise, the dissimilarity is decided by the direction term.

In our hierarchical clustering process, if a cluster contains a large planar area, i.e. both its length and width are larger than 0.5 m, this cluster is regarded as not from nonphotosynthetic components and will be removed before the subsequent cluster combination. This rule works well for the facades and vehicles removal. Besides, after the clustering process, clusters with points less than 100 will be removed. This criterion works well for the leaves removal. Since distances between leaf clusters are larger than nonphotosynthetic component clusters, and there is no consistent principal direction in leaf clusters, the dissimilarity between leaf clusters and other clusters are large. A leaf cluster tends to be converged in a local region, therefore, it usually contains a small number of points. The removal of nonphotosynthetic component clusters based on the planar detection brings isolated point clusters

after the clustering process. Therefore, isolated clusters, which cannot be assigned with a root cluster in 5 m, will be removed after the hierarchical clustering process.

4. Hierarchical clustering based on the combination optimization

4.1. Combination optimization by the energy function minimization

The general hierarchical clustering approach uses a greedy strategy to combine clusters. The idea is to find two most similar clusters based on the proximity matrix and combine them into one cluster. The problem is that the greedy strategy is sensitive to the parameter setting and easy to incur a local optimization. In this section, we will formulate the cluster combination as a problem of the energy function minimization and derive the update formulation for achieving the optimal solution.

Assume that there are n clusters in C . We define a symmetrical matrix \mathbf{N} called the combination matrix to represent the combination solution. The (i, r) th element in \mathbf{N} is denoted as N_{ir} , which has a binary value $\{0, 1\}$. If N_{ir} is 1, N_{ri} will be also 1, which means that the cluster c_i will be combined with c_r in the clustering process. The combination between c_i and c_r is weighed by a weight matrix \mathbf{W} as $W_{ir,ri}$. The matrix \mathbf{W} is also symmetrical and $W_{ir,ir}$ is defined as zero. In the optimization of the cluster combination, we consider two constraints in the energy function formulation, i.e. the uniqueness constraint and the optimization constraint. The uniqueness constraint is based on the fact that at each hierarchy, a cluster can be combined with only one cluster in the combination process. The uniqueness constraint is formulated as

$$\sum_{i=1}^n \left(1 - \sum_{r=1}^n N_{ir} \right)^2. \quad (5)$$

Eq. (5) means that in each row of \mathbf{N} , there is only one element assigned the value 1. The optimization constraint aims to find the combination solution at the minimum cost based on the calculated proximity matrix, which is defined as

$$- \sum_{i=1}^n \sum_{r=1}^n W_{ir,ri} N_{ir} N_{ri}, \quad (6)$$

where

$$W_{ir,ri} = \frac{2}{1 + e^{\gamma \cdot D}} - 1, \quad D = \mathbf{PM}(c_i, c_r) + \mathbf{PM}(c_r, c_i).$$

γ is to tune the smoothness of the weight, e.g. $\gamma = 0.1$ in our work. The weight $W_{ir,ri}$ tends to be 0 when the combination cost D is small. The relationship between the weight and the cost is shown in Fig. 3.

Based on the above-mentioned two constraints, the object energy function is

$$E = - \sum_{i=1}^n \sum_{r=1}^n W_{ir,ri} N_{ir} N_{ri} + \sum_{i=1}^n \left(1 - \sum_{r=1}^n N_{ir} \right)^2. \quad (7)$$

In order to optimize Eq. (7), let us consider the energy function Eq. (7) in a more general case as

$$E = - \sum_{i=1}^n \sum_{r=1}^n \sum_{j=1}^n \sum_{s=1}^n W_{ir,js} N_{ir} N_{js} + \sum_{i=1}^n \left(1 - \sum_{r=1}^n N_{ir} \right)^2. \quad (8)$$

In Eq. (8), we do not require the symmetrical property of \mathbf{N} , and in order to obtain the globally optimal, we let the combination cost of N_{ir} depend on the sum cost of all other elements N_{ir} . At the beginning, we expand the uniqueness constraint as

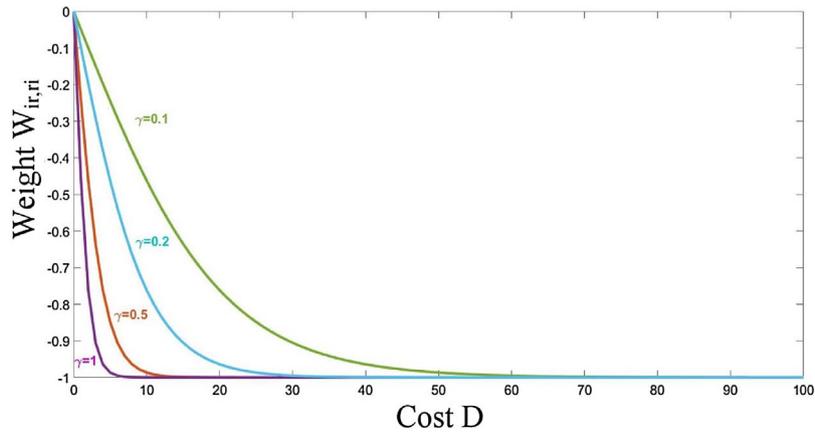


Fig. 3. Function of the weight $W_{ir,r}$ and the combination cost D .

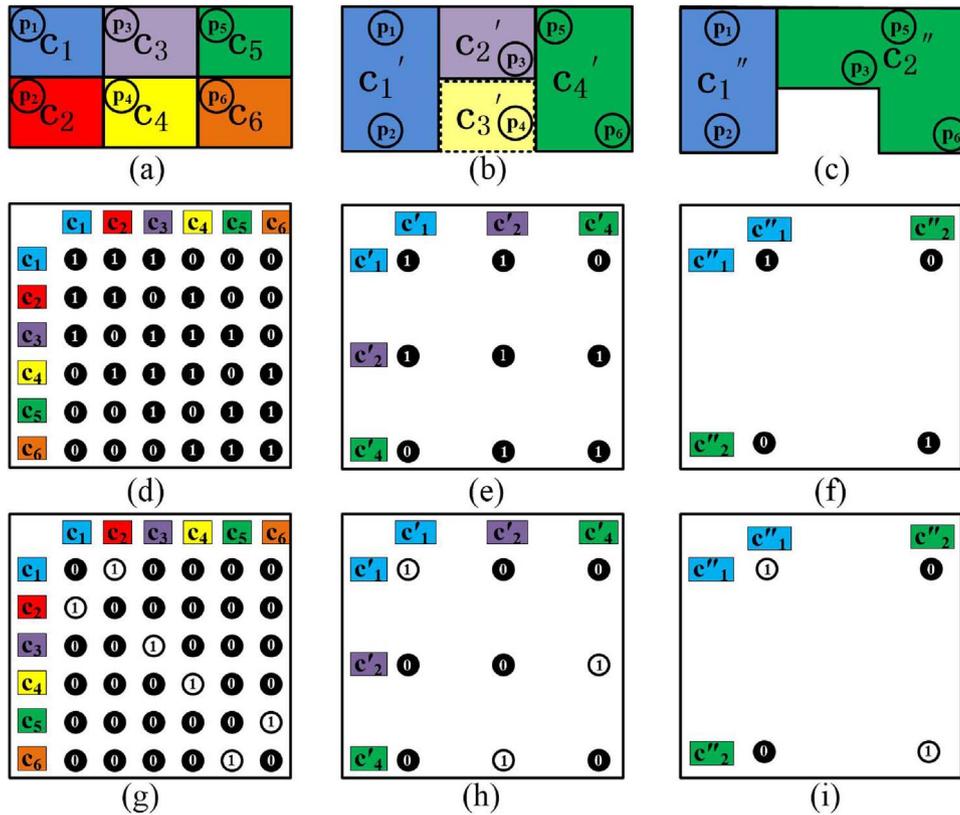


Fig. 4. Example of the proposed hierarchical clustering approach. (a)–(c) show the current cluster set C . (d)–(f) show the initial N calculated by the spatially neighborhood information in (a), (b) and (c), respectively. (g)–(i) show the achieved optimal combination solution for clusters in (a), (b) and (c), respectively.

