Estimating Tree Structural Parameters via Automatic Tree Segmentation From LiDAR Point Cloud Data

Kenta Itakura, Satoshi Miyatani, and Fumiki Hosoi

***Abstract*—In this article, we proposed an automated tree segmentation method using light detection and ranging (LiDAR) point cloud data. Tree segmentation was performed accurately even with bumpy ground, and was validated on more than 1000 samples. For example, 371 out of 374 trees were detected from dataset 2, and the error was caused by the trees with low point densities located in the area far from the LiDAR. Segmentation was accurately performed, including the branches, leading to the retrieval of high-level parameters such as the leaf areas. To obtain the parameters regarding the leaf area from the segmented trees, a method for classifying the leaf and branch points in the three-dimensional point clouds obtained using a terrestrial LiDAR method was proposed. After preprocessing the input point cloud, such as by voxelization, the fast point feature histogram (FPFH) features were calculated. Then, the classifier for classification into leaves and branches was trained using the training dataset to calculate the test accuracy with the test data. Moreover, an unsupervised method for classification using the FPFH feature and *k*-means algorithm was also performed. Consequently, the recall and precision values of the classification were determined as 98.14% and 96.03%, respectively, with the supervised approach.**

***Index Terms*—Instance segmentation, light detection and ranging (LiDAR), point cloud, tree segmentation.**

I. Introduction

MONITORING vegetation plays an important role in evaluating forest resources, environmental protection, and disaster surveillance. Further, the monitoring of vegetation is required in other fields, such as construction and insurance. For example, we must estimate the cost of cutting down trees before constructing a facility. In this case, it is necessary to estimate the number of trees and their density. In the monitoring of vegetation, tree structural parameters such as the tree trunk diameter, tree height, and leaf area density are important. Conventionally, the necessary measurements are performed manually, which is time-consuming and labor-intensive. One alternative is the use of a three-dimensional (3-D) scanner called light detection and ranging (LiDAR). LiDAR emits a laser beam to a region of interest, and 3-D information is obtained by calculating the time for the laser beam to return. Previous studies have reported on tree trunk diameters, tree heights, and leaf area densities using LiDAR [1]–[5]. The ability to measure plant structural parameters can contribute to many fields other than forest monitoring. For example, detecting changes between a LiDAR point cloud before and after a typhoon can be efficient for estimating the area of a corresponding landslide. Omasa *et al.* used LiDAR for the visualization of an urban park and for the quantification of the biophysical variables of trees in the park. A digital canopy height model and digital terrain models were generated using airborne scanning [6]. Holopainen *et al.* [7] evaluated the accuracy and efficiency of airborne laser-scanning (ALS), terrestrial laser-scanning (TLS), and mobile laser-scanning methods for use in urban tree mapping and monitoring. Their results showed the potential of ALS for updating existing tree maps and monitoring the urban forest environment. The laser scanners for airborne LiDAR are mounted on helicopters, aircraft, and drones.

After the LiDAR measurement for tree monitoring, segmenting each tree in the LiDAR point cloud is necessary to automatically acquire the information of each tree, such as the number of trees, tree trunk diameters, and heights. Some methods are available for the tree segmentation of point clouds obtained mainly by airborne LiDAR. Many efforts have been made toward tree segmentation, mainly using airborne LiDAR data. As the data from airborne LiDAR can record the top of a tree canopy, the treetop method is popular. Hosoi *et al.* [8] performed crown-extraction filtering to accurately determine tree apex positions. Luo *et al.* [9] used a top-to-bottom region growing algorithm to segment individual trees from a LiDAR point cloud. Prieditis *et al.* [10] used a local maximum method with a Gaussian filter for the ALS. Another example is a method using a watershed algorithm or its updated version [11].

However, when LiDAR data are obtained on the ground using terrestrial LiDAR or mobile LiDAR, the laser beam emitted from the LiDAR cannot reach the treetop; accordingly, the point cloud data tends to lack data for the treetop. As a result, the corresponding tree detection method, as previously mentioned, does not work well. For tree segmentation in LiDAR point cloud data obtained on the ground, Lovell *et al.* [12] utilized the characteristics of trunks, i.e., they are solid targets with spatial continuity in the vertical direction (and to some extent in the horizontal direction). Then, the tree stems were detected using the intensity information of the point cloud. However, the intensity information may not be available depending on the LiDAR used, and the intensity values located far from the LiDAR tend to be lower than closer ones. Itakura and Hosoi [13] focused on the tree trunk part; the tree trunk part was detected after noise filtering and was expanded upward to segment the trees.

Manuscript received November 4, 2021; revised November 20, 2021 and November 30, 2021; accepted December 12, 2021. Date of publication December 15, 2021; date of current version January 5, 2022. This work was supported by the JSPS KAKENHI under Grant JP19J22701. *(Corresponding author: Fumiki Hosoi.)*

Kenta Itakura and Fumiki Hosoi are with the Graduate School of Agricultural and Life Sciences, The University of Tokyo, Tokyo 113-8657, Japan (e-mail: itakura-kenta095@g.ecc.u-tokyo.ac.jp; ahosoi@g.ecc.u-tokyo.ac.jp).

Satoshi Miyatani is with the ScanX Company, Ltd., Tokyo 100-0004, Japan (e-mail: satoshim@scanx.jp).

Digital Object Identifier 10.1109/JSTARS.2021.3135491

A method of segmenting trees using the tree trunk part was also reported in [14]. However, methods focusing on the tree trunk do not work well when there are many understories or shrubs around the tree trunk. Other studies developed an algorithm to adapt to a more complex environment involving many nontree objects [15], [16]. However, when nontree objects were included in the target point cloud data, many objects were misdetected as trees using the methods mentioned above. For example, a utility pole was detected as a tree, resulting in a false positive. For example, one method first detected the candidate trees, and then reduced the nontrees from the candidates in urban environments. Each of the detected objects was projected onto a 2-D image and classified as trees or nontrees. The trees in the cities were then detected. Although these methods are effective, including for scenes with nontree objects, detection cannot be performed if the tree trunk is not observed in the point cloud. Further, they should be tested with bumpy grounds and with a larger amount of data.

