ISPRS Journal of Photogrammetry and Remote Sensing 174 (2021) 19–34

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

Automated street tree inventory using mobile LiDAR point clouds based on Hough transform and active contours

Amir Hossein Safaie [a], Heidar Rastiveis [a,*], Alireza Shams [b], Wayne A. Sarasua [c], Jonathan Li [d]

[a] Department of Photogrammetry and Remote Sensing, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran
[b] Department of Architecture and Construction Management Farmingdale State College, State University of New York, Farmingdale, NY, USA
[c] Glenn Department of Civil Engineering, Clemson University, Clemson, SC, USA
[d] Department of Geography and Environmental Management, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

ARTICLE INFO

Keywords:
Trees inventory
Mobile LiDAR
Point clouds
Hough transform
Active contour
Road safety

ABSTRACT

Trees are important road-side objects, and their geometric information plays an essential role in road studies and safety analyses. This paper proposes an efficient method for the automated creation of a road-side tree inventory using Mobile Terrestrial Lidar System (MTLS) point clouds. In the proposed method ground points are filtered through preprocessing to reduce processing time. Next, tree trunks are detected by performing a Hough Transform (HT) algorithm on several generated raster images from the point clouds. By initiating an approximate area of a tree's foliage through a Voronoi Tessellation (VT) algorithm, the accurate boundary of the foliage is identified by applying Active Contour (AC) models. By extracting the points within this foliage boundary the geometric characteristics of each tree are obtained. This method was evaluated with two sample point clouds from different MTLS systems, and the algorithm correctly extracted all of the trees from both datasets. Additionally, comparing the calculated parameters with manually observed measures, the accuracy of the obtained geometric parameters were promising.

1. Introduction

Trees are a common sight along urban and intercity roads, and their inventory is useful for roadway maintenance and ensuring road user safety. Trees help to minimize erosion and provide natural beauty to the roadside. They can also serve as a natural snow fence. On the other hand, trees can be a hazard to motorists. A tree with a diameter of as little as 10.2 cm (4 in.) can impale a vehicle (Eck and McGee, 2008). In the United States, roughly ¼ of fixed-object fatal crashes involve trees and over 90% of tree fatal crashes are on two-lane roads that have lower design standards than freeways (Sebastian and Flemons, 1981).

Trees are composed of two main components shown in Fig. 1. The lowest part of a tree above ground is called the trunk, and the portion of the tree from the top of the trunk to the highest tree point is known as foliage (Yue et al., 2015). Geometric parameters of trees include trunk height, trunk diameter, foliage height, foliage diameter, and tree height. Trunk diameter and tree height are especially useful in road safety analysis (Bendigeri et al., 2011; Eck and McGee, 2008).

Roads typically have an established clear zone that serves as an unobstructed, traversable roadside area that allows a driver to stop safely, or regain control of a vehicle that has left the roadway (Safety, 2011). The width of the clear zone varies by roadway type and is influenced by horizontal alignment, traffic volumes, speeds, and slopes (Safety, 2011). Having an efficient method to accurately identify, map, and categorize trees within a roadside clear zone can potentially lead to improved safety. Since traditional ground surveying techniques for collecting geometric information of objects are time-consuming, tedious, and expensive, remote sensing data such as aerial imagery or Light Detection and Ranging (LiDAR) point clouds are useful resources for creating a tree inventory. Automated tree extraction along roadway corridors is more from a safety standpoint or to assist in managing landscaping within a roadway's right-of-way (ROW) (Eck and McGee, 2008).

Mobile Terrestrial Laser Scanning (MTLS) systems are becoming more prominent in road environment modeling because they are capable of obtaining accurate 3D information safely and efficiently.

* Corresponding author at: Department of Photogrammetry and Remote Sensing, School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran.
E-mail addresses: amirsafaee90@ut.ac.ir (A.H. Safaie), hrasti@ut.ac.ir (H. Rastiveis), shamsa@farmingdale.edu (A. Shams), sarasua@clemson.edu (W.A. Sarasua), junli@uwaterloo.ca (J. Li).

https://doi.org/10.1016/j.isprsjprs.2021.01.026
Received 12 August 2020; Received in revised form 7 January 2021; Accepted 28 January 2021

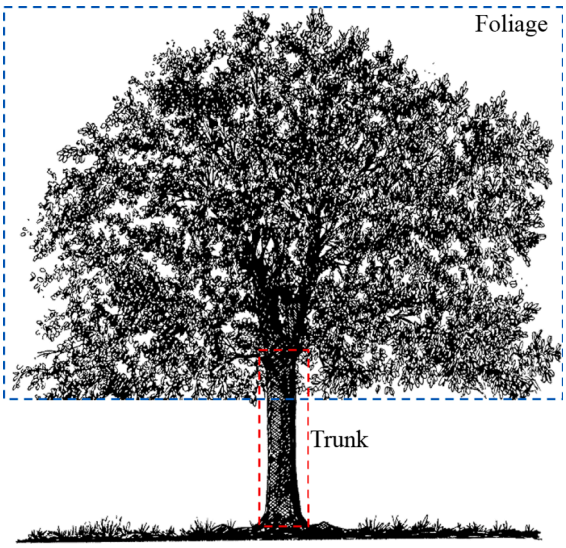

Fig. 1. Tree's main components including trunk and foliage.

They use laser pulses to collect the coordinates of 3D points (up to or exceeding a million points per second) to quickly create dense point clouds. An MTLS is ideally suited for this purpose. An MTLS includes integration of several devices including a laser scanner, a Global Navigation Satellite System (GNSS), an Inertial Measurement Unit (IMU), high-resolution cameras, and an on-board computer system (Kumar et al., 2017). Due to the proven capability and functionality of MTLS systems in road environment modeling, this technology has been widely used to extract marginal road information such as pole-shape objects (Cabo et al., 2014; Shokri et al., 2019), traffic signs (Yang et al., 2013), lane markings (Rastiveis et al., 2020), highway cross slope (Shams et al., 2018), and other roadside attributes (Zaboli et al., 2019). Since this paper aims to propose a new method for tree inventory using this data, we will focus on research studies where MTLS point clouds were used to extract trees or their parameters which are discussed in the following sub-section.

1.1. Related works

Generally, tree inventory methods using MTLS point clouds can be categorized into two groups: 1) manual; and 2) automated. In manual methods, trees are visually extracted from the point clouds with the help of point cloud processing software, and the parameters of the trees are then measured by an expert. These methods are usually time-consuming and labor-intensive. Because of this, it was difficult to find research studies based on manual methods for extracting trees from MTLS point clouds. One study by Holopainen et al. (2013) detected the tree's location manually to compare with an automatic method. The obtained Root Mean Square Error (RMSE) in the manual method was reported to be 0.49 m which was similar to the automatic method. However, they correctly detected about 80% of the trees manually compared to 27% using their automated method.

Although the reliability of manual methods is undeniable, recent research has recently focused on developing automated methods to save cost and time to create tree inventories. Over the last decade, various methods for automated tree extraction from LiDAR point clouds have been developed. In these methods, tree location and its parameters are extracted using data processing techniques developed via computer programming. These methods can be classified into two categories based on the strategy to handle the data: point-based methods, in which the points are directly imported into the algorithms, and grid-based methods that work based on 2D or 3D grid networks generated from the points.

Point-based tree extraction methods have been used by many researchers for tree extraction. For example, Othmani et al. (2011) proposed a primary algorithm for automatic tree detection and Diameter at Breast Height (DBH) computation from point clouds. Their proposed method had five primary main steps: (1) terrain extraction; (2) points clustering; (3) virtual logs creation; (4) skeletonization; and (5) stem mapping and DBH computation. Although their method was developed for Fixed Terrestrial Laser Scanner (FTLS) point clouds, it has since been applied to MTLS point clouds in more recent research (Bauwens et al., 2016; Stal et al., 2020). Bauwens et al. (2016) compared measured DBH from the point clouds with field measured DBH, and reported an RMSE of 1.11 cm."

Yu et al. (2012) used the Bayesian paradigm based on a Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm to optimize tree extraction. Zhong et al. (2013) presented a method for tree extraction from an MTLS point cloud where man-made objects from natural objects are distinguished based on the analysis and examination of features in horizontal space. In their method, other points such as vegetation, trees, and vertical features are identified in a 3D classification process. Other research has also used classification for tree identification (Lindenbergh et al., 2015). Additionally, Cabo et al. (2014) developed a method for accurately calculating the crown volume of individual trees from MTLS data using a concave hull by slices method.

Integration of panoramic images with an MTLS point cloud for tree extraction was proposed by Zhang et al. (2015). They added color information to the points from the images as the action criterion for tree identification in the segmentation process. Similarly, Xu et al. (2018) identified trees based on the integration of multispectral image information with the MTLS point cloud. In their technique, after dividing the points into ground and non-ground groups, vegetation points are separated using a normalized difference vegetation index. Kyul et al. (2019) introduced a methodology that uses a bottom-up hierarchical clustering strategy to combine clusters belonging to the tree's natural components based on the disparity and heterogeneity of the two adjacent clusters. Several other researchers have used clustering techniques for tree detection (Chen et al., 2019; Forsman et al., 2016; Husain and Vaishya, 2019).

For many point-based algorithms, tree characteristics are calculated based on the extracted tree points. For instance, Yan et al. (2019) investigated the feasibility of using MTLS to estimate tree height and trunk Diameter at Breast Height (DBH) along urban streets and in urban parks. In their study, height-above-ground and Pratt circle fit methods were applied to calculate tree height and DBH, respectively. The estimated accuracy of measuring tree heights was evaluated based on the RMSE which was 35.9 cm for street trees and 46.2 cm for the park trees. Moreover, 3.77 cm and 8.95 cm were reported as the accuracy of the DBH parameter for street trees and park trees, respectively. Li et al. (2020) proposed a two-step state-of-the-art method for street tree segmentation from MTLS point clouds based on a supervised learning algorithm. In their method, 16 local statistical features were initially extracted from the sphere domain of each point. Then, these features were fused and tree crown and trunk detectors were trained through a Discrete AdaBoost algorithm. Next, the non-connectivity between adjacent trees and adjacency within single trees were exploited to locate individual trees

Because the density of MTLS point clouds can be hundreds (or even thousands) of points per square meter, the direct input of all points for processing and analysis requires a great deal of computer memory and processing time. To improve efficiency, many researchers extract trees from lidar point clouds using grid-based methods (Rutzinger et al., 2010). Wu et al. (2013) proposed a method based on supervoxel segmentation for identifying trees from MTLS point clouds in four main steps: preprocessing, localization, segmentation, and feature extraction. After extracting the trees' trunk and foliage, the geometric characteristics of each tree such as height and diameter were calculated. Böhm et al. (2016) used IQmulus cloud-based computing to handle heavy data

processing of their point-based algorithm. Grid-based methods may reduce the processing time by merging adjacent points and converting the points into regular 2D or 3D grids. Fan et al. (2016) presented a grid-based approach by calculating the density and height factors in each grid cell and using them in a segmentation process to find the nature of that cell. Zhong et al. (2017) suggested a bottom-up method in six steps: (1) preprocessing, (2) Octree construction, (3) spatial clustering, (4) tree trunk detection, (5) initial segmentation, (6) fine segmentation.