$$\begin{aligned}
 \sum_{i=1}^n (1 - \sum_{r=1}^n N_{ir})^2 &= \sum_{i=1}^n (1 - 2 \times \sum_{r=1}^n N_{ir} + (\sum_{r=1}^n N_{ir})^2) \\
 &= \sum_{i=1}^n 1 - 2 \times \sum_{i=1}^n \sum_{r=1}^n N_{ir} + \sum_{i=1}^n (\sum_{r=1}^n N_{ir})^2 \\
 &= n - 2 \times \sum_{i=1}^n \sum_{r=1}^n N_{ir} + \sum_{i=1}^n \sum_{r=1}^n \sum_{s=1}^n N_{ir} N_{is} \\
 &= n - 2 \times \sum_{i=1}^n \sum_{r=1}^n N_{ir} \\
 &\quad + \sum_{i=1}^n \sum_{r=1}^n \sum_{j=1}^n \sum_{s=1}^n N_{ir} N_{js} \delta_{ij},
 \end{aligned}
 \tag{9}$$

where δ_{ij} is a binary value, i.e. if $i = j$, δ_{ij} is 1, otherwise δ_{ij} is 0. The trick in Eq. (9) is

$$\begin{aligned}
 (\sum_{r=1}^n N_{ir})^2 &= (N_{i1} + N_{i2} + \dots + N_{in}) \cdot (N_{i1} + N_{i2} + \dots + N_{in}) \\
 &= N_{i1}N_{i1} + N_{i1}N_{i2} + \dots + N_{i1}N_{in} + N_{i2}N_{i1} + \dots + N_{in}N_{in} \\
 &= \sum_{r=1}^n \sum_{s=1}^n N_{ir}N_{is}.
 \end{aligned}$$

Then, Eq. (8) is rewritten as

$$\begin{aligned}
 E &= - \sum_{i=1}^n \sum_{r=1}^n \sum_{j=1}^n \sum_{s=1}^n W_{ir,js} N_{ir} N_{js} + n - 2 \times \sum_{i=1}^n \sum_{r=1}^n N_{ir} \\
 &\quad + \sum_{i=1}^n \sum_{r=1}^n \sum_{j=1}^n \sum_{s=1}^n N_{ir} N_{js} \delta_{ij}.
 \end{aligned}$$

Let $E' = \frac{E-n}{2}$, $W'_{ir,js} = (W_{ir,js} - \delta_{ij})$, we have

$$E' = -\frac{1}{2} \sum_{i=1}^n \sum_{r=1}^n \sum_{j=1}^n \sum_{s=1}^n W'_{ir,js} N_{ir} N_{js} - \sum_{i=1}^n \sum_{r=1}^n N_{ir}.
 \tag{10}$$

In order to minimize Eq. (10), we are required deriving the update formulation to decrease the energy through the update process. Our minimization is based on the analysis of the energy change after the update process. In Eq. (10), if the value of an element in N is changed



Fig. 5. Imagery of the test scene @Google 2017 map data (48°51'01.3"N, 2°19'57.9"E). The red line is the manually generated vehicle trajectory based on the point density. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of the article.)

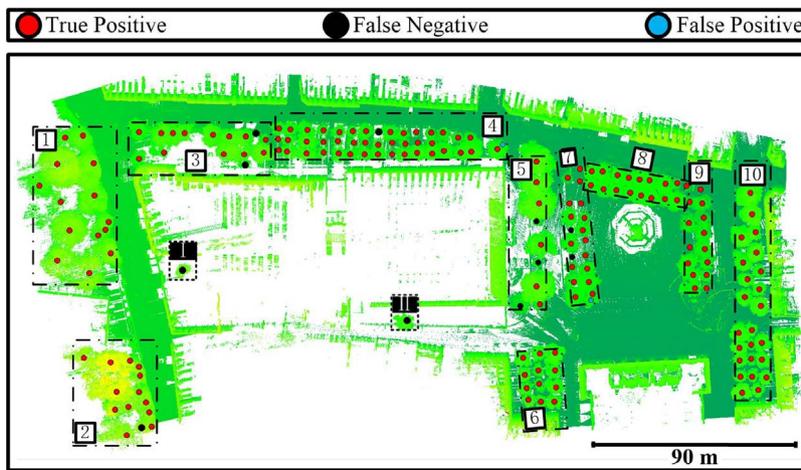


Fig. 6. Performance of the proposed nonphotosynthetic component extraction approach.

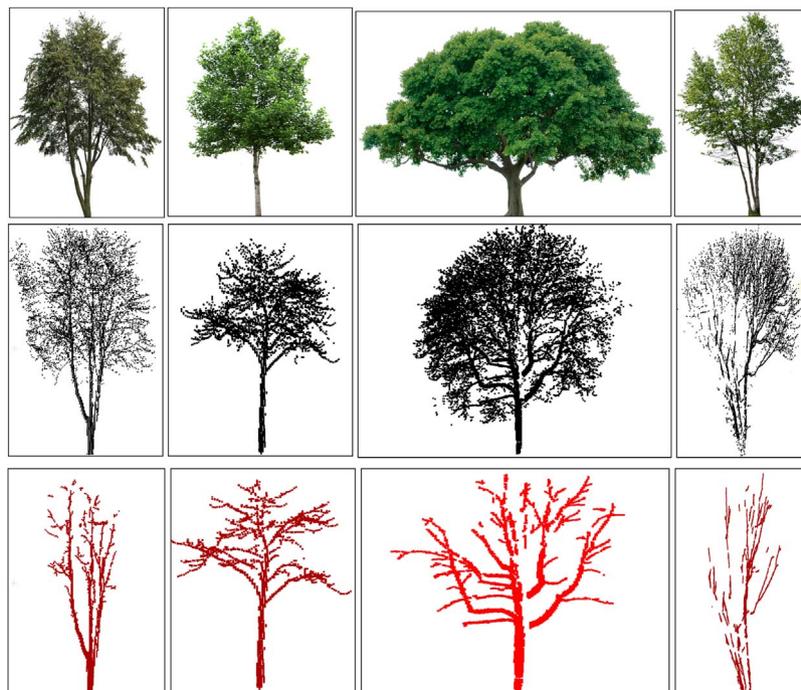


Fig. 7. Comparison between the reality and MLS point clouds of trees.