Recently, a method based on deep learning for point clouds has been studied. In a well-known network called PointNet [17], a pointwise classification called semantic segmentation is performed. After semantic segmentation, each point is classified into categories such as vegetation, building, wires, and car. However, this has mainly been tested in urban areas, and the feasibility of these deep learning-based methods in areas where trees are densely planted, such as in forests, remains unknown. Notably, additional postprocessing steps are needed after semantic segmentation to segment the trees; because semantic segmentation makes a pointwise classification, but it does not segment each tree. Another method combines an image-processing algorithm and PointNet for tree detection [18]. In the corresponding study, a candidate tree was selected using an image processing algorithm (i.e., the watershed algorithm), and then the nontree objects were removed using PointNet. However, as this method separated each tree using the watershed algorithm, it was difficult to find densely planted trees, such as those in forests. A method utilizing stereo vision and object detection based on deep learning has also been reported [19]. In this study, tree trunks were recorded by using a stereo camera to reconstruct 3-D information using stereo vision. Then, the tree trunk was detected using the "You Only Look Once" (YOLO) v2 network [20] to estimate the tree trunk diameter and tree species. However, the stereo camera could scan only trees located close to the camera, and was very unstable outdoors. Therefore, a tree segmentation method that can be used for tree point clouds even when the tree trunk is not scanned or when understories exist around the tree trunks should be tested, and it should be validated with a greater number of tree samples to confirm the feasibility of the method. After this segmentation, tree structural parameters such as the tree trunk diameter and tree height can be estimated automatically.

Furthermore, accurate tree segmentation leads to the retrieval of richer information, such as branching patterns and leaf areas. The separation of the leaf and branch areas is helpful when comparing tree growth under different conditions. For example, the differences in growth owing to different growth conditions can be reflected by the rates of increase in the leaf areas. When a difference is observed, it results in an understanding of the interactions between the environmental conditions and tree structure. The leaf area is also an important parameter for evaluating health conditions and growth stages. To improve the accuracy of the estimation of parameters such as leaf area, separating the leaf and branch points is necessary. For example, if the branches are not excluded from the leaf area estimation, the parameters regarding the leaf area will be overestimated. In addition to the estimation of the plant structural parameters, separation is preferable for other tasks, such as for calculating the radiative transfer model and evaluating the green landscape. Because leaves and branches have different spectral properties, distinguishing between leaves and branches is better for the model calculations.

In prior studies for extracting branches from point cloud data, several initial voxels corresponding to stems were selected, and branches were picked up from the voxelized 3-D tree model. Then, voxels adjacent to the initial voxels were searched for, and the neighboring voxels were categorized as stems and branches [21]. However, this method required the manual selection of the initial points. Iida *et al.* [22] separated leaf and branch points on the 3-D point cloud data of two evergreen trees obtained using a dual-wavelength portable scanning LiDAR system by considering the reflection intensity of the red and near-infrared spectra. Béland *et al.* [23] used the absorptance at 1535 nm for thresholding to separate leaves and branches. However, the LiDAR-based methods able to record the spectra of near-infrared waves are limited. In other approaches, hand-crafted geometric features such as the density have been calculated and fed into a classifier, such as a random forest, to classify the points into leaves and branches [24], [25]. Su *et al.* [26] combined classification and segmentation methods based on the spatial distribution of the point clouds of trees, and on the differences in the structures of the wood and leaf components of the trees. However, the classification accuracy using these methods was tested with only a few trees, or the training and test data were derived from the same tree. To expand the application of the method, training the classifier with several trees and validating the classifier with other trees is required.

Depending on the target tree, type of LiDAR system, measurement conditions, and parameters, the pattern of points of the target trees varies, even for identical trees. This implies that a pretrained classifier is difficult to apply to a new dataset in such cases and that the training dataset should be prepared under each experimental condition. Therefore, it is preferable if the classifier is not required to conduct training with the training data, i.e., that the classification can be performed using an unsupervised approach that does not require preparation for the training dataset. If a good feature for describing the 3-D geometry of the points is defined to represent the relationship among neighboring points, the classification of the branch and leaf can be performed. After classification of the leaf and branch points in the 3-D point clouds, a more detailed analysis can be performed if each branch can be segmented. In a previous study, an approach was presented for extracting a branching architecture from leafless apple trees [27]. As different trees have individual leaf types, as a next step, a branch extraction with leaf points should be tested.

To conduct such a detailed analysis, approaches for tree segmentation from LiDAR point clouds with high accuracy and robust leaf and branch separation algorithms are required. Therefore, in this study, an accurate method for tree segmentation from LiDAR point clouds was proposed, and the feasibility of the method was validated over 1000 samples. Then, a method was proposed for separating the leaf and branch points on the 3-D point cloud data of the trees with supervised and unsupervised approaches, aiming to enable the extraction of high-level tree structural parameters from the isolated tree point clouds.

II. Materials and Method

A. Materials

We used five study sites for the LiDAR measurements to validate the tree segmentation method. Seven datasets were obtained from the study sites. The first study site was a study plot in a forest research center where Japanese cypress was planted (*Chamaecyparis obtuse*) in Okuono, Kochi prefecture, Japan. This measurement was performed with a TLS for three areas to obtain three datasets (datasets 1–3). The laser scanner used in the TLS was a UTM-30LX-EW (Hokuyo Automatic Co., Ltd., Japan). The LiDAR was installed on the ground at intervals of 10 m in *x* and *y* directions. The details of the study site and data acquisition are provided in [28]. Next, a part of a mountain was used for tree measurement for dataset 4. The mountain was located in the southern part of the Tokushima prefecture, Japan. The LiDAR used was a GLS2000 (Tokyo, Japan). The TLS measurement was conducted several times, both on the ground and hills; then, these point clouds were registered into a single point cloud. Such a registration allowed for the recording of points even on the building roofs. The study site where dataset 5 was obtained was the Shinjuku Gyoen National Park in Tokyo, Japan. The target tree species was cherry blossom (*Cerasus Mill.*). The LiDAR used was a Velodyne VLP16 (Velodyne Lidar, Inc., USA), and the LiDAR measurements were performed while walking. This LiDAR scanned the target at 10 Hz to obtain a sparse point cloud. We used a method called simultaneous localization and mapping (SLAM) to build dense point cloud data [29]. Part of this dataset was used in a previous study [15]. Dataset 6 was acquired from a mountain in Izunokuni city, Japan. The LiDAR used was a GLS2000 (Tokyo, Japan). The trees were planted on a steep ground. Dataset 7 was obtained from Kawakami Park, Tsukuba City, Ibaraki, Japan, where the dominant tree species were Japanese cedar (*Cryptomeria japonica [L.f.] D. Don*), Japanese red pine (Pinus densiflora Siebold & *Zuccarini*), ginkgo (Ginkgo biloba *Linnaeus*), and Japanese zelkova (*Zelkova serrata [Thunberg] Makino*) [30]. The LiDAR used was the same as in the third dataset, and this data was also taken while walking. In this area, an understory and shrubs existed on the ground.