Table 1 summarizes the state-of-the-art studies on tree detection from MTLS point clouds. As can be seen from this table, the majority of the papers have focused on measuring DBH and ignore other tree parameters such as foliage diameter or distance to road.

1.2. Contributions

Although point-based methods prevent information loss by direct use of the original points, they are time-consuming and complex. Also, grid-based methods, while more efficient, suffer from information loss when converting points to a grid format. In this paper, a new method is proposed for creating a tree inventory from an MTLS point cloud based on the combination of point-based and grid-based methods. The proposed method uses a set of predefined rules and tree geometric shapes to extract the trunk followed by the tree foliage. Assuming the horizontal section of a trunk is a circle, the Hough Transform (HT) algorithm is used to identify the trunk position. By approximating the area of tree foliage based on a Voronoi Tessellation (VT), the exact boundary of each tree foliage is extracted by using active contour models on the density image at different height levels. This extraction process is one of the main contributions of this paper. Another innovation of this paper is the automatic calculation of various geometric characteristics once the extraction process is complete. In this regard, the distance to road parameter which is important for road safety analysis is also measured. This parameter has not been investigated in the previous work.

2. Method

The workflow of the proposed method is shown in Fig. 2. According to this workflow, an MTLS point cloud is imported as input data and the tree parameters are obtained in four consecutive steps: (1) preprocessing, (2) trunk extraction, (3) foliage extraction, and (4) characteristics measuring. In the preprocessing step, low height points are first removed from the input data. Then the tree trunks are identified by identifying the area and height ranges. Next, the trees' foliage is identified using the active contour algorithm. Finally, tree parameters including "Planimetric Coordinates", "Trunk Height", "Trunk Diameter at Breast Height", "Foliage Height", "Maximum Foliage Diameter", "Total Tree Height", and "Distance from Road Edge" are calculated by analyzing the extracted points of individual trees. More details of these steps are described in the following sub-sections.

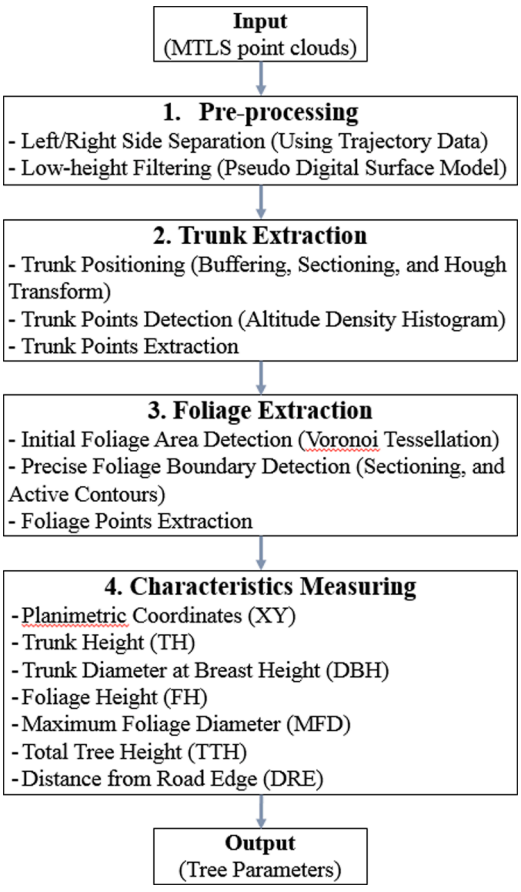

Fig. 2. The workflow of the proposed method for tree inventory from MTLS point clouds.

2.1. Preprocessing

The purpose of preprocessing is to remove redundant points to reduce the data volume and prepare the points for the next steps of the

Table 1
Summary of the previous studies on tree detection from MTLS point clouds.

Authors	Area type	Tree Parameter	No. of. Trees	Tree Extraction Accuracy	DBH Accuracy
Rutzinger et al. (2010)	Urban Roadside	Tree height, crown diameter, crown height DBH, base height	40	93%	N/A
Holopainen et al. (2013)	Urban, Forest	Tree location, DBH	118	79.22%	N/A
Zhong et al. (2013)	Urban Roadside	DBH, Crown diameter, Tree height	N/A	100%	N/A
Wu et al. (2013)	Urban Roadside	Tree height, crown diameter, DBH, crown base height	72	98.52%	1 cm
Zhang et al. (2015)	Urban Roadside	Tree location, Crown diameter, Tree height	2652	93.5%	N/A
Sirmacek and Lindenbergh (2015)	Urban	N/A	N/A	98.0.86%	N/A
Bauwens et al. (2016)	Forest	DBH, location	331	93%	less than4cm
Fan et al. (2016)	Urban	Tree height, Number of pole point	268	86%	N/A
Böhm et al. (2016)	Urban	N/A	N/A	N/A	N/A
Zhong et al. (2017)	Urban	N/A	488	94%	N/A
Xu et al. (2018)	Urban	N/A	172	98.9%	N/A
Husain and Vaishya (2019)	Urban	Crown diameter, DBH, Tree height	93	96.80%	N/A
Kyul et al. (2019)	Urban, Park	DBH, Tree height	28	N/A	3.77 cm (street) 8.95 cm (park)
Chen et al. (2019)	Urban Roadside	DBH, Tree location, Crown diameter, Tree height	151	93%	5% error
Yan et al. (2019)	Urban	Tree height	30	100%	N/A
Li et al. (2020)	Urban Roadside	Tree height, crown diameter	7	99%	N/A

algorithm. Because of the vast amount of points typical of an MTLS data set, a majority of studies initially divide the point cloud into a number of small sections (tiles) along the road (Rastiveis et al., 2020; Yu et al., 2019; Yue et al., 2015). Similarly, our algorithm uses tile sections rather than processing the whole data set at once. Moreover, the points of each section are divided into a left-side and a right-side, and the calculations for each side are performed, separately. For each side, low-height points are then filtered. For this purpose, a Pseudo-normalized Digital Surface Model (PnDSM) is created using a Coarse Digital Terrain Model (CDTM). The CDTM is generated by gridding the area and considering the minimum height in each cell as the ground level. Subtracting the CDTM from the original points (DSM), PnDSM can be obtained by

$$PnDSM = DSM - CDTM \quad (1)$$

Eventually, by considering a small height threshold and removing points below the threshold, ground points and a number of low-height points are filtered.

This low-height filtering step increases the computation speed due to two main reasons: first, by eliminating the low height points many layers near the ground surface are excluded. These points, which usually are ground points and shrubs, can be problematic in the Trunk Extraction step. Second, by using the PnDSM instead of the original points, the number of layers is limited based on the tallest object in the area instead of the difference between the minimum and maximum heights of the area that may be an inordinate amount due to changes in topography.

In this article, a PnDSM was generated using a grid size of 2 m, and low-height points were removed by applying a lower threshold of 0.5 m on the generated PnDSM. These thresholds are obtained based on trial and error on different data. They depend on the topography, point density, and trees shape and size. Selecting larger dimensions for the grid may remove the lower part of the tree trunk. A smaller grid size will increase the computational volume and may not remove all of the terrain. Regarding the height threshold, if the trunks are high in one area, a higher threshold can be considered to remove more low points. Fig. 3 shows the results of the preprocessing phase for the point cloud dataset of the intercity freeway section.

2.2. Trunk extraction

The purpose of this step is to identify a tree's trunk. The trunk can be defined as "the main stem of a tree, including wood and bark, starting at the root collar at or near the ground line and going up to a more-or-less indeterminate point, often as low the first large branch, or as high as can be traced before the stem branches into divisions that can no longer be identified as belonging to a central main stem" (Leverett and Bertolette, 2015). In this context, the points between the lowest tree point and the starting point of the foliage are defined as trunk points (Yue et al., 2015). Trunks are extracted in two steps as shown in Fig. 4: (i) trunk positioning which the location of the trunks are detected, and (ii) elevation range determination that the lowest point of the tree and the starting point of the foliage are estimated. Each of these steps is described in the following subsections.

2.2.1. Trunk positioning

The concept of this step is based on an assumption that the horizontal section of a tree trunk is approximately circular. By extracting horizontal sections from the generated PnDSM and performing a circle detection algorithm all circles can be extracted. For this purpose, a height threshold interval is used that will contain at least a portion of the trunk. Then, the points are stratified by the altitude at a specified interval so that the points in a specified height range fall into one group. In this article, a 3 m height threshold was considered to extract a portion of the trunk. All points above this threshold were eliminated. The remaining points were layered at 30 cm intervals giving 9 layers. These values were chosen empirically based on trial and error during test experiments, however, future interval heights could vary depending on the characteristics of the point cloud and tree species. For each height group, a binary image is created using its points. In this regard, if one or more points are available inside a pixel, its pixel value is assigned 1, and otherwise 0 (see Fig. 5). Proper choice of image pixel size should be meticulously assigned considering point cloud density and the fact that the binary images will be used to detect trunk circular sections of trees.

Once the binary images are generated, the circle detection step can be executed. Different methods such as Fast Circle Detection (Ayala-Ramirez et al., 2006), Genetic Algorithm (GA) (Atherton and Kerbyson, 1999), and Hough Transform (HT) (Ronfard, 1994) have been applied to detect circles in a binary image. In this study, we use the HT algorithm where points along the circle are defined based on polar coordinates. Given the coordinates of the circle center (a, b), radius of the circle (r), and the angle from the reference meridian (θ), the coordinates of each

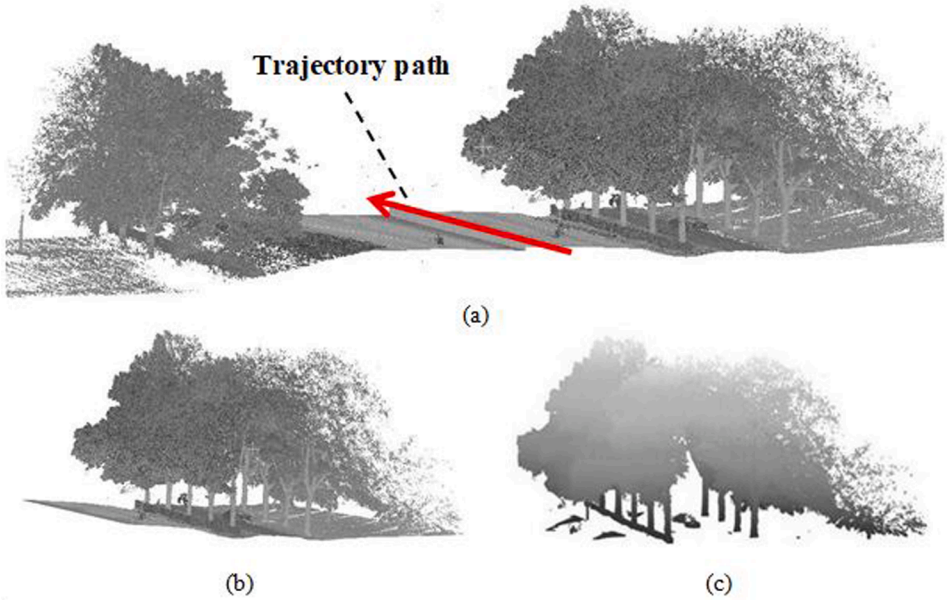

Fig 3. Sample point cloud data of the intercity road section: (a) overview of the data and the MLS trajectory, (b) extracted right roadside of the data, and (c) generated PnDSM of the right roadside of the road after the low-height point removal.