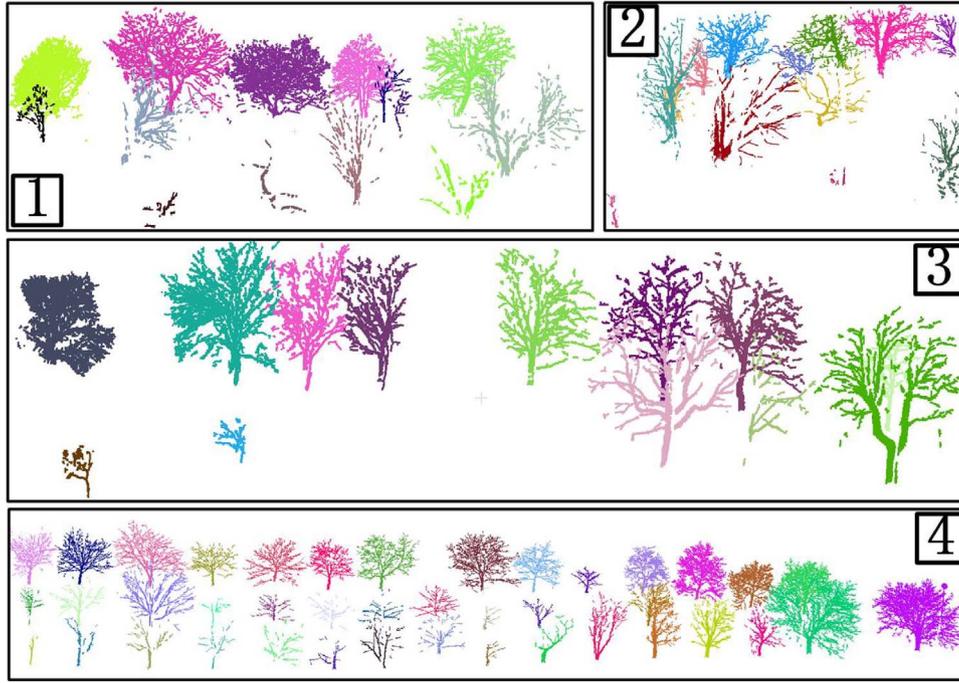


Fig. 8. Close-view of our performance on regions 1–4.

from N_{pq} to N'_{pq} , the output energy E_O will be

$$E_O = -\frac{1}{2} \sum_{i=1, i \neq p}^n \sum_{r=1, r \neq q}^n \sum_{j=1, j \neq p}^n \sum_{s=1, s \neq q}^n W'_{ir, js} N_{ir} N_{js} - \sum_{i=1, i \neq p}^n \sum_{r=1, r \neq q}^n N_{ir} - \frac{1}{2} \sum_{j=1, j \neq p}^n \sum_{s=1, s \neq q}^n W'_{pq, js} N'_{pq} N_{js} - \frac{1}{2} \sum_{i=1, i \neq p}^n \sum_{r=1, r \neq q}^n W'_{pq, js} N_{ir} N'_{pq} - \frac{1}{2} W'_{pq, pq} N'_{pq} N'_{pq} - N'_{pq}.$$

Since $W'_{pq, pq}$ is 0, we have

$$E_O = -\frac{1}{2} \sum_{i=1, i \neq p}^n \sum_{r=1, r \neq q}^n \sum_{j=1, j \neq p}^n \sum_{s=1, s \neq q}^n W'_{ir, js} N_{ir} N_{js} - \sum_{i=1, i \neq p}^n \sum_{r=1, r \neq q}^n N_{ir} - \sum_{j=1, j \neq p}^n \sum_{s=1, s \neq q}^n W'_{pq, js} N'_{pq} N_{js} - N'_{pq}. \quad (11)$$

The energy difference between the input E_I and the output E_O through the update process is calculated as

$$\begin{aligned} \Delta E &= E_O - E_I \\ &= - \left[\sum_{j=1, j \neq p}^n \sum_{s=1, s \neq q}^n W'_{pq, js} (N'_{pq} - N_{pq}) N_{js} \right] - (N'_{pq} - N_{pq}) \\ &= - \left[\sum_{j=1, j \neq p}^n \sum_{s=1, s \neq q}^n W'_{pq, js} N_{js} + 1 \right] (N'_{pq} - N_{pq}) \\ &= - \phi_{pq} \Delta N_{pq}. \end{aligned} \quad (12)$$

where ϕ_{pq} is the required update solution, which meets the condition that if $\phi_{mn} > 0$, $N'_{pq} = 1$, otherwise $N'_{pq} = 0$. Under this condition, (1) if $\Delta N_{pq} = 0$, ΔE will be 0; (2) if $\Delta N_{pq} > 0$, ΔE will be negative; (3) if $\Delta N_{pq} < 0$, ΔE will be also negative. Therefore, the energy will not be increased in all cases based on the provided update formulate as

$$\begin{cases} \left[\sum_{j=1}^n \sum_{s=1}^n W'_{ir, js} \cdot N_{js} + 1 \right] < 0, & N'_{ir} = 0; \\ \left[\sum_{j=1}^n \sum_{s=1}^n W'_{ir, js} \cdot N_{js} + 1 \right] > 0, & N'_{ir} = 1; \\ \left[\sum_{j=1}^n \sum_{s=1}^n W'_{ir, js} \cdot N_{js} + 1 \right] = 0, & N'_{ir} = N_{ir}. \end{cases} \quad (13)$$

In the implementation of our hierarchical clustering, in order to keep the symmetrical property of N , when we update N_{ir} , we will also update N_{ri} as N_{ir} . The update criterion is based on the calculation of $\left[\sum_{j=1}^n \sum_{s=1}^n W'_{ir, js} \cdot N_{js} + 1 \right]$. The initial N is based on the spatial

neighborhood information, i.e. if the cluster c_i is close to c_r (< 0.2 m), N_{c_i, c_r} will be assigned as 1. In the update process, for each element N_{ir} , we will update it based on the proposed criterion. After we update all elements in N , we start to combine clusters using the updated N . If the (i, r) th element N_{ir} is 1, combine c_i and c_j into one cluster. To converge the clustering process, we set a cutoff distance threshold as the diagonal element of PM , e.g. 0.2 in our work. If the combination cost is larger than the threshold, the cluster will be combined with itself, which means the convergence. The update process conducted iteratively until the cluster set C converges. The number of final clusters is inversely proportional to the cutoff distance. The computational complexity of each update process depends on the size of N which is $O(n^2)$

4.2. Implementation of the proposed hierarchical clustering approach

Key steps in the implementation of the proposed hierarchical clustering include (1) computing the proximity matrix PM using Eq. (4), (2) removing clusters that are not from nonphotosynthetic components, e.g. those clusters in a large planar area, (3) updating the value of elements in N based on the proposed criterion and combining clusters using the updated N .

The following is a simulated example to show the implementation of the hierarchical clustering based on the optimal combination. As shown in Fig. 4(a), the input P is $\{p_1, p_2, p_3, p_4, p_5, p_6\}$.

- Step 1: initialize the cluster set C as $\{c_1, c_2, c_3, c_4, c_5, c_6\}$, where $c_1 = \{p_1\}$, $c_2 = \{p_2\}$, $c_3 = \{p_3\}$, $c_4 = \{p_4\}$, $c_5 = \{p_5\}$ and $c_6 = \{p_6\}$;
- Step 2: initialize elements in $N_{6 \times 6}$ based on the neighborhood information as shown in Fig. 4(d);
- Step 3: calculate the proximity matrix PM for the current cluster set C ;
- Step 4: update elements in N based on the proposed criterion to obtain the optimal combination as shown in Fig. 4(g);
- Step 5: from Fig. 4(g), we know that $N_{12}, N_{21}, N_{33}, N_{44}, N_{56}, N_{65}$ are 1, therefore, combine clusters c_1 and c_2 as a new cluster c'_1 , and combine c_5 and c_6 as a new cluster c'_4 . The cluster c_3 and c_4 are unchanged and renamed as c'_2 and c'_3 , respectively.
- Step 6: update the cluster set C as $\{c'_1, c'_2, c'_3, c'_4\}$;