B. Dataset for Branch and Leaf Segmentation

In this section, the dataset used for the branch and leaf separation is explained. To evaluate the performance of the branch and leaf separation, another dataset with branch/leaf label information (other than the LiDAR data mentioned in the previous section) was used. The accuracy assessment was performed with the dataset to confirm the feasibility, and the method was adopted for the tree segmentation result with the dataset, as shown in the last section. The first dataset used was the LeWos dataset [31], [32]. This dataset was derived from the tress in the eastern part of Cameroon and includes 15 tropical species, such as Terminalia superba Engl. et Diels and Triplochiton scleroxylon K. Schum. For details, please refer to [31]. This dataset contains 61 LiDAR data points from a single tree, without color information. The label information of the branch or leaf was provided for each point. For the accuracy assessment of the leaf/branch separation and to compare with prior studies, the dataset was used in this study, not for tree segmentation (into single wood).

The second dataset we used was the LiDAR data of the American tulip tree (*Liriodendron tulipifera*), as measured using a terrestrial LiDAR system (LPM-25HA, RIEGL, Australia). The height and diameter at breast height were 30 and 1.5 m, respectively. The measurements were performed in February and June 2005, corresponding to the deciduous and leafy seasons, respectively. The two 3-D point clouds measured during the different seasons were registered so that the points corresponding to the leaf and branches were distinguished by differentiating between the two LiDAR data with and without leaves. These data were also used in [21].

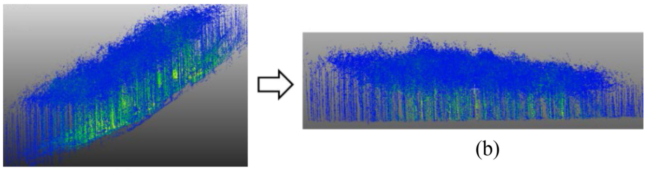

Fig. 1. Normalization of the height. The input point cloud is shown in panel. (a) The normalization process transforms the input point cloud into the one in panel. (b) The nearest ground point was found to each nonground points. Then, by subtracting the *z*-value of the nearest ground, the height information was normalized.

C. Tree Detection

The first step in tree segmentation was to extract the ground. The input point cloud was voxelized at intervals of 5 cm. In this process, each *xyz* value of the points was rounded off to the nearest integer. Next, a simple morphological filter [33] was used to exclude the ground. Initially, the input point clouds were converted into a regularly spaced grid in the *xy* coordinate plane as if the input point cloud was divided from the top view, so as to calculate the minimum *z*-value in each grid. Morphological opening operations were then conducted to detect the outliers and objects. The missing value, which we called the NAN value, was interpolated using an image inpainting method [34]. The ground was identified based on height information, considering the thresholds of the tree height and terrain slope. The thresholds were set at approximately 1.0 and 0.5 m, respectively. The grid resolution was set at 1.0 m. The input point cloud was separated into ground and nonground points.

After the ground extraction, the height of each point was normalized based on the ground information, as shown in Fig. 1. In this step, the baseline of each tree was normalized. For this purpose, the following process was performed: each nonground point was selected, and the nearest point belonging to the ground was searched. The difference between the *z*-value of the point of interest and the nearest ground point represented the tree height. The baseline of each tree was normalized by subtracting the *z*-value of the ground point. By repeating this process for all nonground points, the trees similar to those on the flat ground were obtained, as shown in Fig. 1(b).

Then, the region whose height ranged from 1.0 to 2.0 m was selected, the points were clustered based on the Euclidian distance, and the regions were expanded upward. This process was performed according to previous studies [13], [15]. In this process, the points whose distances were less than 1 m were assigned to the same cluster. By repeating this process, clusters corresponding to tree trunks were created. In this process, the tree trunk region appeared to grow upward to segment each tree.

Moreover, postprocessing proceeded as shown in the following. After the tree segmentation process as previously mentioned, the watershed algorithm was applied to a certain region. If there were excessive understories or noise around the tree trunks, a collection of the tree trunks could be recognized as a single cluster, leading to a failure to detect the trees. Accordingly, the condition for which this postprocessing of the watershed algorithm was performed was as follows. Each segmented tree in the last step was selected, and the length of the crown diameter was calculated. Then, if the length was greater than the threshold, such as 30 m, the watershed algorithm was performed. This postprocessing was performed only when the length corresponding to the crown diameter was comparatively large. In the postprocessing, first, the 3-D point cloud was projected onto a 2-D space, such as in a depth image. Next, the input image was segmented based on the local minimum. The watershed algorithm was used for the tree crown segmentation. For comparison with our method, the watershed method was used to segment the trees. Notably, our method is a hybrid of tree detection and watershed algorithms. Here, we attempt to compare the hybrid method with the watershed algorithm itself.

D. *Estimation of Tree Trunk Diameter*

After tree segmentation, the tree trunk diameter estimation was performed for the segmented trees.

This experiment was conducted with datasets 1–3, as mentioned in Section II-A. Each tree trunk diameter was measured using a tree caliper. For the estimation, the points of the tree trunks whose heights were between 0.8 and 1.6 m (meaning the average height of the points was approximately 1.2 m) were collected, and the points were projected onto a 2-D space to perform ellipse fitting. A total of 237 trees were randomly selected for the tree trunk diameter estimation. Samples with fewer points were not chosen.

As a preprocessing step prior to the fitting, the input point clouds corresponding to the tree trunks were clustered based on the Euclidian distance to create a group of close points. The neighboring points whose Euclidian distances were less than 3 cm were clustered into the same group. Then, ellipse fitting was performed for the largest group. The minor axis of the ellipse after fitting was used to estimate the tree trunk diameter. After the calculation, the estimated and actual values of the tree trunk diameter were compared for evaluation.

E. *Classification of Leaf and Branch Points in the LiDAR Point Clouds*

As a preprocessing step, the LiDAR point clouds of the tree were downsampled to a resolution of 1 cm. After preprocessing, a fast point feature histogram (FPFH) [35] was calculated for each point. The 3-D feature called the FPFH is known for its effectiveness in object detection and classification. It can be defined as the location of the point of interest and the locations of the neighboring points and is used to describe the local geometry. The FPFHs of 33 dimensions were input to a classifier to return the class of the point. The number of samples was equal to the number of points after the downsampling. The data were divided into training and test datasets at a ratio of 8:2. For classification, a discrimination analysis was performed. Other classifiers such as support vector machines were also tested, but the performance of the discrimination analysis was better. The workflow of this study is illustrated in Fig. 2.

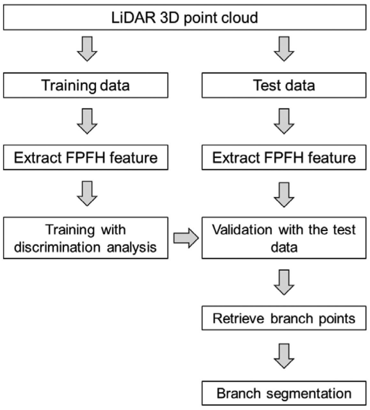

Fig. 2. Workflow of the leave/branch separation with FPFH.