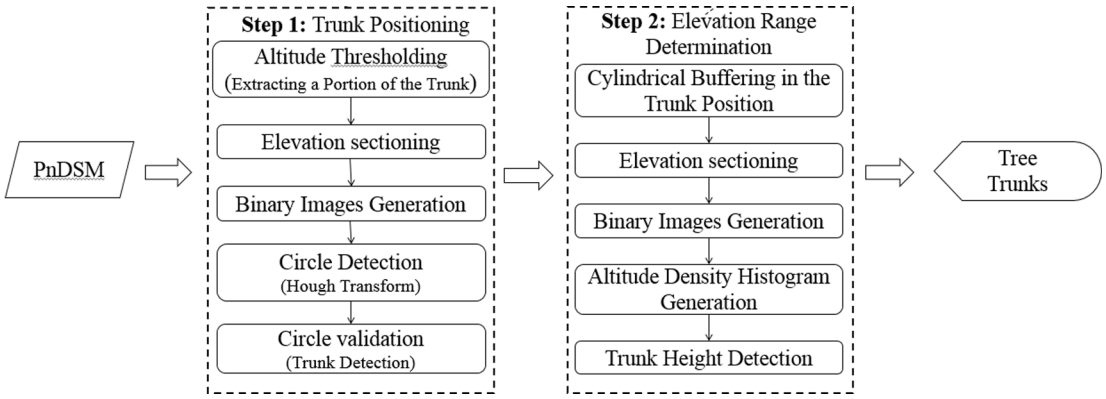

Fig. 4. Flowchart of the trunk extraction step.

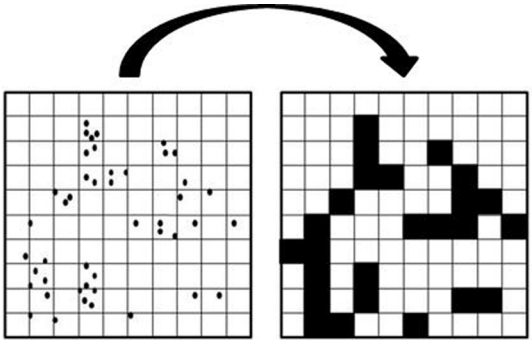

Fig. 5. Generating binary image.

point on that circle (x, y) can be calculated by

$$x = a + r\cos\theta \tag{2}$$

$$y = b + r\sin\theta \tag{3}$$

where θ ranges from 0to 360. In practice, to extract circles from a binary image using an HT circle detection algorithm, all edge pixels are first transformed to a parametric space (a, b, r) by

$$a = x_1 - r\cos\theta \tag{4}$$

$$b = y_1 - r\sin\theta \tag{5}$$

In summary, the HT algorithm detects circles on an image in four steps:

1) Extracting edges in the image using edge detection techniques.
2) Transforming edge pixels to the parametric space using Eqs. (4) and (5).
3) Detecting points with high value (peak points) in the parametric space (a, b, r).
4) Extracting the corresponding circle of each peak point on the image.

Once all circles are extracted from the binary images, a determination is made to see if the circles represent a trunk. The basic assumption is that if a circle belongs to a trunk section, there should be another circle in the adjacent layers (either above or below) in the neighborhood of that circle. Accordingly, if a series of circles are extracted from a number of layers in the same neighborhood the circles are then considered to belong to a trunk. It should be noted that it is not possible to reliably detect tree trunks by using only one section. This is because finding the right height to extract this section is difficult and in some sections, the circle may not be extracted with high reliability. Besides, every extracted circle does not necessarily represent a tree trunk. Thus, to ensure that tree trunks are detected multiple sections must be used followed by the validation step to find true tree trunks. The condition for considering the range and number of layers to be considered as a trunk can be defined as:

$$\frac{S_i}{T} \geq |C_t| \tag{6}$$

where T is the total number of layers (images), S_i is the number of images that have a circle in the radial range of the i-th circle, and C_t is a predefined threshold. For example, for a $C_t = 80\%$, if there is a circle in more than 80% of the layer images in the range of a specific coordinate, it can be assumed that all of the circles in that range are the sections of a specific trunk. After extracting all ranges, the centroid of the detected circle with the lowest height is considered as the trunk position.

Fig. 6 shows the trunk positioning process from the right side of the intercity freeway data. A binary image with a pixel size of 5 cm was generated for each of the 9 sections and circles in each image were identified by the HT algorithm. In this figure, the binary images and circles obtained from this sample are shown; the detected tree trunks are bolded in green after validating detected circles.

2.2.2. *Elevation range determination*

The goal of this stage is to extract all the points of the trunks in the detected places in the trunk positioning stage. This is performed in five steps: (1) buffering, (2) elevation sectioning, (3) raster image generation in each section, (4) altitude density histogram generation, and (5) density thresholding (see Fig. 7).

As illustrated in Fig. 7, a cylindrical buffer space is considered around each trunk location on the generated PnDSM with a radius larger than the maximum trunk radius in the area. In this case, by considering a thinner buffer the density changing at the start point of the foliage would not be clear. On the other hand, the wider buffer may import an adjacent tree into computation and may cause an error. Then, all the points in the PnDSM inside that buffer space are extracted, and similar to the previous stage, altitude sectioning of the extracted points is performed for the extracted points of each tree. The distance between each section is determined depending on how accurately the tree trunk height is to be estimated. For example, to calculate the tree trunk height by precision of 10 cm, sectioning should be done at intervals of less than 10 cm. Next, a binary image is generated for each section. The pixel size of these binary images may vary for different projects based on point cloud density, tree species, and foliage shape. The typical pixel size for most projects is between 5 cm and 15 cm. Due to the tree trunk shape and the distribution of its points, it is expected that the number of pixels that have a value of 1 in the binary images for each section within the trunk's altitude range is similar. The binary images are used to estimate the density of each section to generate an altitude density histogram in the fourth step. It is expected that the distribution of points in the trunk altitude range is less than the foliage range. In fact, where the altitudinal

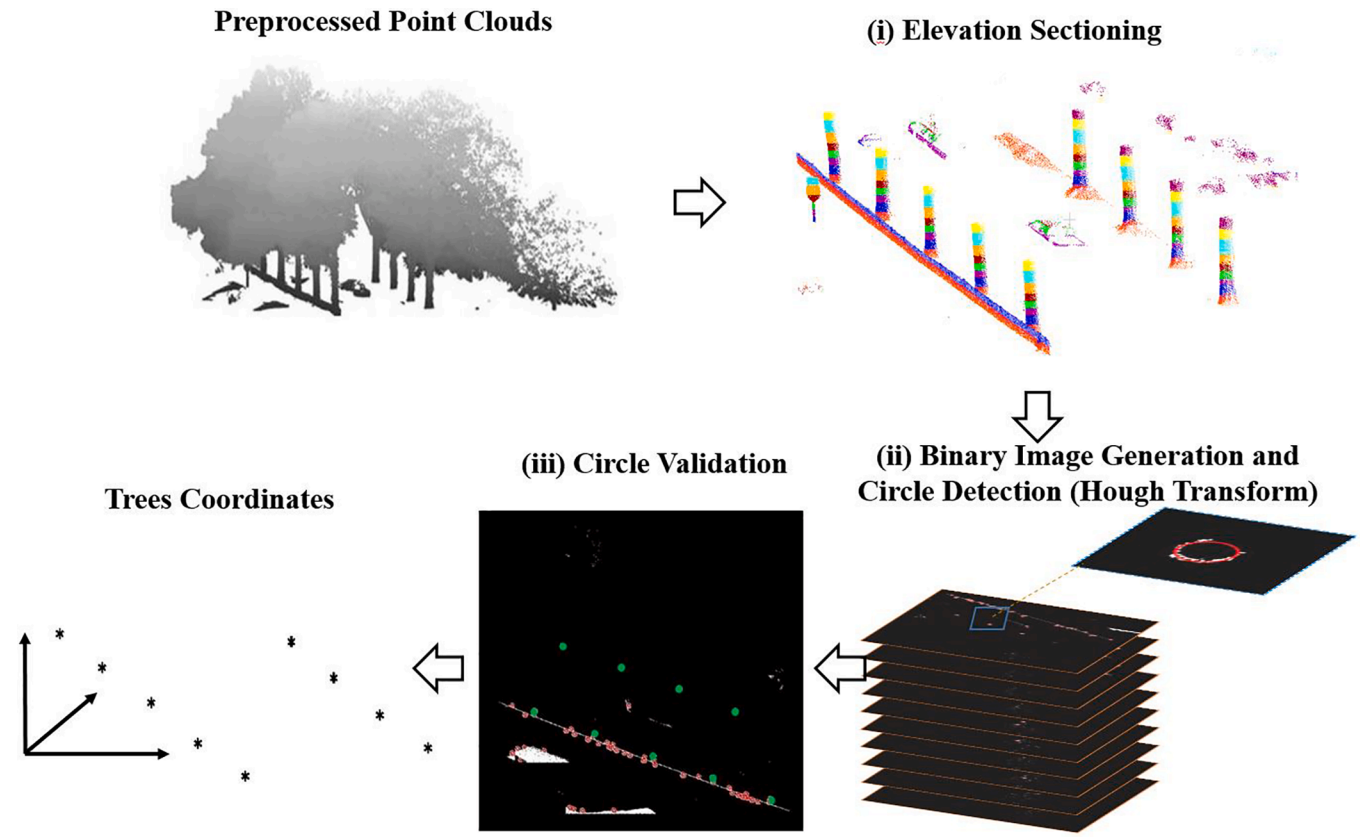

Fig. 6. Trunk position detection process.

range of the tree trunk ends and the altitude range of the foliage begins, the dispersion of the points is expected to increase sharply. Therefore, in the fifth step, the difference in geometric distribution of trunk and foliage is used to find the first section of the foliage whose dispersion of points will be much greater than the dispersion of points for the highest trunk section. Once the first foliage section is found, all of the points below this level and in neighborhood of the tree trunk position are considered as trunks points.

In the trunk point extraction step, the buffering radius was considered based on the trunk positions obtained in the previous step. To ensure separation of trunk points for individual trees, this radius cannot exceed half of the minimum distance between the extracted trunks. It is worth mentioning that, by considering a smaller buffer the change in density at the start point of the foliage may not be apparent in the density histogram. On the other hand, the wider buffer may import an adjacent tree into the computation and may cause an error. In our dataset, a 3-meter radius around the trunk center (circle center) was considered to determine the trunk height. The point dispersion histogram of each tree was obtained by considering the sectioning interval of 20 cm, and the pixel size of 10 cm for binary image generation. Fig. 8 illustrates the obtained histogram for three sample trees. The separation between trunk points and foliage points is identified based on the change in point dispersion between sections. The extracted trunk points are shown in red color in Fig. 8. These steps are also implemented to extract tree points for other trees on both sides of the road (see Fig. 9).

2.3. Foliage extraction

The difference in tree species is mostly evident in their foliage shape. Unlike a tree trunk which is geometrically determinable and predictable, the geometric behavior of a tree's foliage is largely irregular and asymmetric which increases uncertainty in its extraction. The proposed method for foliage extraction is done in five steps: (1) determining the initial range, (2) extracting the initial foliage points, (3) altitude sectioning, (4) density image generation for each section, and (5) precise boundary detection using active contours in each level (see Fig. 10).