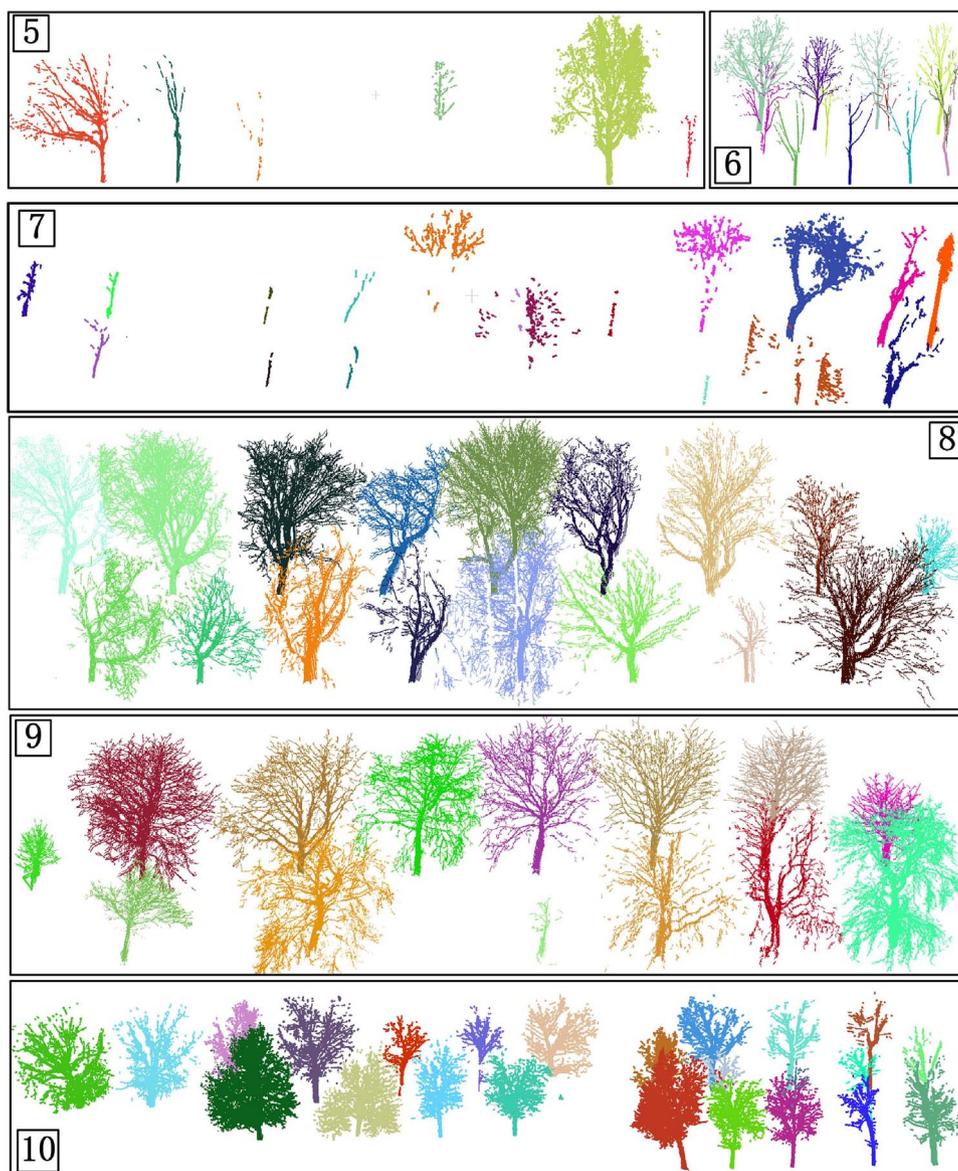


Fig. 9. Close-view of our performance on regions 5–10.

Now the first iteration is done. Repeat steps 2–6. Fig. 4(e) and (f) shows the initial N corresponding to Fig. 4(b) and (c), respectively. Fig. 4(h) and (i) are the achieved optimal combination solution in the second and third iteration, respectively. In Fig. 4(b), c_3' is removed because it is from a large planar area, which is regarded as not the nonphotosynthetic component. Finally C converges to $\{c_1'', c_2''\}$ as solved in Fig. 4(i). The cluster c_1'' and c_2'' are regarded as two complete nonphotosynthetic components.

5. Experiments and evaluations

5.1. Performance of the nonphotosynthetic component extraction

This section aims to evaluate the performance of the nonphotosynthetic component extraction based on the proposed hierarchical clustering approach. The test urban scene is located in Paris collected by MLS system in January 2013 and the imagery of the test scene is shown in Fig. 5. The size of the test data is 173 m by 352 m containing 49,512,718 points. Details of the data collection description are shown in Vallet et al. (2015).

Our nonphotosynthetic component extraction results are shown in

Fig. 6. TP (true positive) means that a nonphotosynthetic component is extracted correctly from the input scene. FN (false negative) means that a nonphotosynthetic component is wrongly detected as the background points. FP (false positive) means that a background cluster is wrongly extracted as a nonphotosynthetic component. The ground truth of nonphotosynthetic components for the reference is obtained manually from the input scene. We segment the nonphotosynthetic component of each individual tree attentively through the point cloud visualization tool CloudCompare (<http://www.danielgm.net/cc/>). We add some photographs of individual trees as reference for the better comparison between reality and MLS point cloud as shown in Fig. 7. The first row shows the photographs of real trees, the second row shows their MLS point clouds and the third row shows the manually segmented nonphotosynthetic components.

In order to show the close-view of our extraction results, we demonstrate them in 10 regions as shown in Figs. 8:1–4 and 9 :5–10. In our results, if street trees are close to the vehicle LiDAR sensor (< 10 m), trunks and branches of a tree will be completely extracted as shown in regions 1, 2, 3, 4, 8, 9, 10. However, if trees are far from the sensor (> 30 m), extraction results may only contain the main stem and branches of a tree due to the sparsity of tree points as shown in regions

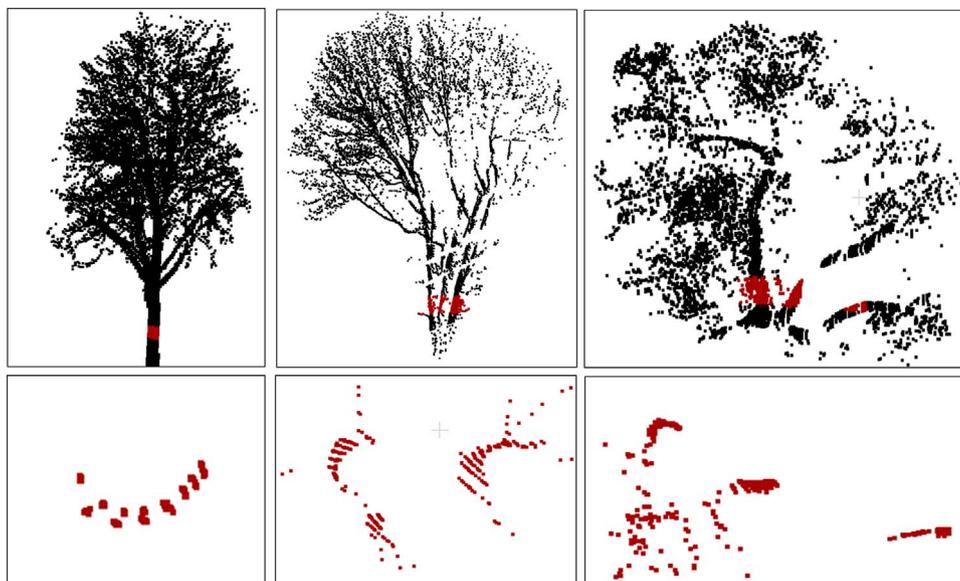


Fig. 10. The cross-section of individual trees.

5, 6, 7.