For the parameter setting of the classifier, the training data were divided at a ratio of 7:1, and the greater portion was used for training. A smaller portion was used for validation, i.e., to check the classification accuracy. The training and validation datasets were combined after optimizing the training parameters. The classification accuracy was then calculated using the test dataset.

An identical classifier was applied to the second dataset, as explained in the previous section; this classifier was significantly different from the training data. For the tulip tree in dataset 2, the FPFH feature was calculated, and classification was performed using the training dataset called dataset 1. The classification accuracy was then examined.

In addition to the supervised learning method for the tulip tree, unsupervised learning was performed for the classification of the leaves and branches. The FPFH features were extracted as discussed in the previous section. The *k*-means algorithm was then used to separate them into two clusters. As the target classes were leaves and branches, the class number was two. The distance used was the squared Euclidean distance. The MATLAB 2020b software (MathWorks, USA) was used for the analysis.

F. *Separation of Each Branch With the Branching Points*

Next, each branch was segmented from the point cloud data of the tree. In this step, the leaves and branches in the 3-D LiDAR point cloud were already separated, as explained in the previous section. The branch segmentation method is shown in Fig. 3(a). After the voxelization of the input point cloud, as shown in Fig. 3(a), thinning of the branches was performed using the

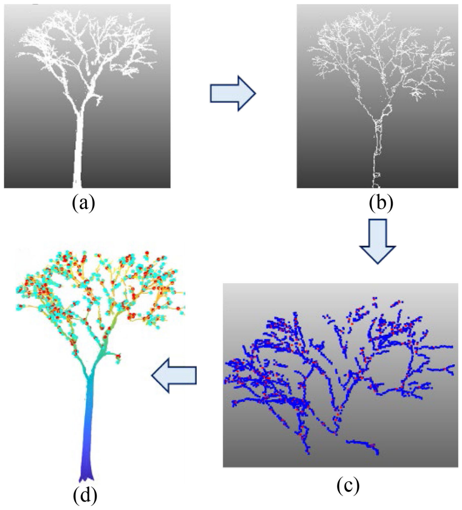

Fig. 3. Workflow for the branch segmentation.

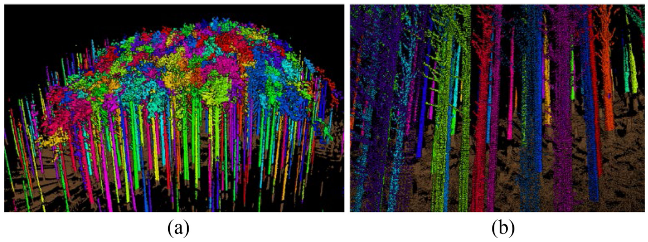

Fig. 4. Tree segmentation result. Each tree was illustrated by different colors as shown in panel. (a) Panel (b) shows the close-up view of the segmentation result under the canopy.

method proposed by Lee *et al.* [36] and Kerschnitzki *et al.* [37], as shown in Fig. 3(b). Then, the number of voxels at each level in the vertical direction was calculated after thinning. Forty percent of the maximum number was set as a threshold, and the number of voxels at each level was scanned from top to bottom. When the number of voxels was lower than the threshold, this procedure was terminated, and the part above was used for the analysis. Then, each voxel whose number of neighboring voxels was more than three was considered as a branching point, as shown in Fig. 3(c). Finally, the branch point determined in the previous step was deleted, and each branch was separated by expanding each region if there were neighboring voxels (region-growing algorithm). In Fig. 3(d), the branching and end points of the branch are colored red and cyan, respectively.

After segmenting the branches, the inclination angle of each branch was calculated using principal component analysis. The angle with the z-axis was calculated, implying that the inclination angle of the branch parallel to the ground was 90°. The centroid of the target tree was calculated, and if the branch was inclined to the ground from the centroid, the inclination angle was expected to be between 90° and 180°.

III. Results

A. Tree Segmentation

Fig. 4(a) shows a typical example of the tree segmentation. Each tree is illustrated by different colors. Fig. 4(b) denotes a close-up view under the tree canopy. As shown, each tree is segmented accurately, including the branches. Both high and low trees can be detected using our method, even when the trees are planted densely.

TABLE I
Tree Segmentation Result

Dataset	Total number	Omission	Commission
Dataset 1	183	2	0
Dataset 2	374	3	0
Dataset 3	282	3	0
Dataset 4	174	1	16
Dataset 5	140	0	5
Dataset 6	143	8	0
Dataset 7	75	2	7

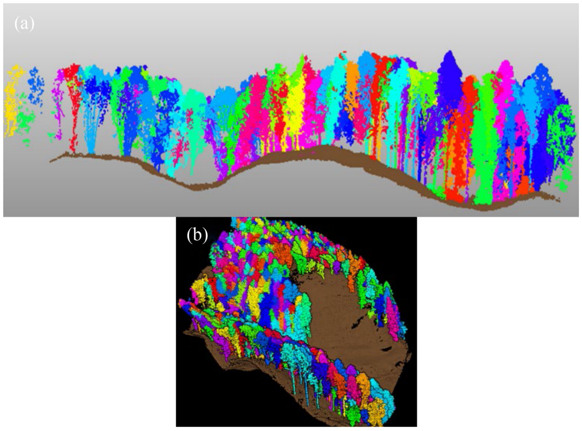

Fig. 5. Tree segmentation result when the ground is bumpy.

Table I presents the validation results for tree segmentation. Omission and commission refer to the number of cases where the tree could not be detected and where the object rather than a tree, such as a part of the tree canopy, was detected as a single tree. The accuracy varies according to the dataset. Overall, tree segmentation can be performed accurately.

Tree detection can be performed even when the tree trunk is not clearly observed, as shown in Fig. 5. Fig. 5(a) represents the cross section of the tree segmentation result. The tree segmentation is performed in the densely planted areas; simultaneously, the ground extraction is also accurately conducted. Fig. 5(a) and (b) shows that the ground is bumpy, but the ground is successfully segmented and colored as brown. Moreover, the trees whose trunks were not scanned are also successfully detected, as shown in panel (a).