To extract the tree foliage, it is necessary to first estimate the altitude range of each tree's foliage. A Voronoi Tessellation (VT) of the extracted trunk center coordinates is applicable for this purpose. In a two-dimensional space including n vertices, the VT algorithm divides a plane into n separated regions where all points within a region are closer to the generator point of that region than the other generator points (Aurenhammer et al., 2013). The mathematical definition of the VT algorithm can be defined as

$$V_i = \left\{ x \in R^N \middle| |x - z_i| < |x - z_j| \quad for\ j = 1, \cdots, k, j \neq i \right\} \tag{7}$$

where $\{z_i\}_{i=1}^{k}$ represents the generator vertex, and V_i is the boundary that is defined for each point (x). Once an approximate boundary of a tree's foliage is determined using the VT algorithm a 2-meter buffer is added to account for uncertainty caused by the interference of adjacent trees' foliage.

After extracting the tree region points and removing the trunk points identified in the previous stage, all possible foliage points are extracted. Then, altitude sectioning is applied in the third step. Depending on the foliage height and Tree species, the height of each section may vary based on the tree size and shape. Given the heterogeneous and asymmetric plane distribution of the foliage geometry, it is necessary to distinguish its planar range in several height sections. The altitude sectioning was done at 2 m intervals because of the size of the trees in our dataset. We reached this number based on trial and error. Larger trees are expected to have sizable foliage in comparison to smaller trees and are of greatest interest—especially from a road user safety standpoint. A smaller interval can be used for smaller trees or if more precise tree information is needed however this would increase computation

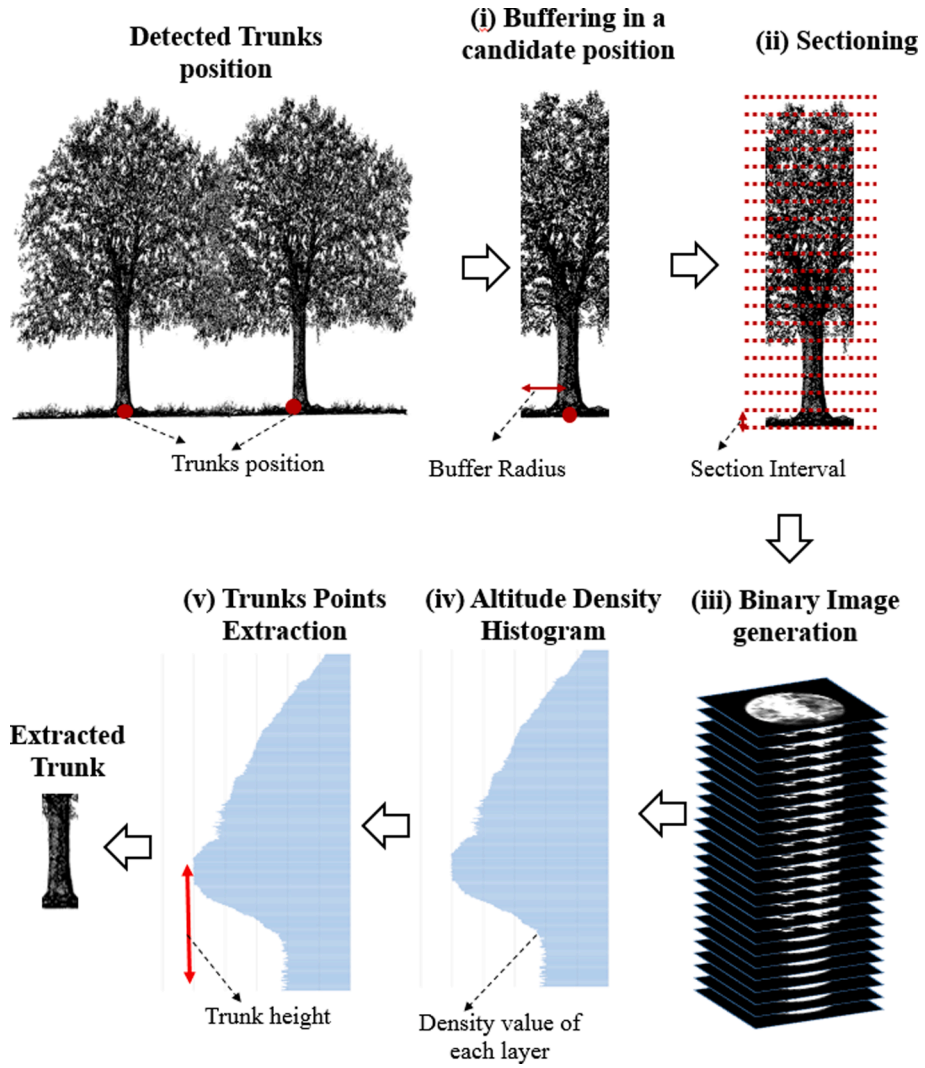

Fig. 7. Trunk's elevation range determination.

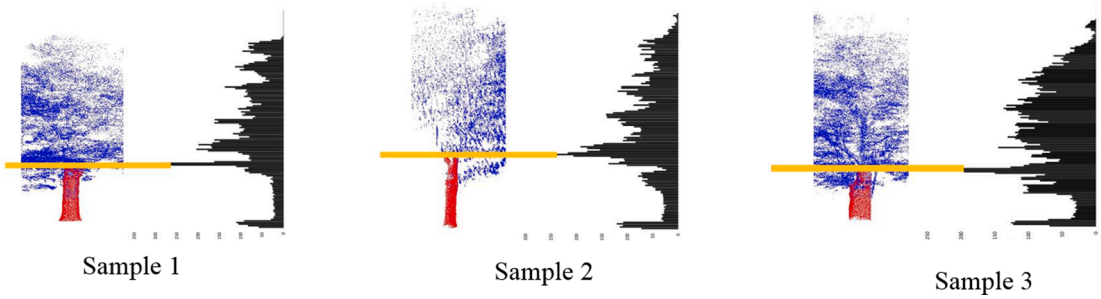

Fig. 8. Detecting the range of tree trunk height.

requirements. Then, in the fourth step, a density image is created for each section in which the gray value of each pixel in the image space is the number of points in that. The dimensions of the image grid can vary depending on point density, tree species, and foliage shape. Generally, a larger pixel size will reduce the polygon detection accuracy in the next stage. On the other hand, reducing grid dimensions will blur the density image. In this study, a 10 cm pixel size was selected for generating density images. Fig. 11 (a) shows the resultant VT polygons for the right-side of the intercity freeway data, and Fig. 11 (b) shows a generated density image of a sample section of a tree foliage in a selected VT polygon.

In the last step of the foliage extraction process, for each density image produced for a section, foliage pixels are bounded by polygons. To identify these polygons in the density image, Active Contours (AC) are used. The AC models can be divided into parametric and geometric groups; each of which is performed either edge-based or region-based to provide a method for determining a specific range (horizontal foliage range for each tree) (Bryant, 1989). The geometric region-based AC model is applied in the proposed method. This model is defined non-explicitly in the form of a surface equal to a function with a higher dimension, (a three-dimensional surface) and the evolution process is performed on the defined function. The core of this model is the

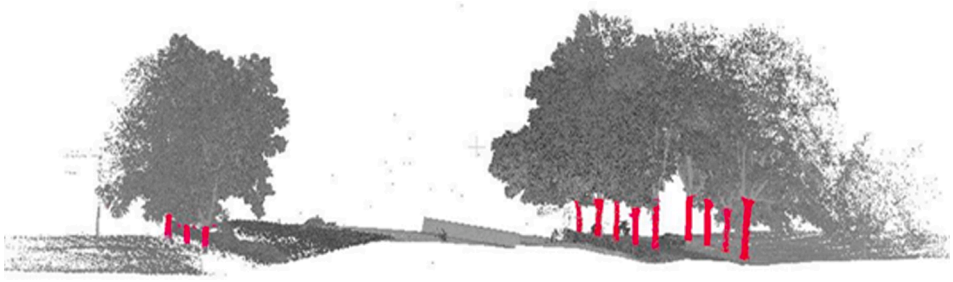

Fig. 9. Trunk points extracted in intercity data.

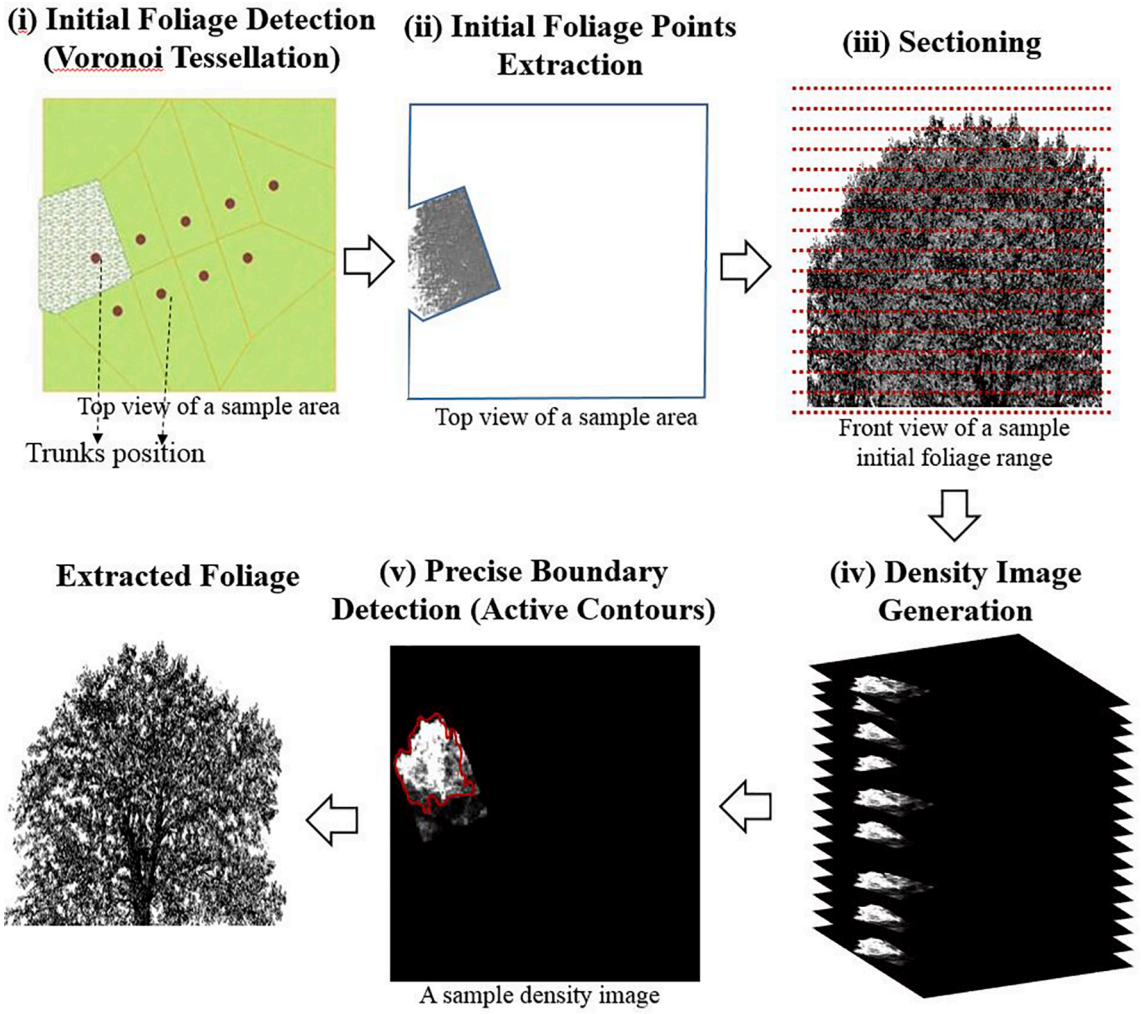

Fig. 10. Foliage extraction process.