The close-view of TP is shown in Fig. 11. In the experimental scene, trees that are less than 50 m from the sensor can be extracted effectively with the proposed extraction algorithm. The algorithm performs well on those common cases of street trees as shown in Fig. 11(a)–(c). In the extraction, the challenge #1 is that a tree contains a short main stem as shown in Fig. 11(e)–(h). In Fig. 11(g) and (l), an individual tree seems contain several stems in point clouds, which is because their short roots are occluded by the low elevation objects. This challenge is difficult for methods based on the cylinder fitting (Liang et al., 2014; Pfeifer et al., 2004; Raumonon et al., 2013). The ideal cross-section of an individual tree stem is represented as the half-moon shape in 3D point clouds, which can be fitted by using the cylinder model as shown in the first column of Fig. 10. However, in the challenge #1, it is difficult to set a priori model for those stems as shown in the second and third columns of Fig. 10. The challenge #2 is that stems are covered by dense leaves as shown in Fig. 11(d), (i) and (j). This challenge is addressed by our direction term, which aims to analyze the principal direction of current dense points. The challenge #3 is that a tree contains a thin stem as shown in Fig. 11(k). In this challenge, although we miss some branches which are close to treetops, tree stems are extracted accurately. The challenge #4 is the split of nonphotosynthetic components from overlapping trees as shown in Fig. 11(i), which is solved effectively by the proposed extraction algorithm.

It is worth noting that as shown in Fig. 9, the input data contain trees with parallel branches. This is because trees are scanned for several times in the data collection, which means that there is a registration error in the MLS point cloud. Those unregistered trunks will also be extracted as nonphotosynthetic components of trees, because these points are in a high density and have consistent principal direction. Although the inaccurate registration will cause errors in the estimation of the diameter at breast height of trees, the number of nonphotosynthetic components in the input data will not be affected.

The close-view of FN is shown in Fig. 12. Fig. 12(a) and (b) are from the region 3 and 4, respectively. The missing tree in Fig. 12(a) is caused by its extremely low elevation (< 2 m). The missing tree in Fig. 12(a) and (c) are caused by that their roots are occluded completely and the rest of tree points do not contain the sufficient principal direction information. When the incompleteness of a tree stem is larger than 80%, we will miss it in the extraction.

The close-view of FP is shown in Fig. 13. Fig. 13(a) and (b) are from the region 4 and 7, respectively. Errors caused by the mix of the

streetlight and branches. Streetlights in Fig. 13 are extracted as nonphotosynthetic components in our result as shown in Figs. 8:4 and 9:7.

To evaluate our extraction results, we calculate the correctness r , completeness p and F -score f as

$$r = \frac{TP}{TP + FP}, \quad p = \frac{TP}{TP + FN}, \quad f = \frac{2 \times TP}{2 \times TP + FP + FN}. \quad (14)$$

The correctness measures the ratio of correctly extracted nonphotosynthetic components in results, the completeness measures the percentage of correctly extracted nonphotosynthetic components in the reference. F -score is the harmonic mean of correctness and completeness. The evaluation of our performance on each region is shown in Table 1. We extract all nonphotosynthetic components successfully in regions 1, 6, 8, 9 and 10. Except for FP results in regions 4 and 7, our extracted nonphotosynthetic components are all accurate. There are 182 trees in the test scene and we extract 172 individual nonphotosynthetic components from the input MLS data. The average correctness, completeness and F -score are 98.9%, 94.0% and 0.96, respectively.

The above-mentioned extraction were done on a Windows 10 Home 64-bit, Intel Core i7-4790 3.6GHz processor with 16 GB of RAM and computations were carried on Matlab R2017b. It took us around 87 min to finish the entire extraction of 172 out of 182 nonphotosynthetic components from MLS data. The disadvantage of the extraction algorithm is the high memory complexity ($O(n^2)$), which can be improved by sparse storage techniques.

5.2. Performance of the stem detection

After we obtained the nonphotosynthetic components from input data, we can detect the main stem of each individual component based on the consistency of the growing direction. We perform the Euclidean clustering on each extracted nonphotosynthetic component and then generate a voxel for each cluster. After this, we calculate the principal direction between each two neighbor voxels. Once there exist paths with different directions ($> 30^\circ$) and both paths are longer than 0.5 m, we extract points between the split point and the ground point as the stem points. Take the nonphotosynthetic component in Fig. 14(a) as an example, Fig. 14(b) shows the result of the Euclidean clustering on component points, and Fig. 14(c) shows the generated voxels for each cluster and the obtained split point based on the path direction.

Results of our stem detection are shown in Fig. 15. We detect 164

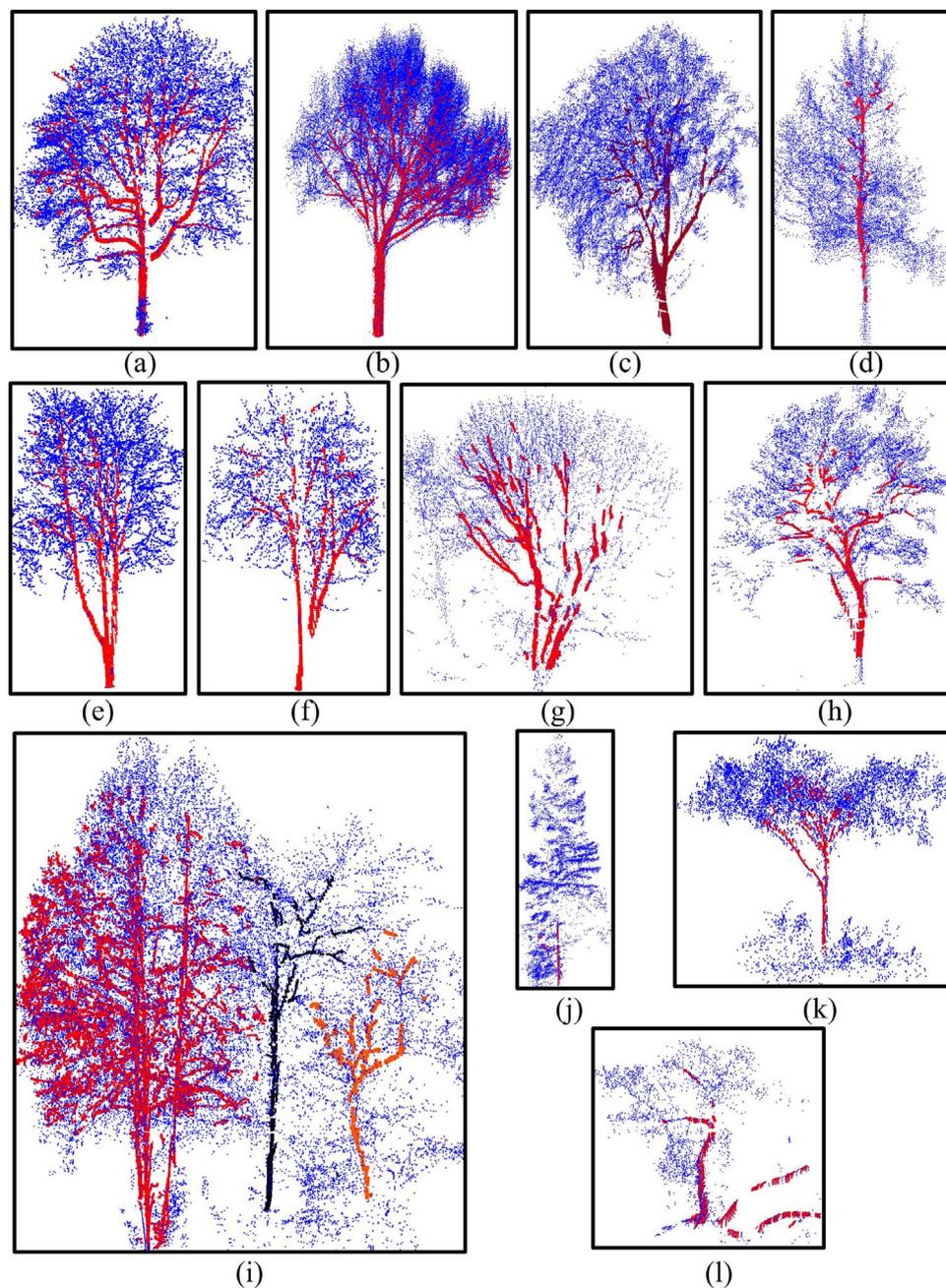


Fig. 11. The close-view of the true positive results in the extraction.