Conventionally, tree segmentation is performed using a watershed algorithm, where the input point cloud is converted into a bird's eye view. The validation result with the watershed algorithm with dataset 1 is as follows. The total number of trees correctly segmented is 161 out of 181 trees, and 56 objects (such as parts of tree crowns) are incorrectly detected as trees. Because the LiDAR data in this study were obtained by TLS, the points

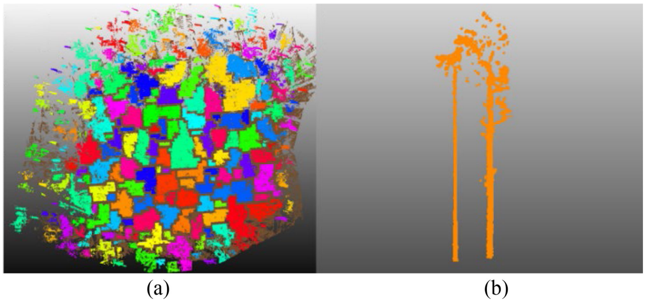

Fig. 6. Result of the ellipse fitting for tree trunk diameter estimation. Panels (a) and (b) show the result without and with preprocessing prior to the fitting (a clustering based on the Euclidian distance).

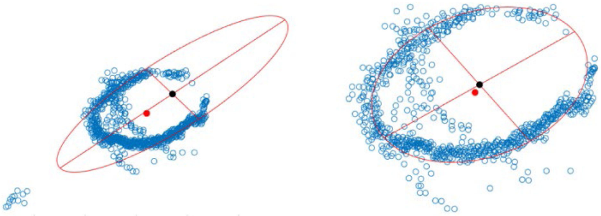

Fig. 7. Result of the leaf and branch classification. Panels (a) and (c) represent the input point cloud. Panels (b) and (d) show the results of the automatic classification. The leaf and branch points are shown in green and brown, respectively.

on the treetops are difficult to scan; accordingly, some trees are detected as a single tree, owing to the local maximum tree height of the neighboring trees.

B. Estimation of Tree Trunk Diameter

The mean absolute error of the tree diameter estimation via ellipse fitting is 2.4 cm ($N = 237$). Other metrics of relative bias, root-mean-square error (RMSE), and coefficient of variation of RMSE were –0.026, 0.031, and 8.6%, respectively. Fig. 6 shows the results from the ellipse fitting for the tree trunk diameter estimation. Panels (a) and (b) show the results without and with preprocessing prior to fitting, respectively. A small cluster can be found at the bottom left of panel (a). This is a noise or a fragment of the understory. A clustering based on the Euclidian distance was performed to reduce this noise, and only the main part of the trunk was used for ellipse fitting, as shown in panel (b).

C. Classification of Leaves and Branches

The overall accuracy of the classification into leaves and branches is 94.83%, and a typical example is illustrated in Fig. 7. Panels (a) and (c) show the point-cloud images of the input trees. These point clouds are fed into the classifier. Panels (b) and (d) show the results of the classification, in which the leaves and branches are illustrated in green and brown, respectively. The branches inside the canopy are classified accurately, and consequently, the branch shapes are clearly observed. Table II presents the confusion matrix of the classification results. The recall and precision values are 98.14% and 96.03%, respectively, when the leaf is considered as "positive." Conversely, when the branch is considered as "positive," the recall and precision values are 71.79% and 84.76%, respectively.

TABLE II
EVALUATION METRICS OF THE LEAF AND BRANCH CLASSIFICATION

	Leaf	Branch
Recall	0.9814	0.7179
Precision	0.9603	0.8476

The recall and precision were calculated when the leaf and branch were considered as "positive" data.

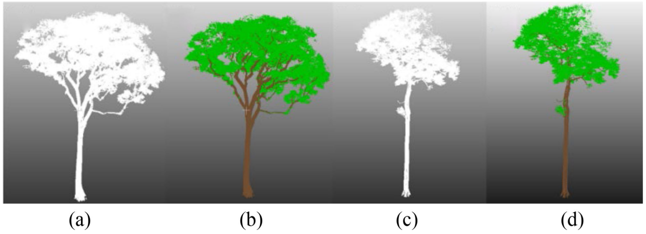

Fig. 8. Result of the classification of leaves and branches of tulip tree (*Liriodendron tulipifera*) in dataset 2 when the classifier was trained with dataset 1.

TABLE III
EVALUATION METRICS OF THE LEAF AND BRANCH CLASSIFICATION FOR THE TULIP TREE IMAGE WITH SUPERVISED AND UNSUPERVISED METHODS

	Supervised		Unsupervised	
	Leaf	Branch	Leaf	Branch
Recall	0.9935	0.5362	0.9916	0.5204
Precision	0.9647	0.8657	0.9634	0.8292

The recall and precision were calculated when the leaf and branch were considered as "positive" data.

Fig. 8 represents the result of the classification of leaves and branches of the tulip tree (*Liriodendron tulipifera*) in dataset 2. Panel (a) shows the results when the classifier is trained using different types of datasets. This result is listed in Table II under the column titled "Supervised." Panel (b) shows the results with the FPFH feature and k-means algorithm. This is denoted as "Unsupervised" in Table III. Panel (c) shows the separation of leaves and branches after tree segmentation, as shown in Section III-A. The tree segmentation and leaf/branch separation can be simultaneously performed. In computer memory, after each tree is divided, the leaf/branch separation is automatically completed without human intervention or manual input of the single tree point clouds, as shown in Fig. 8(c).

D. Branch Segmentation

After the classification of the branch and leaf points as shown in Section III-A, the branches were segmented as shown in Fig. 9. Each branch was illustrated randomly using different colors. The *xyz* coordinates were extracted from each segmented branch to calculate the inclination angle. Fig. 10 shows the distribution of the inclination angles in the trees as illustrated in Fig. 9. The histograms in Fig. 10(a) and (b) are derived from panels (a) and (c) in Fig. 9, respectively.

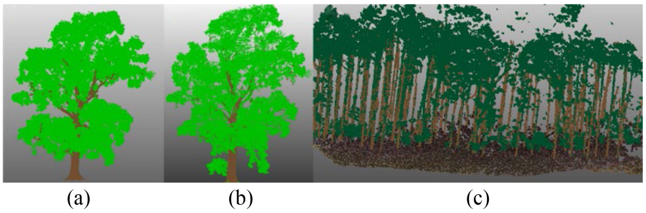

Fig. 9. Example of the branch segmentation. The leaf was automatically separated using the classifier trained with dataset 1 and branch segmentation was performed to the rest of the tree point cloud.

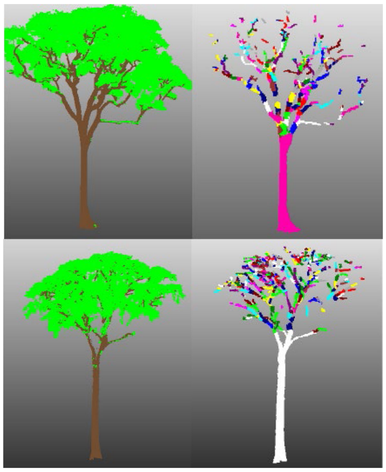

Fig. 10. Distribution of the inclination angle of the leaf branch. This distribution was calculated after the leaf/branch segmentation.