Mumford-shah energy function provided by Palo and Alto (1988). The main idea of image segmentation in these models is to examine the degree of pixel convergence inside and outside the curve based on the weighting coefficients of the two internal and external factors defined in the energy function. In these models, the level set theory is used to display the curve and minimize the energy function, and the geometric function φ changes in a repetitive process in such a way that the model's energy function falls to its lowest value. The introduced energy function is defined as follows:

$$E(C) = \mu.Length(C) + v.Area(insicde(C)) + \lambda_1 \cdot \iint_{inside} |\mu_0(x,y) - c_1|^2 dxdy + \lambda_2 \cdot \iint_{outside} |\mu_0(x,y) - c_2|^2 dxdy \quad (8)$$

To minimize the above-mentioned energy function using level set theory, curve C is replaced with the function φ. The variation equation for the function φ must be calculated somehow that the energy function $E(\varphi)$ is minimized. To do this, the Steepest Descent method is used to minimize the energy function in a duplicate process. In this research, the mathematical background and equations of the AC model are not presented, and readers are referred to (Chan and Vese, 1977; Zhong et al., 2013) for more information. Using the AC model, a precise tree foliage boundary was identified in the density image of each section. Fig. 12 (b), (c), and (d) illustrates the performance of the AC model for 500, 600, and 630 (favorable) repetitions, respectively.

After identifying the boundary of the foliage and separating its points in each section, a final set of foliage points is extracted. This process was

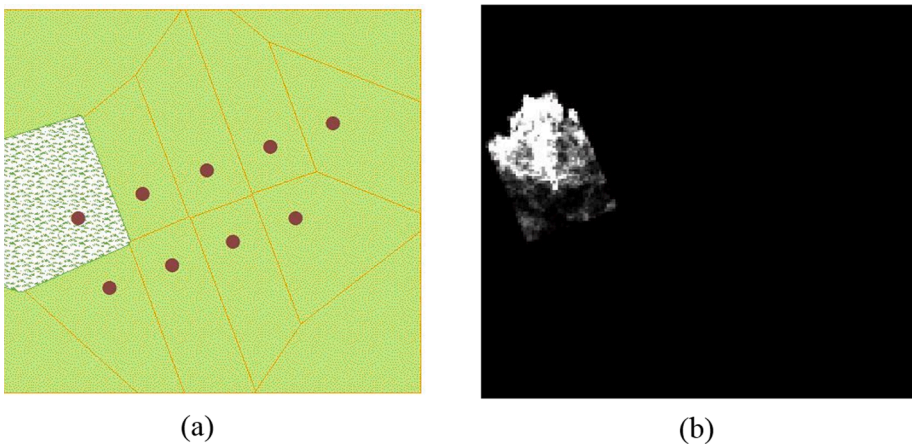

Fig. 11. Initial foliage detection step: (a) Detected initial foliage range using VT algorithm, and (b) A generated density image of the first tree.

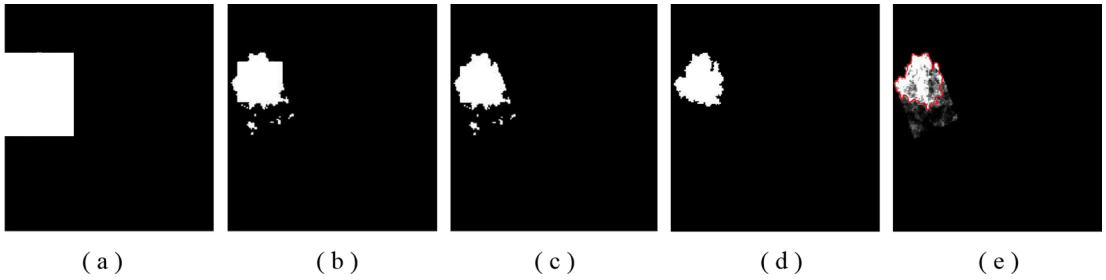

Fig. 12. Applying the AC model to identify the precise foliage boundary of the sample section: (a) initial polygons of the AC model, (b) result of AC model after 500 repetitions, (c) result of AC model after 600 repetitions, (d) result of AC model after 630 repetitions, and (e) overlaying the final resultant boundary on the density image.

performed on all trees in the intercity freeway dataset and the results are shown in Fig. 13. To this point, the trunk points and the foliage points are separated into two distinct categories to facilitate the calculation of the geometric characteristics of each tree.

2.4. Characteristics measuring

In this paper, Planimetric Coordinates (XY), Trunk Height (TH), Trunk Diameter at Breast Height (TDBH), Foliage Height (FH), Maximum Foliage Diameter (MFD), Total Tree Height (TTH), and Distance from Road Edge (DRE) are calculated for each tree. These tree characteristics are illustrated in Fig. 14 and are described in detail in the following paragraphs.

Planimetric Coordinates (XY): In this research, the planimetric coordinates of a tree's location are represented by the center of the circle with the lowest height among all detected circles during the tree trunk extraction stage.

Trunk Height (TH): After extracting the tree's trunk points, the difference between the maximum and minimum heights of the trunk points is calculated as its trunk height as follows:

$$H_{trunk} = Z_{tmax} - Z_{tmin} \tag{9}$$

where, Z_{tmax} and Z_{tmin} are the maximum and minimum height of trunk points, respectively.

Trunk Diameter (TD) and Trunk Diameter at Breast Height (DBH): Because a tree's trunk diameter can vary, an average value is calculated for each tree. This diameter is given as the mean circle diameter of all of the tree sections that make up the trunk. It is calculated as follows:

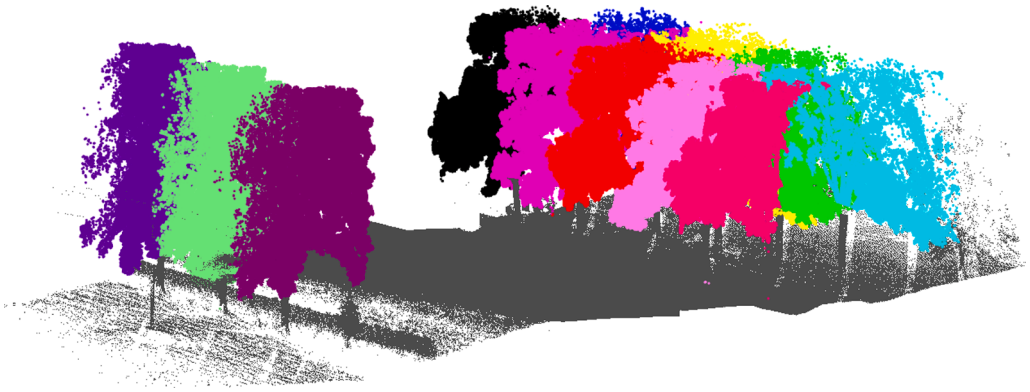

Fig. 13. Extracted foliage of all the trees in the intercity freeway dataset.

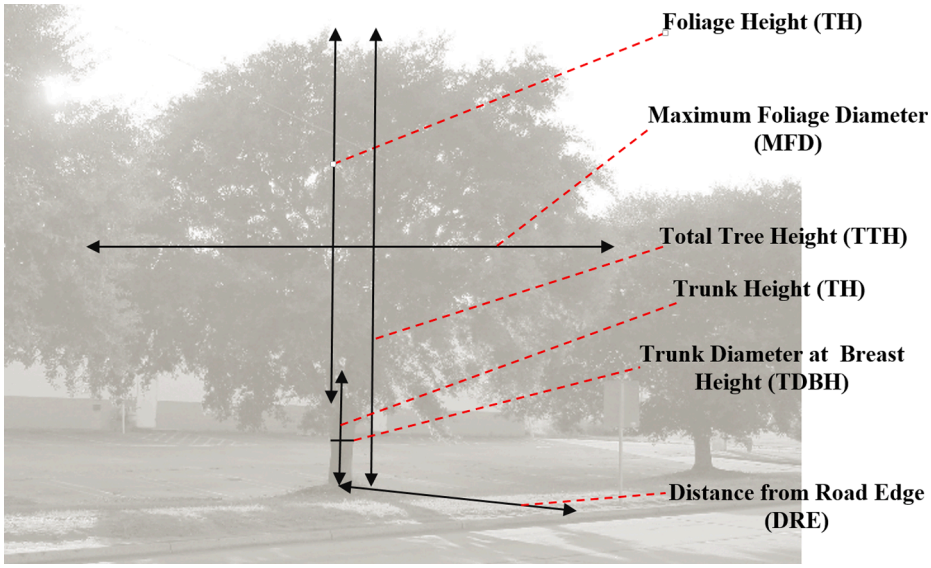

Fig. 14. Characteristics of a roadside tree.

$$D_{trunk} = \frac{\sum_{i=1}^{n} d_i}{n} \tag{10}$$

where $d_i$and n are the tree diameter in the i-th section and number of sections, respectively. A common definition for a tree's trunk diameter is the diameter at breast height (DBH). DBH is assumed at a height of 1.37 m (4.3 ft) above the ground (Brack, 2009). DBH is also acceptable for use in traffic safety.

Foliage Height (FH): The height difference between the maximum and minimum values of the heights of all of the foliage points is used to calculate the tree foliage height as follows:

$$\mathbf{FH} = Z_{f\,max} - Z_{f\,min} \tag{11}$$

where $Z_{f\,max}$ and $Z_{f\,min}$ are the maximum height of foliage points and the minimum height of foliage points, respectively.

Maximum Foliage Diameter (MFD): a peripheral circle can be defined from the foliage polygon boundary of each section. In this paper, MFD is the maximum diameter of all sections' peripheral circles.

D_{pc} : Peripheral circle diameter

Total Tree Height (TTH): TTH of a tree is equal to the difference between the minimum height of the trunk points and the maximum height of the foliage points.

$$\mathbf{TTH} = Max(Z_{FP}) - \min(Z_{TP}) \tag{12}$$

Distance from Road Edge (DRE): One of the important parameters for road safety analysis is tree distance from the road edge. Trees too close to the road edge can be hazardous if their DBH is greater than 10 cm. Trees can also limit a driver's sight distance. In this research, a digital map is used to represent the road edge. DRE is calculated as the distance between the trunk position and the road edge.

3. Experiments and results

To test the performance of the proposed method, this algorithm was implemented on two MTLS sample datasets. This section discusses the sample data sets, the results, and the accuracy assessment.

3.1. Test data

The two MTLS point cloud datasets were collected on roadways in the upstate of South Carolina, USA. The first dataset was collected along an intercity freeway in Spartanburg, South Carolina. In addition to the road-side trees, there are several other roadside and objects such as advertising billboards, power poles, and associated cabling, fencing, and nearby buildings. From this dataset, a sample section includes 3.3 million points covering a 195 m long road section which approximately is 440 m wide was selected. The MTLS system used to collect the roadway point clouds in this study area was the Teledyne Optech Lynx Mobile Mapper M1, a mobile LiDAR mapping system that first appeared in September 2010. This system consists of two LiDAR sensors operating at 500 kHz each and includes 4 digital cameras. The system's published range precision and absolute accuracies are 8 mm and ± 5 cm, respectively. The scanner field of view is 360° without obscurations.