out of 172 individual stems as shown in Fig. 15(a). Fig. 15(b) is the close-view of the region A in Fig. 15(a), which is a mix of trees with different sizes and shapes. Fig. 15(c) is the close-view of the region B in Fig. 15(a), which mostly contains common street trees. Fig. 15(d) is the close-view of the region C in Fig. 15(a), which mainly contains thin stems.

In order to demonstrate the superiority of the proposed method on stem detection, we show the comparison with other methods, including Lehtomaki10 (Lehtomäki et al., 2010), Hetti13 (Hetti Arachchige, 2013), Liang14 (Liang et al., 2014), Xia (Xia et al., 2015) and Xu (Xu et al., 2018). Table 2 shows the description of datasets and test scenes in the mentioned stem detection methods.

Our stem detection is based on the nonphotosynthetic components, therefore, low robustness caused by scatter points around stems in Lehtomäki et al. (2010) will not appear in our detection. The split point is detected based on the path direction of the generated voxels, which is

insensitive to the problem in Hetti Arachchige (2013) caused by the small break or hole on stems. Besides, our stem detection does not rely on the shape fitting used in Liang et al. (2014), Xu et al. (2018) which fails to work well when stems are not naturally separated or do not stand vertically. In the implementation, the proposed stem detection does not require a classification process as in Xia et al. (2015) and works automatically in the complex urban environment. Quantitative comparison is shown in Table 3, we achieve the highest correctness, completeness and *F*-score in our experimental scene.

5.3. Performance of the individual tree segmentation

For each nonphotosynthetic component, the leaf region is defined as a cylinder, which ranges between the split point achieved in the stem detection and the top of its nonphotosynthetic component. The axis of the cylinder is the line fitted by the stem points, and the radius is the

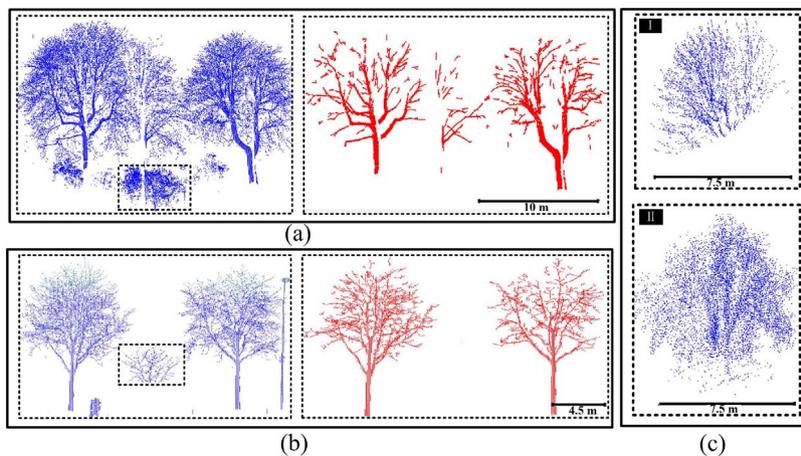


Fig. 12. The close-view of the false negative results in the extraction.

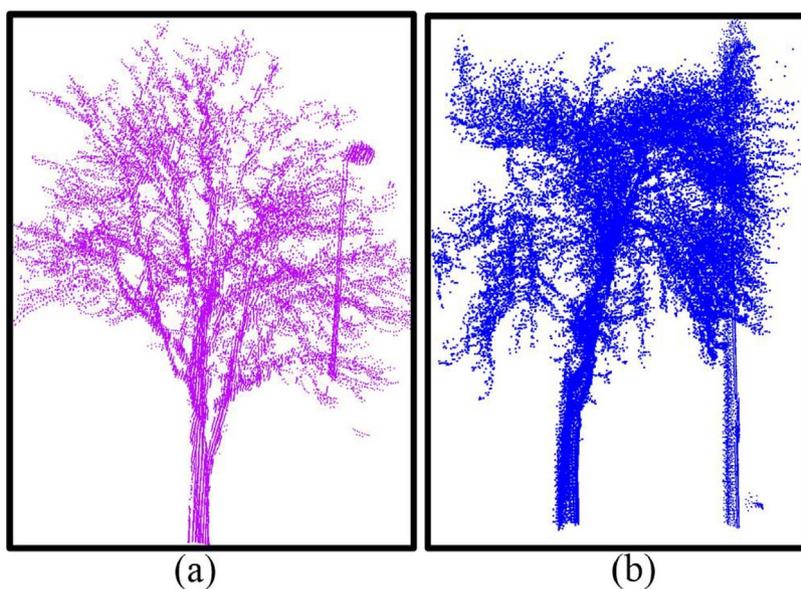


Fig. 13. The close-view of the false positive results in the extraction.

Table 1
Evaluation of the extracted nonphotosynthetic component.

Region	Reference	TP	FN	FP	r (%)	p (%)	f
1	14	14	0	0	100	100	1
2	14	13	1	0	100	92.9	0.96
3	15	13	2	0	100	86.7	0.93
4	43	42	1	1	97.7	97.7	0.98
5	9	6	3	0	100	66.7	0.80
6	12	12	0	0	100	100	1
7	20	17	2	1	94.4	89.5	0.92
8	17	17	0	0	100	100	1
9	14	14	0	0	100	100	1
10	24	24	0	0	100	100	1
Total	182	172	9+2	2	98.9	94.0	0.96

furthest distance between the nonphotosynthetic component points and the axis line as shown in Fig. 16(a). In the individual tree segmentation, we first find all leaf regions, and then assign leaf region points to their closest nonphotosynthetic component as shown in Fig. 16(b).

Results of our individual tree segmentation based on the extracted nonphotosynthetic components are shown in Fig. 17. One disadvantage is that artifact objects mixed in leaf regions will be regarded as tree crown points.

In order to demonstrate the superiority of the proposed method on individual tree segmentation, we compare our performance with other segmentation methods, including Pu11 (Pu et al., 2011), Wu13 (Wu et al., 2013), Fan16 (Fan et al., 2016), Tao15 (Tao et al., 2015), Li16 (Li et al., 2016) and Zhong17 (Zhong et al., 2017). Table 4 shows the description of datasets and test scenes in the mentioned individual tree segmentation methods.

Our individual tree segmentation is based on the nonphotosynthetic component extraction results which succeed to obtain those curved trunks. This is difficult for methods (Tao et al., 2015; Li et al., 2016), which mainly proposed for trees growing vertically. As shown in Fig. 17, our test scene is much more complex than (Fan et al., 2016), which shows the high applicability of the proposed segmentation algorithm for the mix of streets trees. The assignment of leaf points is based on the Euclidean distance rule rather than the statistic analysis of points which depends on the point density as in Zhong et al. (2017). In the implementation, we do not require the complex parameter setting and much priori criteria as in the knowledge-based method (Pu et al., 2011). Besides, we do not have to generate voxels for all point clouds as in Wu et al. (2013), which is easy to be time-consuming and lose details of tree branches in the segmentation.