IV. Discussion

A. *Tree Detection*

The following list outlines the different types of graphics published in IEEE journals. They are categorized based on their construction and use of color/shades of gray.

In general, the tree segmentation can be performed accurately. As shown in Fig. 9(a), a tree whose tree trunks were not scanned could be detected. This type of tree cannot be detected with other widely used algorithms, such as cylinder fitting. Nevertheless, a tree composed of fewer points, such as the one far from the laser scanner, could potentially be dismissed and deleted, owing to the thresholding based on the number of points for noise reduction. This case would lead to an omission. To alleviate such omissions, the input point cloud was downsampled by voxelizing the point clouds. The voxelization of the input point clouds also contributed to speeding up the tree segmentation. Points with almost the same *xyz* value or duplicate points should be removed, as they do not contribute to accurate tree detection. In addition, noise such as the artifacts created by people moving through LiDAR and a part of a canopy located near the ground could be misrecognized as a tree trunk, leading to a commission. Another case of commission was observed when the trees were over-segmented with the watershed algorithm being performed as the postprocessing. In this study, the segmentation result was validated with more than 1000 trees, as presented in Table I, indicating that the method is more reliable than those employed in previous studies.

Without normalizing the heights of the trees as shown in Fig. 1, the tree trunk detection and expansion process for the segmentation would be difficult to perform, as each tree trunk has a different *z*-value. Thus, this process enabled the segmentation of the trees on bumpy ground. Moreover, the accurate ground extraction contributed to an accurate tree segmentation, because the height normalization could not be performed without an accurate ground extraction.

The postprocessing using the watershed algorithm enabled the segmentation of trees even if high understories existed around trees of interest; it also allowed for the detection of trees even when tree trunks could not be observed. In previous studies, methods using cylinder/circle fitting have been proposed. Although these methods are simple and easy to understand, tree trunks whose cross sections do not describe a complete circle cannot be detected. We compared our method with the watershed algorithm itself; then, the performance of our method was significantly higher. The tree point clouds measured on the ground tend to have stems and the segmentation utilizing the tree stems was effective; then, the performance was greatly better. We compared our method with the watershed algorithm, and confirmed that the performance of our method was significantly higher. The tree point clouds measured on the ground tend to have stems, and the segmentation utilizing the tree stems was effective; thus, the performance was considerably better.

For segmentation in a related topic, the leaf segmentation for the point cloud was performed using the watershed algorithm [13], [15]. Although point clouds contain more information than 2-D images in terms of structural and spatial aspects, simplified approaches such as 2-D projection can sometimes work more efficiently. In our case, the regions around the tree trunks can be noisy owing to the presence of scan artifacts, shrubs, or understories. Postprocessing with the watershed algorithm was therefore performed, utilizing the tree crown structure. A prior method [38] used an omnidirectional camera to capture trees, and 3-D reconstruction was performed using a photogrammetric approach called SfM-MVS. In this method, tree detection was performed with an omnidirectional image using a deep learning-based approach of YOLOv2. Although such tree detection from an image has a high accuracy compared to detection from a point cloud, it entails a high computational cost for the 3-D reconstruction of the SfM-MVS. Furthermore, 3-D reconstruction with LiDAR is more precise than that using a photogrammetric approach. In future work, we would like to construct a tree detection method by combining tree detection from an omnidirectional camera and 3-D reconstruction using LiDAR and SLAM.

A previous method called TreeSeg was reported as a method of segmenting trees in point clouds [39]. Two key enablers in this method are stem detection via cylinder fitting and ground identification using the random sample consensus (RANSAC) algorithm. However, stem detection is difficult when the stem point clouds are not clearly scanned, and ground identification is challenging when the ground is bumpy. In contrast, our method is fully automated with high accuracy, as tested with many datasets to demonstrate its feasibility, and can be used even when the tree stems are not clearly scanned, as shown in Fig. 5. In addition, trees on bumpy grounds can be successfully segmented. Moreover, tree segmentation to leaf/branch separation can be performed accurately using our method, which shows the effectiveness of our method for forest management.

Many efforts have been made for object detection in 2-D images using, for example, YOLO and single-shot detector algorithms. Some algorithms have utilized these object detection methods and made it possible to detect objects in a point cloud. For example, PointPillars [40] is famous for providing object detection from point clouds. Although we tested the algorithm for our dataset, accurate detection was difficult. It appeared that the density of the target trees was high and that the canopies of neighboring trees overlapped in many cases, resulting in failures of tree detection when using deep learning methods for object detection. Nevertheless, the methodology based on deep learning with point clouds was rapidly developed. We would like to continue testing such deep learning-based methods in future studies. Because trees in forests are densely planted in many cases, the annotation required as a preparation for the training process takes a longer time, because annotation with high-density point clouds is laborious and time-consuming. This can be one of the drawbacks of deep learning methods.

As shown in Fig. 5(b), tree segmentation, even in the wild forest, works. If there are several understories and shrubs around tree stems, it is generally difficult to segment trees from point clouds. However, our method performed a watershed algorithm after tree segmentation as postprocessing, which enabled the separation of the unsegmented trees. However, we need to explore the feasibility of our algorithm for considerably wilder forests, such as in [42], which we included as our future work.

B. Estimation of Tree Trunk Diameter

A few centimeters of fluctuation in the tree trunk diameter measurement on site was inevitable, as the measurement result was prone to change depending on the points on which the tree caliper was touching. Therefore, the tree trunk diameter estimation needed to be performed accurately in this study. Noises and points of the understory were observed around the tree trunk, and those points were included in the tree points. When performing an ellipse fitting to the tree trunk area, the fitting result was influenced by those points, resulting in the failure of the fitting, as shown in Fig. 6(a). By reducing the noise with the Euclidian distance, the ellipse fitting became more accurate, leading to a highly accurate tree trunk diameter estimation. We validated the accuracy of tree trunk diameter with many samples ($N = 237$) to obtain higher reliability. For the tree trunk diameter estimation, a method using RANSAC is also available. In this method, a few points are randomly selected, and the fitting is performed. If the number of points within the fitting result is more than a certain point, the fitting is regarded as a correct detection. This method is particularly useful for finding an object of interest from a noisy environment including other objects. However, in this study, RANSAC sampled a few points and performed the fitting, i.e., all points were not used for the fitting, resulting in a lower fitting to the model. The cross section of the tree trunks was not always a complete circle, meaning that the circle fitting did not work in those cases. Thus, an accurate tree trunk diameter estimation was performed using the proposed method. Tree trunk diameter estimation was accurately performed after tree segmentation and leaf/branch separation. This parameter estimation leads to various applications, such as estimating biomass, forest resources, and biodiversity. Furthermore, by combining with the leaf/branch segmentation, the curvature of tree stems can be calculated, which allows for a more precise evaluation of the forest assets. As this study proposed an effective method from tree segmentation to tree trunk diameter estimation and leaf/branch separation, the LiDAR application in forests should be further encouraged.