The second dataset was collected along an urban road in Anderson, SC. In addition to trees and other side-effects of the road, there are also urban buildings and facilities. The selected sample section from this dataset contains almost 2 million points covering a 226 m long urban road section and has a maximum width of approximately 174 m wide. The MTLS system used in this study area was the Teledyne Optech Lynx SG1 Mobile Mapper that first appeared in June 2013. The SG1 system also consists of two LiDAR sensors and 4 cameras. It has a 600-kHz dual LiDAR scanning system that can collect up to 1.2 million measurements per second, spread over an unobstructed 360° field of view.

The MTLS datasets for the intercity freeway and the urban roadway include point clouds collected using one pass in each direction. The roadway sections where the algorithm was implemented are shown in Fig. 15(a) for the urban roadway and 15(b) for the intercity freeway.

3.2. Results

Fig. 16 shows the extracted trees in the intercity freeway dataset by combining their trunk and foliage points. They were analyzed to compute their geometric characteristics which are summarized in Table 2.

Fig. 17 shows the results obtained from the urban road dataset using the proposed algorithm that contained three trees. A raw point cloud of the dataset is shown in Fig. 17(a); off-ground points after low-height point removal are shown in Fig. 17 (b); detected tree trunks are shown in Fig. 17(c); detected tree canopies are shown in Fig. 17 (d); and the extracted individual trees are shown in Fig. 17 (e). Table 3 lists the geometric characteristics of the three individual trees extracted from the urban road dataset.

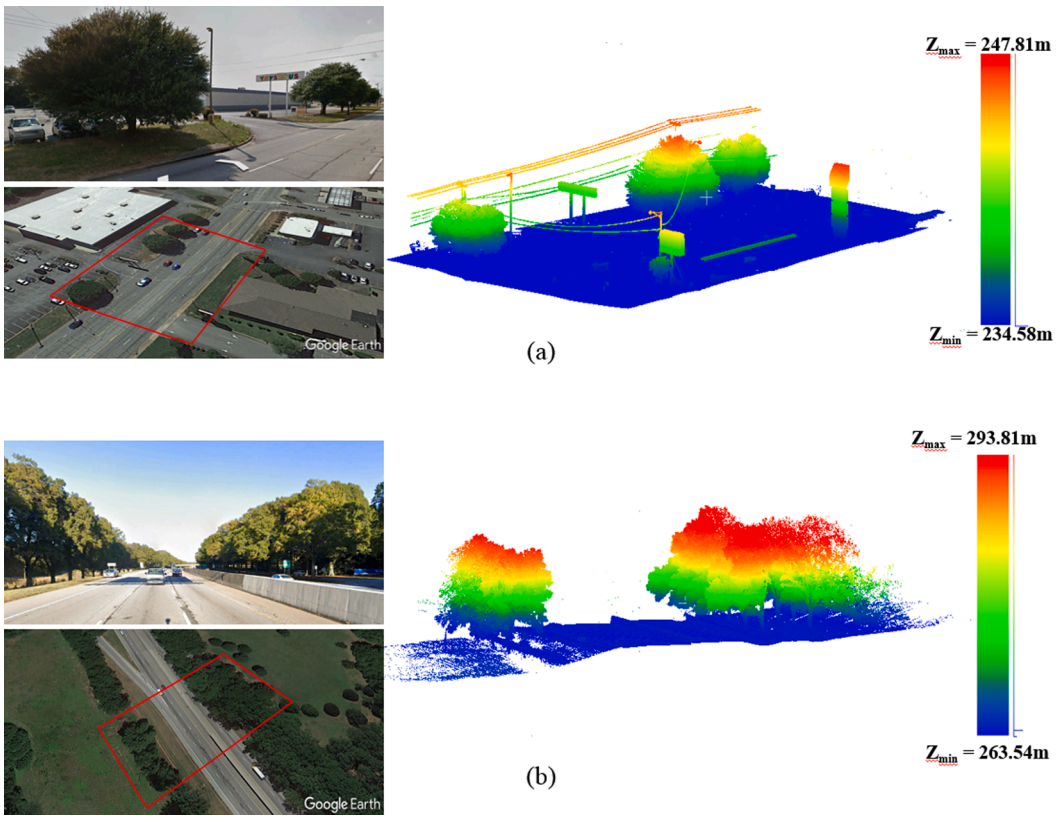

Fig. 15. Sample point cloud data and the corresponding Google images (street view and front view) covering; (a) an urban road section; (b) an intercity freeway section.

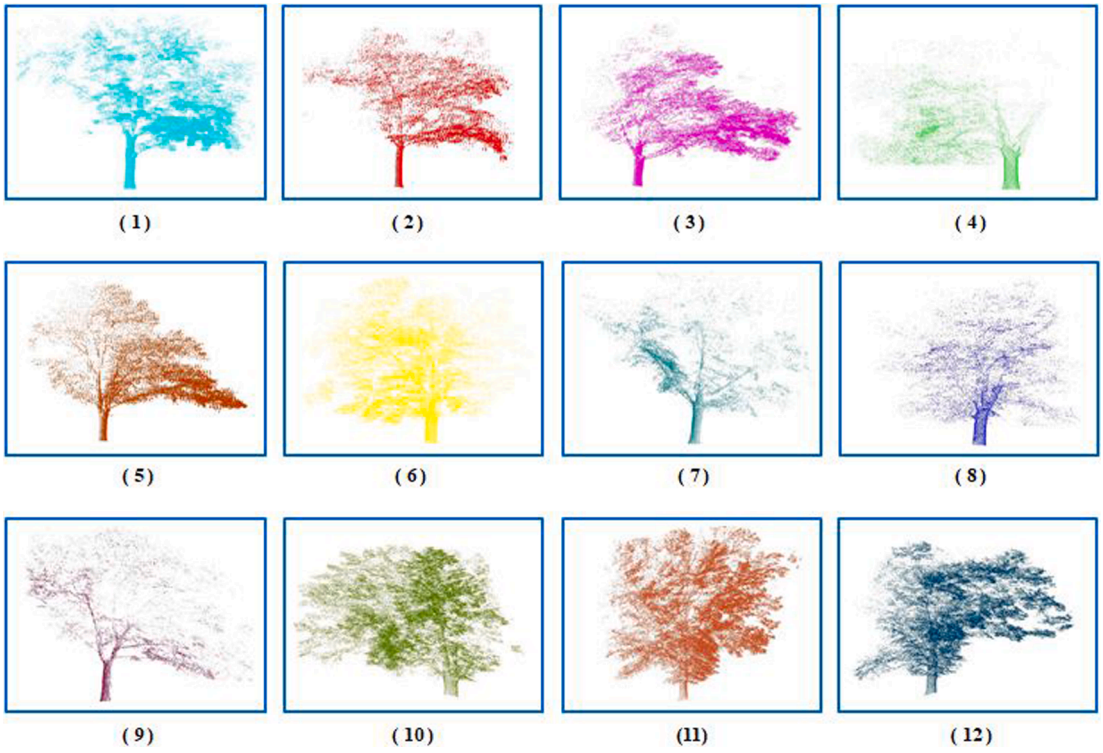

Fig. 16. Extracted trees in the intercity area.

Table 2
Extracted tree characteristics in the intercity freeway dataset. Planimetric Coordinates (Xc, Yc), Trunk Height (TH), Trunk Diameter at Breast Height (TDBH), Foliage Height (FH), Maximum Foliage Diameter (MFD), Total Tree Height (TTH), and Distance from Road Edge (DRE).

ID	***Xc*** (m)	***Yc***(m)	**TDBH**(m)	**TH**(m)	**FH**(m)	**TTH**(m)	**MFD**(m)	**DRE**(m)
1	521828.40	350598.12	0.93	8.96	24.67	28.40	21.37	22.52
2	521851.82	350606.95	0.89	7.39	21.71	26.62	21.89	22.39
3	521874.93	350615.48	0.90	5.92	21.44	24.65	20.25	22.34
4	521867.99	350598.18	0.92	6.52	19.10	24.45	14.40	22.23
5	521840.11	350602.61	0.89	7.98	21.80	25.90	22.78	22.31
6	521863.43	350611.23	0.70	6.72	24.74	28.14	20.55	36.19
7	521845.49	350598.66	0.90	6.83	20.70	24.95	16.21	36.32
8	521856.66	350593.91	0.89	6.54	21.15	24.10	21.75	36.78
9	521,834	350585.39	0.96	6.50	24.89	28.22	23.30	36.51
10	521833.54	350666.84	1.01	2.75	20.11	20.32	19.89	11.87
11	521844.97	350670.84	1.17	2.76	19.93	19.65	20.01	11.84
12	521857.17	350674.95	1.21	3.57	20.75	21.54	19.41	18.86

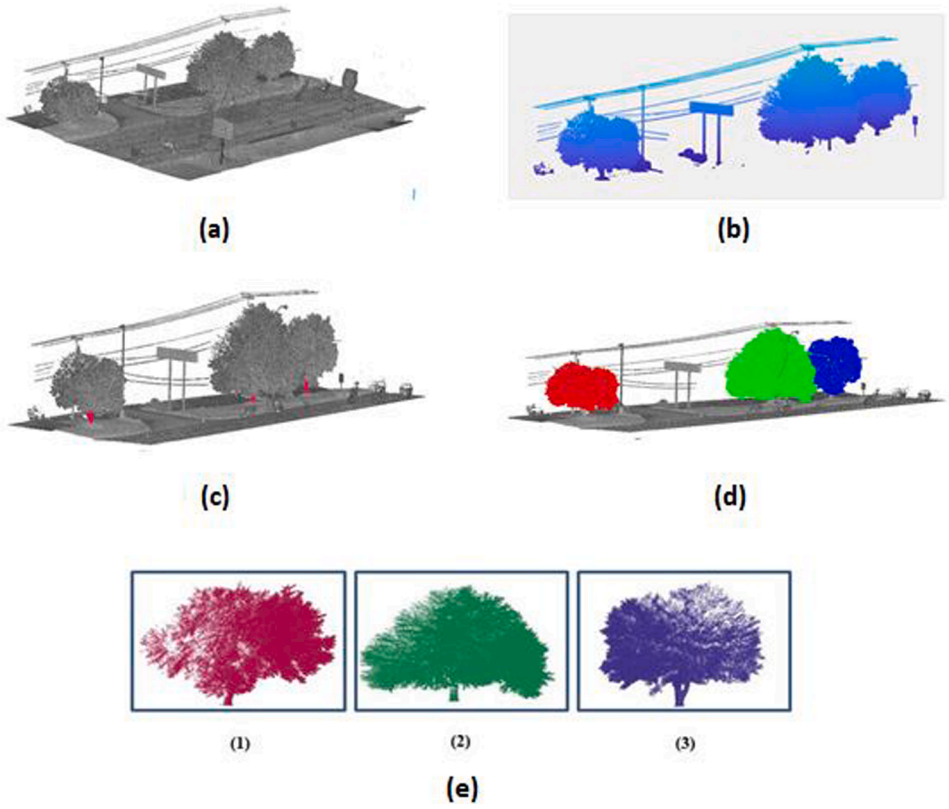

Fig. 17. Results obtained from the urban road dataset: (a) raw point cloud, (b) generated PnDSM after ground removal, (c) extracted trunks of the area, (d) extracted foliage, and (e) extracted individual trees.