Quantitative comparison is shown in Table 5, Wu13 and Tao12 achieve a higher correctness and completeness than us, this is because

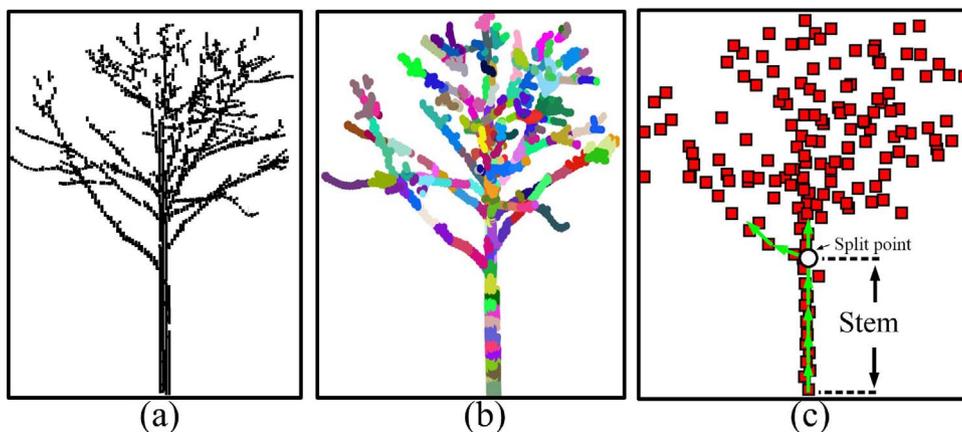


Fig. 14. Detection of the split point. (a) Extracted nonphotosynthetic component points. (b) Euclidean clustering of (a). (c) Generated voxels based on (b).

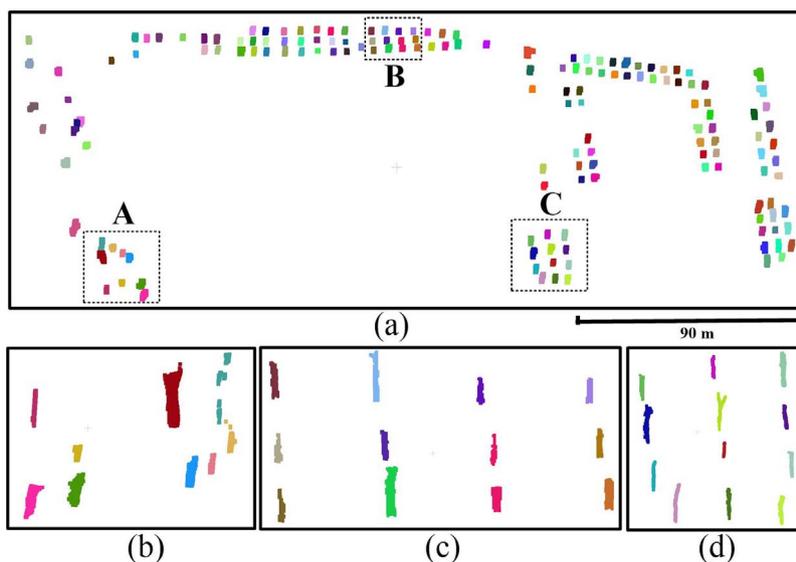


Fig. 15. Performance of the stem detection based on the nonphotosynthetic component extraction results.

Table 2
Description of datasets and test scenes in the stem detection.

Algorithms	Dataset		Test scene			
	Scanner	Density (points/m ²)	Environment	Type of trees	Number of trees	Time of acquisition
Lehtomaki10	MLS	/	Suburban area	/	79	/
Hetti13	MLS	/	Urban area	Mixed	42	/
Liang14	MLS	/	Mixed forest	Spruce, birch	80	/
Xia15	TLS	1000	Homogeneous forest	Bamboo	166	11/2013
Xu2018	MLS	750	Residential area	Mixed	129	10/2015
Proposed	MLS	700	Urban area	Mixed	182	01/2013

Table 3
Comparison with other stem detection methods proposed for ground-based LiDAR.

Algorithm	Lehtomaki10	Hetti13	Liang14	Xia15	Xu2018	Proposed
Evaluation	<i>r</i> 86.5	97.5	/	93.0	95.7	98.80
	<i>p</i> 83.5	92.5	87.5	88.0	94.2	95.34
	<i>f</i> 0.85	0.95	/	0.90	0.95	0.97

that most test trees in their work are in a similar shape and have a large diameter at breast height, e.g. the plane tree and camphor tree in Wu13 (Wu et al., 2013), and the pecan and pinus in Tao15 (Tao et al., 2015). In the comparison of the completeness, the accuracy of Li16 (Li et al.,

2016) and Zhong17 (Zhong et al., 2017) is a little higher than us, this is because that our missing trees are almost occluded completely in MLS data as discussed in the FN nonphotosynthetic component extraction in Fig. 12.

6. Conclusions

This paper investigates an automatic cluster-based algorithm for trees' nonphotosynthetic components extraction from MLS data. In the extraction, we do not require initializing any priori knowledge of input trees, e.g. the number, location and shape. The combination of non-photosynthetic component points is globally optimized by minimizing the proposed energy function based on the Euclidean distance and

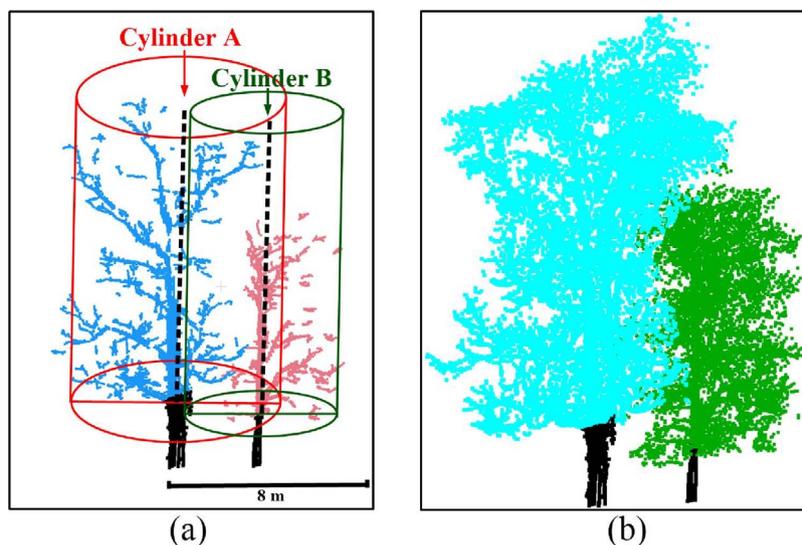


Fig. 16. The detection and assignment of leaf points.

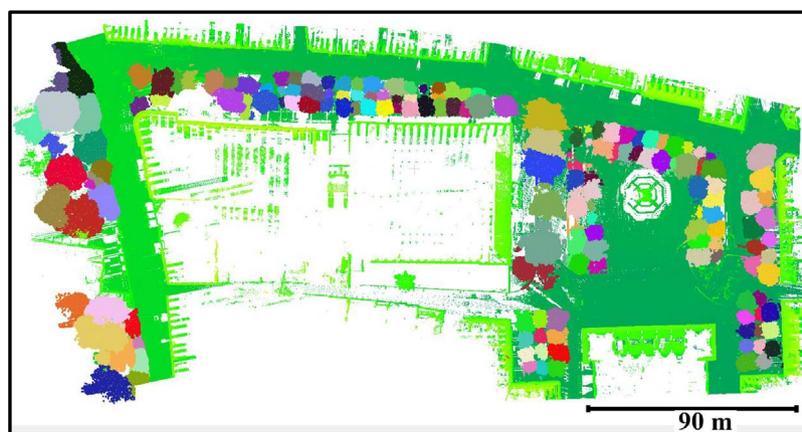


Fig. 17. Performance of the individual tree segmentation based on the nonphotosynthetic component extraction results.