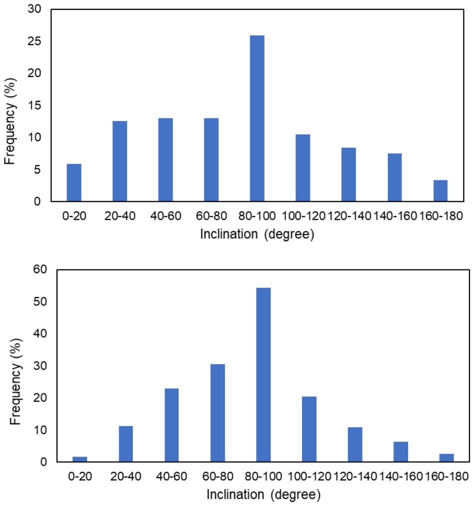

Fig. 11. Tree segmentation result with the watershed algorithm. Subfigure (a) represents the tree segmentation result using the watershed algorithm. Subfigure (b) shows a typical example of incorrect segmentation where two adjacent trees are detected as a single tree.

C. Separation of Leaf and Branch Points

The leaves in the tree crown, as well as the leaves in the middle part of the tree as shown in Fig. 7(d), could be accurately classified. The overall accuracy was higher than that obtained in a previous study [32]. The FPFH feature was calculated based on the locations of neighboring points. For example, the points corresponding to tree trunks tend to be arranged in a plane, whereas the leaf points tend to exist randomly in the leaf regions. Thus, the patterns of the points corresponding to the leaf and tree trunk are significantly different, enabling classification with high accuracy. We compared our method using FPFH with LeWos and found that the accuracy of our method is higher than that of LeWos. The article reported the result of a mean value of 0.91 ± 0.03 with a certain setting. Our method achieved an accuracy of 94.83% with the same dataset. Furthermore, we confirmed that the same classifier had a high accuracy with different datasets of different species and taken by different LiDAR, as presented in Table III. This suggests that our method can be used across different datasets, and can be applied to other point cloud files. When adding more training data to the classifier, a higher classification can be obtained.

When using supervised learning, if the new data to apply the classifier to are obtained from different research sites and different sensors, it becomes much more difficult to utilize the pretrained classifier. As we can observe from the result presented in Fig. 8(a), the leaf/branch classification could be performed accurately even for the point cloud data belonging to a different dataset from the training dataset, e.g., even when the classifier was applied to the tulip tree image completely different from the training dataset. This result shows the potential of applying the classifier to a new dataset. As it is significantly tedious to prepare a training dataset with labels for leaves and branches, if the classifier can be applied to a new dataset, it can act as a powerful tool for tree analysis.

Furthermore, the unsupervised learning for the leaf/branch classification was successful, as shown in Fig. 8(b). Identical classifiers are generally not applied to different species; thus, it is significant if unsupervised learning suffices for classification. The unsupervised method has the potential to be used for the new data because the leaf/branch classification is conducted based on the difference in the FPFH features within the target tree. The accurate tree detection proposed in this study leads to the retrieval of tree structural parameters such as tree trunk diameter and leaf area density by using leaf/branch separation. This method can also be combined with automatic tree segmentation.

Tree segmentation and branch leaf separation are important links in forest LiDAR point-cloud processing. As for the effectiveness and efficiency, our algorithm does not require any tree models to specify prior to the segmentation; moreover, the process from the tree segmentation to the leaf/branch separation and tree trunk diameter can be conducted at the same time without any manual selection. Further, we tested both with many datasets collected by TLS and mobile LiDAR to verify the effectiveness of our method. Our method links the two processes of tree segmentation and branch/leaf separation with high accuracy. For example, in many countries, the number of people managing forests has decreased, as the work of tree monitoring and management is very tedious and time-consuming. One of the hardest works is measuring tree properties individually, such as measuring tree trunk diameter, height, and tree volume. Our method can automate such work and demonstrate the feasibility of various types of datasets. This means that our method has a good potential for replacing such hard work. Furthermore, our study also shows a method for extracting branches/leaves automatically from the point cloud. This method can also contribute to forest research, not only for tree management.

D. Branch Segmentation

As shown in Fig. 9, each branch could be segmented accurately because the classification of branches and leaves was performed accurately. This segmentation was performed based on the fact that the branching point has multiple neighboring points (voxels) after thinning. When the branching points are reduced, the branches become isolated, and each branch can be segmented using a region-growing algorithm. However, the performance of this segmentation depends on parameters such as the voxel size; therefore, the parameters should be optimized based on the target trees.

As proposed above, this method automated the step from the classification of the leaf and branch points to the segmentation of the branches, and several applications can be considered based on this. For example, the distribution of the branch inclination angle can be calculated as shown in Fig. 11. After typhoons and storms, detecting changes in the inclination distribution can contribute to providing damage estimations for trees. This type of analysis is difficult to perform using satellite images; hence, this is an advantage of using LiDAR measurements in real time. A study on estimating the biomass from the length, tree trunk diameter, and inclination angle of each branch has also been reported [41]. In addition to biomass monitoring, information regarding branches and tree trunks can be used for regular tree inspections. Our method can be used to reveal the relationships between tree structures and properties.

Acknowledgment

The authors are very grateful to Dr. F. Kitahara in Forestry and Forest Products Research Institute, which offered the test sample for the study. The authors would like to thank Otake-gumi and Shoji-Gumi for providing the dataset for their work. The authors would like to appreciate H. Tran and H. Matsuda for their help in labeling and providing suggestions for this work.

References

[1] C. Chen, Y. Wang, Y. Li, T. Yue, and X. Wang, “Robust and parameter-free algorithm for constructing pit-free canopy height models,” *ISPRS Int. J. Geo-Inf.*, vol. 6, Jul. 2017, Art. no. 219.

[2] W. Yuan, J. Li, M. Bhatta, Y. Shi, P. Baenziger, and Y. Ge, “Wheat height estimation using LiDAR in comparison to ultrasonic sensor and UAS,” *Sensors*, vol. 18, Nov. 2018, Art. no. 3731.

[3] F. Hosoi and K. Omasa, “Voxel-based 3-D modeling of individual trees for estimating leaf area density using high-resolution portable scanning lidar,” *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 12, pp. 3610–3618, Dec. 2006.

[4] F. Hosoi and K. Omasa, “Estimating leaf inclination angle distribution of broad-leaved trees in each part of the canopies by a high-resolution portable scanning lidar,” *J. Agricultural Meteorol.*, vol. 71, no. 2, pp. 136–141, Jun. 2015.