Table 3
Characteristics of the trees in the urban road dataset. Planimetric Coordinates (Xc, Yc), Trunk Height (TH), Trunk Diameter at Breast Height (TDBH), Foliage Height (FH), Maximum Foliage Diameter (MFD), Total Tree Height (TTH), and Distance from Road Edge (DRE).

	Xc (m)	***Yc***(m)	T***D***BH(m)	TH(m)	FH(m)	TTH(m)	MFD(m)	DRE(m)
1	456298.50	302066.30	0.48	1.45	7.82	8.46	10.27	6.87
2	456305.63	302.53.63	0.57	2.11	11.48	12.11	15.10	6.42
3	456327.99	302013.78	0.74	1.33	6.42	7.01	11.70	6.35

3.3. Accuracy assessment

To evaluate the accuracy of the proposed method, the extracted characteristics obtained from the proposed method were compared with the actual values of the characteristics of individual trees. These actual values were painstakingly determined by manually analyzing each of the trees in each dataset. Table 4 illustrates the difference between the values obtained from the manual extraction process and the extracted values by the proposed method. Because DRE is based on the planimetric position of a tree relative to the road edge, the RMSE of Xc and Yc was calculated as dDRE.

The differences values given in Table 4 indicate that the extraction of the tree parameters can be done with a promising accuracy. It an be seen that the foliage diameter had the greatest error which is likely due to

Table 4
Difference between the parameters extracted using the proposed method and the manual measurement (units are in centimeters). Difference (d), Planimetric Coordinates (Xc, Yc), Trunk Height (TH), Trunk Diameter at Breast Height (TDBH), Foliage Height (FH), Maximum Foliage Diameter (MFD), Total Tree Height (TTH), and Distance from Road Edge (DRE).

	ID	d***Xc***	d***Yc***	d***DBH***	dTH	dFH	dTTH	dMFD	dDRE
Intercity Data	1	23	12	1	10	5	5	0.18	26
	2	2	21	2	14	5	8	22	21
	3	6	1	2	9	10	1	17	6
	4	2	4	1	2	2	6	20	4
	5	2	9	1	17	5	14	20	9
	6	2	2	1	10	8	2	11	3
	7	1	4	1	3	17	15	11	4
	8	2	8	1	13	6	7	13	14
	9	9	9	1	1	10	7	15	13
	10	12	5	2	10	16	24	19	13
	11	3	10	1	6	7	5	2	10
	12	9	4	1	7	6	13	19	10
Urban Area	1	30	19	2	1	2	3	12	33
	2	2	5	1	7	1	7	5	5
	3	21	22	10	4	3	8	12	27
min		1	1	1	1	1	1	2	4
max		30	22	10	17	17	24	22	53
Avg.		11	6	2	8	7	8	14	16

foliage interference between adjacent trees. Of particular importance from a roadway safety standpoint is that the algorithm extracted TDBH parameters had a median difference from actual of only 1 cm. However, the dTDBH of the third tree in the urban dataset was considerably higher than the other trees which may be due to the bifurcation in the trunk of the tree giving rise to two roughly equal diameter branches (see the tree with ID 3 in Fig. 17(e).

The accuracy of the proposed algorithm was also evaluated by comparing the extracted TDBH and DRE parameters with values obtained from field surveying. Field measurement of the TDBH was done by using a surveying fiberglass tape to measure the circumference of the tree at a height of 1.37 m and converting the circumference measurement to diameter by dividing the circumference by pi. The DRE was considered as the closest point of the tree trunk at ground level to the edge of the asphalt. Table 5 summarizes the evaluation results. As can be seen from this table, the average error between measured TDBH and DRE through field measurement and the proposed method are 9.4 cm and 52 cm (0.52 m), respectively.

4. Discussion

4.1. Analysis of the results

The accuracy assessment indicates that the proposed method is capable of identifying trees and calculating tree characteristics at a relatively high level of accuracy. It is noteworthy that the accuracy of the proposed method was not dependent on the size of the tree or the point cloud density of the tree foliage. For example, as shown in Fig. 16, the point densities for trees with IDs 4, 7, 8, and 9 are lower than the other trees. This is because the distance between the MTLS vehicle and these trees is greater than the other trees. With regard to tree size, the maximum and minimum TDBH in the intercity section was 1.20 m and 0.49 m, respectively. Both were measured within 2 cm of actual. In terms of elevation, the tallest tree is the first tree of the intercity section with a height of 32.33 m and the shortest tree is the third tree of the urban section with a height of 7.75 m. Both trees' heights were modeled with a high degree of accuracy.

Another point that is apparent in the output of the algorithm is the very good foliage recognition of each tree, despite interference between trees nearby. Fig. 18 shows how the method can distinguish between the foliage of two adjacent trees.

Another feature of the proposed method in this study is the proper

Table 5
The difference in the extraction of characteristics using the proposed method and manual method.

Measures		TDBH(cm)			DRE(m)		
Dataset	Tree ID	Proposed Method	Field Measurement	Diff.	Proposed Method	Field Measurement	Diff.
Intercity Data	1	93	102	9	22.52	22.03	0.49
	2	89	105	16	22.39	21.8	0.59
	3	90	97	7	22.34	21.79	0.55
	4	92	97	5	22.23	21.39	0.84
	5	89	103	14	22.31	21.89	0.42
	6	70	82	12	36.19	35.71	0.48
	7	90	104	14	36.32	35.86	0.46
	8	89	104	15	36.78	35.96	0.82
	9	96	105	9	36.51	35.87	0.64
	10	101	109	8	11.87	11.29	0.58
	11	117	125	8	11.84	11.19	0.65
	12	121	125	4	18.86	18.27	0.59
Urban Data	1	58	67	9	6.87	6.49	0.38
	2	57	64	7	6.42	6.34	0.08
	3	64	68	4	6.35	6.16	0.19
		Min		4	Min		0.08
		Max		16	Max		0.84
		Avg.		9.4	Avg.		0.52

Fig. 18. Detecting accurate foliage of adjacent trees despite interference.

detection and identification of the tree, despite interference with other objects such as poles, traffic signs, power cables, etc. Fig. 19 shows how the utility pole is differentiated from a tree in the urban roadway dataset. The trunk extraction step was able to distinguish the tree trunk from the utility pole and the AC model eliminated the top part of the pole from the foliage.

In terms of computing time, the algorithm was implemented in MATLAB 2018-a environment on a computer with an Intel® Core™ i7-4500U CPU and 4.0 GB of RAM processing, and extraction took about 4–6 min while the tree parameters were calculated in less than a minute. Specifically, the approximate time of extraction of trunk XY position, extraction of trunk points, and extraction of tree foliage points were 1.5 min, 1.5 min, and 2 min, respectively.

4.2. Sensitivity analysis of effective parameters

A sensitivity analysis was conducted to optimize the values of important input parameters. Optimizing each of these parameters will improve algorithm results and enhance the precision of measured parameters. For example, selecting larger dimensions for the grid may remove the lower part of the tree trunk. A smaller grid size will increase the computational volume and may not remove all of the terrain.

In the proposed method, a binary image is produced in each of the two stages. In the trunk positioning stage, because the purpose is to identify trunk sections similar to a circle, the produced binary image should have the actual curvature and shape of the sections with proper detail. For this dataset, a 5 cm pixel size for the binary image gave the best results. When testing larger pixel sizes, fewer details of circular-like sections were visible which impaired the ability of the HT algorithm to detect the trunk. On the other hand, smaller pixel sizes caused excessive detail of various effects shown in the binary image leading to erroneous detection of these effects and subsequent incorrect detection of the tree trunk sections.

In the trunk point extraction stage, to differentiate the range of trunk and foliage, the geometrical distribution of the planar surfaces of these two components is used. At this point, the binary image is used to represent this different distribution. Pixel size at this stage is less effective than at the previous stage; however, increasing pixel size will make it harder to detect the difference in distribution. At this stage, the optimum size was assumed to be 10 cm.

The choice of pixel size for the formation of a density image in the foliage extraction step is also of great importance. In this paper, the optimum pixel size of 10 cm was found by testing different values. By decreasing this pixel size, there are likely to be fewer points in the pixel resulting in a lower grayscale and lower contrast which makes it difficult for the AC algorithm to determine the range of foliage. On the other hand, with increasing pixel size, the contrast of the image is increased making the range detection process is easier; however, the edge detection is done in less detail and accuracy causing incorrect foliage range detection (see Fig. 20).

The magnitude of the sectioning interval depends on tree size and the height range of the tree's foliage. In this study, a sectioning interval of 2 m was determined to be the optimal value for sectioning the tree's foliage. Sectioning to detect the height range of the tree trunk is more sensitive than the foliage detection stage because the purpose is to accurately detect the height of the tree trunk. Thus, the section interval in this stage should be relatively small to accurately detect the height of the tree trunk. Further, the choice of the sectioning height is directly dependent on tree size. For this dataset, an interval height of 20 cm was determined to be sufficient for detecting an accurate tree trunk height. Reducing this interval unnecessarily increases computation. Conversely, using higher interval values will decrease the accuracy of detecting the height of a tree trunk.

The algorithm uses height threshold limits in the low-height removal stage and the trunk positioning stage. In the low-height removal stage, after creating the PnDSM, the lower elevation points should be eliminated as ground points by applying a height threshold. A value of 50 cm was used in this study with good results. Values greater than 50 cm removes a significant portion of the tree trunk or may even remove small trees. Conversely, if the threshold value is set too low, several extra layers may be investigated in the next step. In the trunk positioning process, the purpose of applying a high threshold is to remove additional points and to also limit the study space. The value of this threshold largely depends on the tree trunk height. In this paper, this value was assumed to be 3 m considering the size of trees in the study area. By increasing this threshold, foliage points may not completely be eliminated, which may hamper the performance of the HT algorithm to identify circular sections of tree trunks in the upper layers.

Fig. 19. Identify the tree despite the interference with other objects.

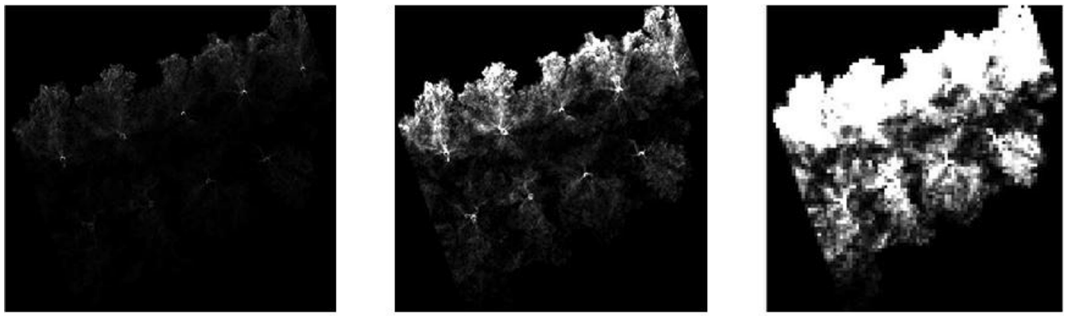

Fig. 20. Influence of pixel size on density image type and detection of foliage range of trees. (a) Pixel size = 3 cm – (b) Pixel size = 10 cm (optimum) – (c) Pixel size = 25 cm.