Table 4
Description of datasets and test scenes in the individual tree segmentation.

Algorithms	Dataset		Test scene			
	Scanner	Density (points/m ²)	Environment	Type of trees	Number of trees	Time of acquisition
Pu11_1	MLS	/	urban area	mixed	33	12/2008
Pu11_2	MLS	/	urban area	mixed	84	06/2009
Wu13_1	MLS	212	residential area	planetree	72	03/2012
Wu13_2	MLS	224	residential area	camphortree	68	12/2012
Fan16	MLS	/	urban area	palmtree	311	/
Tao15_1	TLS	/	garden	pecan	14	02/2014
						09/2014
Tao15_2	TLS	/	homogeneous forest	pinus	51	12/2014
Tao15_3	TLS	/	homogeneous forest		49	01/2015
Li16_1	MLS	123	urban area	mixed	66	/
Li16_2	MLS	306	urban area	mixed	29	/
Zhong17_1	TLS	2821	University		/	01/2015
Zhong17_2	MLS	222	urban area		/	06/2011
Proposed	MLS	700	urban area	mixed	182	01/2013

principal direction. The proposed algorithm succeeds to extract non-photosynthetic components from a complex urban environment. Performances of the stem detection and individual tree segmentation based on the extracted nonphotosynthetic components are competitive against the state-of-the-art methods in terms of the correctness, completeness and *F*-score. Future work will focus on the nonphotosynthetic

component extraction from different terrain areas where tree roots are difficult to be found.

Funding information

National Key Research and Development Plan of China

Table 5
Comparison with other individual tree segmentation methods proposed for ground-based LiDAR.

Algorithm		Pu2011_1	Pu2011_2	Wu13_1	Wu13_2	Fan15	Tao15_1	Tao15_2
Evaluation	<i>r</i>	84.6	85.7	100	100	89.0	100	100
	<i>p</i>	63.5	29.5	100	98.5	86.2	100	100
	<i>f</i>	0.73	0.44	1	0.99	0.88		1
Algorithm		Tao15_3	Li16_1	Li16_2	Zhong17_1	Zhong17_2	Proposed	/
Evaluation	<i>r</i>	92.0	98.4	97.4	92.4	94.0	98.9	/
	<i>p</i>	100	98.2	96.8	95.4	93.7	94.0	/
	<i>f</i>	0.96	0.98	0.97	0.94	0.94	0.96	/

(2016YFD0600101), National Natural Science Foundation of China (31770591, 41701510), China Postdoctoral Science Foundation (2016M601823).

References

- Edson, C., Wing, M.G., 2011. Airborne light detection and ranging (LiDAR) for individual tree stem location, height, and biomass measurements. *Remote Sens.* 3 (11), 2494–2528.
- Fan, W., Chenglu, W., Jonathan, L., 2016. Automated extraction of urban trees from mobile LiDAR point clouds. In: 2nd ISPRS International Conference on Computer Vision in Remote Sensing (CVRS 2015), vol. 9901. International Society for Optics and Photonics. pp. 99010P.
- Hetti Arachchige, N., 2013. Automatic tree stem detection—a geometric feature based approach for MLS point clouds. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* (2), 109–114.
- Korpela, I., Ørka, H.O., Maltamo, M., Tokola, T., Hyypä, J., et al., 2010. Tree species classification using airborne LiDAR-effects of stand and tree parameters, downsizing of training set, intensity normalization, and sensor type. *Silva Fennica* 44 (2), 319–339.
- Kwak, D.-A., Lee, W.-K., Lee, J.-H., Biging, G.S., Gong, P., 2007. Detection of individual trees and estimation of tree height using LiDAR data. *J. Forest Res.* 12 (6), 425–434.
- Lehtomäki, M., Jaakkola, A., Hyypä, J., Kukko, A., Kaartinen, H., 2010. Detection of vertical pole-like objects in a road environment using vehicle-based laser scanning data. *Remote Sens.* 2 (3), 641–664.
- Li, L., Li, D., Zhu, H., Li, Y., 2016. A dual growing method for the automatic extraction of individual trees from mobile laser scanning data. *ISPRS J. Photogramm. Remote Sens.* 120, 37–52.
- Liang, X., Hyypä, J., Kukko, A., Kaartinen, H., Jaakkola, A., Yu, X., 2014. The use of a mobile laser scanning system for mapping large forest plots. *IEEE Geosci. Remote Sens. Lett.* 11 (9), 1504–1508.
- Pfeifer, N., Gorte, B., Winterhalder, D., et al., 2004. Automatic reconstruction of single trees from terrestrial laser scanner data. In: Proceedings of 20th ISPRS Congress. ISPRS Istanbul. pp. 114–119.
- Pu, S., Rutzinger, M., Vosselman, G., Elberink, S.O., 2011. Recognizing basic structures from mobile laser scanning data for road inventory studies. *ISPRS J. Photogramm. Remote Sens.* 66 (6), S28–S39.
- Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., Holopainen, M., Disney, M., Lewis, P., 2013. Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sens.* 5 (2), 491–520.
- Rusu, R.B., Marton, Z.C., Blodow, N., Dolha, M., Beetz, M., 2008. Towards 3D point cloud based object maps for household environments. *Robot. Auton. Syst.* 56 (11), 927–941.
- Tao, S., Wu, F., Guo, Q., Wang, Y., Li, W., Xue, B., Hu, X., Li, P., Tian, D., Li, C., et al., 2015. Segmenting tree crowns from terrestrial and mobile LiDAR data by exploring ecological theories. *ISPRS J. Photogramm. Remote Sens.* 110, 66–76.
- Vallet, B., Brédif, M., Serna, A., Marcotegui, B., Paparoditis, N., 2015. TerraMobilita/iQmulus urban point cloud analysis benchmark. *Comput. Graphics* 49, 126–133.
- Wu, B., Yu, B., Yue, W., Shu, S., Tan, W., Hu, C., Huang, Y., Wu, J., Liu, H., 2013. A voxel-based method for automated identification and morphological parameters estimation of individual street trees from mobile laser scanning data. *Remote Sens.* 5 (2), 584–611.
- Xia, S., Wang, C., Pan, F., Xi, X., Zeng, H., Liu, H., 2015. Detecting stems in dense and homogeneous forest using single-scan TLS. *Forests* 6 (11), 3923–3945.
- Xu, S., Xu, S., Ye, N., Zhu, F., 2018. Individual stem detection in residential environments with MLS data. *Remote Sens. Lett.* 9 (1), 51–60.
- Yao, W., Krzystek, P., Heurich, M., 2012. Tree species classification and estimation of stem volume and DBH based on single tree extraction by exploiting airborne full-waveform LiDAR data. *Remote Sens. Environ.* 123, 368–380.
- Yun, T., An, F., Li, W., Sun, Y., Cao, L., Xue, L., 2016. A novel approach for retrieving tree leaf area from ground-based LiDAR. *Remote Sens.* 8 (11), 942.
- Zhong, L., Cheng, L., Xu, H., Wu, Y., Chen, Y., Li, M., 2017. Segmentation of individual trees from TLS and MLS data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 10 (2), 774–787.