4.3. Comparative study

The proposed method in this paper has advantages over other state-of-the-art tree extraction algorithms. For example, in most previous studies, e.g. (Sirmacek and Lindenbergh, 2015; Xu et al., 2018; Zhong et al., 2017), the tree components were not identified and the tree characteristics were not calculated. Further, their method had difficulty in detecting the interference of adjacent trees. In many of the articles reviewed in the Introduction section (Table 1), not all trees have been completely extracted in the study environment. For example, Yao and Fan (2013) were only able to identify about 90% of the trees in their dataset. It is noteworthy that their study area was considerably large in comparison with previous studies, but the method presented in this research resulted in all of the trees being successfully extracted in both urban and intercity areas. The methods described in (Yan et al., 2019) and (Li et al., 2020) also successfully extracted all trees but that was for a very small dataset.

The reported accuracy of tree locations in Holopainen et al. (2013) was reported to be about 0.5 m for both manual and automated methods which are significantly lower than the location accuracy of 11 cm average in the x-direction and 6 cm average in the y-direction achieved on this research. Concerning DBH, only a few researchers have reported their accuracy in measuring DBH. However, the observed 1 cm DBH accuracy is much better than that observed by Bauwens et al. (2016) and Kyul et al. (2019) and is almost equal to Wu et al. (2013).

Another advantage of the method proposed in this paper is that the extraction process is completely automated except for the user-defined parameters needed for some steps. Because the method depends on the inherent geometric form of the tree, it is not prone to false detections from roadside objects that are not trees. Additionally, the algorithm can accurately extract seven different tree measures from an MTLS point cloud which is unique among other methods. The DRE measure is an important parameter for road safety analysis and has not been investigated in previous studies.

5. Conclusions

Having an efficient method to accurately identify, map, and categorize trees adjacent to roads can enhance roadside management and potentially lead to improved road user safety. This paper presents a novel method for automatically extracting trees from MTLS point clouds. The method consists of four steps: (1) preprocessing, (2) trunk extraction, (3) foliage extraction, and (4) characteristics measuring. The method was evaluated on two point cloud datasets. The average error in the extracted height of trees and other characteristics was less than 15 cm, and the average error of the TDBH was less than 10 cm. Accurate results were achieved for all of the trees regardless of their geometric shape and size. Additionally, the algorithm was able to distinguish tree foliage between closely spaced trees.

The method performed well in identifying trees in the presence of other planimetric features such as utility poles and billboards however the trees in the datasets used in this study had tree trunks that had circular geometric shapes. Future research should consider additional datasets that may not have unique tree geometry. Leveraging multi-spectral/photographic data combined with MTLS point clouds may lead to increased robustness of the method. Finally, it is suggested that by the generalization of the proposed method, an algorithm be introduced for road-side shrub extraction.

Declaration of Competing Interest

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

References

Atherton, T.J., Kerbyson, D.J., 1999. Size invariant circle detection. Image Vis. Comput. 17, 795–803.
Aurenhammer, Franz, Klein, Rolf, Lee, Der-Tsai, 2013. Voronoi Diagrams and Delaunay triangulations. World Scientific Publishing Company.
Ayala-Ramirez, V., Garcia-Capulin, C.H., Perez-Garcia, A., Sanchez-Yanez, R.E., 2006. Circle detection on images using genetic algorithms. Pattern Recogn. Lett. 27, 652–657.
Bauwens, S., Bartholomeus, H., Calders, K., Lejeune, P., 2016. Forest inventory with terrestrial LiDAR: A comparison of static and hand-held mobile laser scanning. Forests 7, 127.
Bendigeri, V., Ogle, J.H., Sarasua, W.A., 2011. Analysis of Factors Contributing to Roadside Tree Crashes in South Carolina Using Laser Data Collection Technologies.
Böhm, J., Brédif, M., Gierlinger, T., Krämer, M., Lindenbergh, R., Liu, K., Michel, F., Sirmacek, B., 2016. The IQmulus urban showcase: Automatic tree classification and identification in huge mobile mapping point clouds, International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives. International Society of Photogrammetry and Remote Sensing (ISPRS) 301–307.
Brack, C., 2009. Standard point on tree bole for measurement. Forest measurement and modelling. Computer-based course resources for Forest Measurement and Modeling (FSTY2009) at the. Australian National University, Canberra, Australia.
Bryant, D., 1989. Optimal Approximations by Piecewise Smooth Functions and Associated Variational Problems.
Cabo, C., Ordoñez, C., García-Cortés, S., Martínez, J., 2014. An algorithm for automatic detection of pole-like street furniture objects from Mobile Laser Scanner point clouds. ISPRS J. Photogramm. Remote Sens. 87, 47–56.
Chan, T.F., Vese, L.A., 1977. Active Contours Without Edges. Br. Dent. J. 142, 73.
Chen, Y., Wang, S., Li, J., Ma, L., Wu, R., Luo, Z., Wang, C., 2019. Rapid Urban Roadside Tree Inventory Using a Mobile Laser Scanning System. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12, 3690–3700.
Eck, R.W., McGee, H.W., 2008. Vegetation control for safety, a guide for local highway and street maintenance personnel. United States. Federal Highway Administration, Office of Safety.
Fan, W., Chenglu, W., Jonathan, L., 2016. Automated extraction of urban trees from mobile LiDAR point clouds. 9901, 99010P.
Forsman, M., Holmgren, J., Olofsson, K., 2016. Tree stem diameter estimation from mobile laser scanning using line-wise intensity-based clustering. Forests 7, 206.
Holopainen, M., Kankare, V., Vastaranta, M., Liang, X., Lin, Y., Vaaja, M., Yu, X., Hyyppä, J., Hyyppä, H., Kaartinen, H., Kukko, A., Tanhuanpää, T., Alho, P., 2013. Tree mapping using airborne, terrestrial and mobile laser scanning – A case study in a heterogeneous urban forest. Urban For. Urban Greening 12, 546–553.

Husain, A., Vaishya, R.C., 2019. Detection and thinning of street trees for calculation of morphological parameters using mobile laser scanner data. Remote Sens. Appl.: Soc. Environ. 13, 375–388.
Kumar, G., Patil, A., Patil, R., Park, S., Chai, Y., 2017. A LiDAR and IMU Integrated Indoor Navigation System for UAVs and Its Application in Real-Time Pipeline Classification. Sensors 17, 1268.
Kyul, H., Dong, H., Lee, K., Han, J., James, P., 2019. Estimating the heights and diameters at breast height of trees in an urban park and along a street using mobile LiDAR. Landscape Ecol. Eng.
Leverett, B., Bertolette, D., 2015. American Forests-Measuring Guidelines handbook. Retrived from http://www.americanforests.org/wp-content/uploads/2014/12/AF-Tree-Measuring-Guidlines_LR.pdf.
Li, Q., Yuan, P., Liu, X., Zhou, H., 2020. Street tree segmentation from mobile laser scanning data. Int. J. Remote Sens. 41, 7145–7162.
Lindenbergh, R., Sirmacek, B., Herrero-Huerta, M., Wang, J., Berthold, D., Ebersbach, D., 2015. Automated Large Scale Parameter Extraction Of Road-Side Trees Sampled By A Laser Mobile Mapping System. International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences 40.
Othmani, A., Piboule, A., Krebs, M., Stolz, C., Voon, L.L.Y., 2011. Towards automated and operational forest inventories with T-Lidar.
Palo, S., Alto, P., 1988. Snakes : Active Contour Models. 331, 321–331.
Rastiveis, H., Shams, A., Sarasua, W.A., Li, J., 2020. Automated extraction of lane markings from mobile LiDAR point clouds based on fuzzy inference. ISPRS J. Photogramm. Remote Sens. 160, 149–166.
Ronfard, R., 1994. Region-based strategies for active contour models. Int. J. Comput. Vision 13, 229–251.
Rutzinger, M., Pratihast, A.K., Oude Elberink, S., Vosselman, G., 2010. Detection and modelling of 3D trees from mobile laser scanning data. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci 38, 520–525.
Safety, T.O.T.F.f.R., 2011. Roadside design guide. AASHTO.
Sebastian, A., Flemons, D., 1981. NHTSA's Fatal Accident Reporting System.
Shams, A., Sarasua, W.A., Famili, A., Davis, W.J., Ogle, J.H., Cassule, L., Mammadrahimli, A., 2018. Highway Cross-Slope Measurement using Mobile LiDAR. Transp. Res. Rec. 2672, 88–97.
Shokri, D., Rastiveis, H., Shams, A., Sarasua, W., 2019. Utility Poles Extraction From Mobile Lidar Data In Urban Area Based On Density Information. Int. Arch. Photogram., Remote Sens. Spat. Inform. Sci.
Sirmacek, B., Lindenbergh, R., 2015. Automatic Classification Of Trees From Laser Scanning Point Clouds. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial. Inf. Sci. 2.
Stal, C., Verbeurgt, J., De Sloover, L., De Wulf, A., 2020. Assessment of handheld mobile terrestrial laser scanning for estimating tree parameters. J. For. Res. 1–11.
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, 584–611.
Xu, S., Xu, S., Ye, N., Zhu, F., 2018. Automatic extraction of street trees' nonphotosynthetic components from MLS data. Int. J. Appl. Earth Obs. Geoinf. 69, 64–77.
Yan, Z., Liu, R., Cheng, L., Zhou, X., Ruan, X., Xiao, Y., 2019. A Concave Hull Methodology for Calculating the Crown Volume of Individual Trees Based on Vehicle-Borne LiDAR Data. Remote Sensing 11, 623.
Yang, B., Fang, L., Li, J., 2013. Semi-automated extraction and delineation of 3D roads of street scene from mobile laser scanning point clouds. ISPRS J. Photogramm. Remote Sens. 79, 80–93.
Yao, W., Fan, H., 2013. Automated Detection of 3D Individual Trees Along Urban Road Corridors by Mobile Laser Scanning. International Symposium on.
Yu, Y., Guan, H., Li, D., Jin, C., Wang, C., Li, J., 2019. Road Manhole Cover Delineation Using Mobile Laser Scanning Point Cloud Data. IEEE Geosci. Remote Sens. Lett. 1–5.
Yu, Y., Li, J., Guan, H., Wang, C., Cheng, M., 2012. A Marked Point Process for Automated Tree Detection from Mobile Laser Scanning Point Clouds. positions 4, 5.
Yue, G., Liu, R., Zhang, H., Zhou, M., 2015. A Method for Extracting Street Trees from Mobile LiDAR Point Clouds. 204–209.
Zaboli, M., Rastiveis, H., Shams, A., Hosseiny, B., Sarasua, W., 2019. Classification of mobile terrestrial Lidar point cloud in urban area using local descriptors. Int. Arch. Photogram., Remote Sens. Spat. Inform. Sci. 42, 1117–1122.
Zhang, C., Zhou, Y., Qiu, F., 2015. Individual tree segmentation from LiDAR point clouds for urban forest inventory. Remote Sens. 7, 7892–7913.
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, 774–787.
Zhong, R., Wei, J., Su, W., Chen, Y.F., 2013. A method for extracting trees from vehicle-borne laser scanning data. Math. Comput. Modell. 58, 727–736.