[5] K. Itakura and F. Hosoi, “Estimation of leaf inclination angle in three-dimensional plant images obtained from lidar,” *Remote Sens.*, vol. 11, Feb. 2019, Art. no. 344.

[6] K. Omasa, F. Hosoi, T. M. Uenishi, Y. Shimizu, and Y. Akiyama, “Three-dimensional modeling of an urban park and trees by combined airborne and portable on-ground scanning LIDAR remote sensing,” *Environ. Model. Assessment*, vol. 13, pp. 473–481, Jul. 2008.

[7] M. Holopainen *et al.*, “Tree mapping using airborne, terrestrial and mobile laser scanning—A case study in a heterogeneous urban forest,” *Urban Forestry Urban Greening*, vol. 12, no. 4, pp. 546–553, Jul. 2013.

[8] F. Hosoi, H. Matsugami, K. Watanuki, Y. Shimizu, and K. Omasa, “Accurate detection of tree apexes in coniferous canopies from airborne scanning light detection and ranging images based on crown-extraction filtering,” *J. Appl. Remote Sens.*, vol. 6, no. 1, Mar. 2012, Art. no. 063502.

[9] L. Luo, Q. Zhai, Y. Su, Q. Ma, M. Kelly, and Q. Guo, “Simple method for direct crown base height estimation of individual conifer trees using airborne LiDAR data,” *Opt. Express*, vol. 26, no. 12, pp. A562–A567, Apr. 2018.

[10] G. Prieditis, I. Šmits, S. Daģis, and D. Dubrovskis, “Individual tree identification using combined lidar data and optical imagery,” presented at the Annu. 18th Int. Scientific Conf. “Research for Rural Development,” Jelgava, Latvia, May 16–18, 2012.

[11] B. Koch, U. Heyder, and H. Welnacker, “Detection of individual tree crowns in airborne lidar data,” *Photogramm. Eng. Remote Sens.*, vol. 72, no. 4, pp. 357–363, Apr. 2006.

[12] J. L. Lovell, D. L. B. Jupp, G. J. Newnham, and D. S. Culvenor, “Measuring tree stem diameters using intensity profiles from ground-based scanning lidar from a fixed viewpoint,” *ISPRS J. Photogramm. Remote Sens.*, vol. 66, no. 1, pp. 46–55, Sep. 2011.

[13] K. Itakura and F. Hosoi, “Automatic individual tree detection and canopy segmentation from three-dimensional point cloud images obtained from ground-based lidar,” *J. Agricultural Meteorol.*, vol. 74, no. 3, pp. 109–113, Jul. 2018.

[14] C. Cabo, C. Ordóñez, C. A. López-Sánchez, and J. Armesto, “Automatic dendrometry: Tree detection, tree height and diameter estimation using terrestrial laser scanning,” *Int. J. Appl. Earth Observ. Geoinf.*, vol. 69, pp. 164–174, Mar. 2018.

[15] K. Itakura and F. Hosoi, “Automated tree detection from 3D lidar images using image processing and machine learning,” *Appl. Opt.*, vol. 58, no. 14, pp. 3807–3811, May 2019.

[16] K. Itakura and F. Hosoi, “Three-dimensional tree monitoring in urban cities using automatic tree detection method with mobile LiDAR data,” *Intell. Informat. Infrastructure*, vol. 2, pp. 1–10, Nov. 2021.

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, “PointNet: Deep learning on point sets for 3D classification and segmentation,” in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 77–85.

[18] K. Itakura and F. Hosoi, “Tree detection from airborne LiDAR data using image processing and 3D deep learning,” *Intell., Informat. Infrastructure*, vol. 1, pp. 320–328, 2020.

[19] K. Itakura and F. Hosoi, “Simple and effective tool for estimating tree trunk diameters and tree species classification,” *Appl. Opt.*, vol. 59, no. 2, pp. 558–563, 2020.

[20] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, 2017, pp. 7263–7271.

[21] S. Negoro, F. Hosoi, and K. Omasa, “Extraction of nonphotosynthetic organs from portable scanning lidar data for a deciduous broadleaf tree in leafy season,” *Eco-Engineering*, vol. 24, no. 2, pp. 51–55, Apr. 2012.

[22] Y. Iida, F. Hosoi, and K. Omasa, “A study of a method for separation of plant tissues on dual wavelength portable scanning lidar data for evergreen tree,” *Eco-Engineering*, vol. 28, no. 4, pp. 113–118, Nov. 2016.

[23] M. Béland, J. L. Widlowski, R. A. Fournier, J. F. Côté, and M. M. Verstraete, “Estimating leaf area distribution in Savanna trees from terrestrial LiDAR measurements,” *Agricultural Forest Meteorol.*, vol. 151, no. 9, pp. 1252–1266, Jun. 2011.

[24] D. Wang, M. Hollaus, and N. Pfeifer, “Feasibility of machine learning methods for separating wood and leaf points from terrestrial laser scanning data,” in *Proc. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, Wuhan, China, 2017, pp. 157–164.

[25] J. Zhou, H. Wei, G. Zhou, and L. Song, “Separating leaf and wood points in terrestrial laser scanning data using multiple optimal scales,” *Sensors*, vol. 19, no. 8, Apr. 2019, Art. no. 1852.

[26] Z. Su, S. Li, H. Liu, and Y. Liu, “Extracting wood point cloud of individual trees based on geometric features,” *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 8, pp. 1294–1298, Feb. 2019.

[27] A. Bucksch and S. Fleck, “Automated detection of branch dimensions in woody skeletons of fruit tree canopies,” *Photogramm. Eng. Remote Sens.*, vol. 77, no. 3, pp. 229–240, Mar. 2011.

[28] F. Kitahara, T. Nishizono, K. Hosoda, and E. Kodani, “Comparison of forest measurement errors using two types of terrestrial laser scanning,” *Japanese J. Forest Plan.*, vol. 54, no. 1, pp. 63–66, Oct. 2020.

[29] J. Zhang and S. Singh, “LOAM: Lidar odometry and mapping in real-time,” in *Proc. Robot., Sci. Syst.*, 2014, vol. 2, pp. 1–9.

[30] F. Hosoi, S. Umeyama, and K. Kuo, “Estimating 3D chlorophyll content distribution of trees using an image fusion method between 2D camera and 3D portable scanning lidar,” *Remote Sens.*, vol. 11, no. 18, Sep. 2019, Art. no. 2134.

[31] S. Momo Takoudjou *et al.*, “Using terrestrial laser scanning data to estimate large tropical trees biomass and calibrate allometric models: A comparison with traditional destructive approach,” *Methods Ecol. Evol.*, vol. 9, no. 4, pp. 905–916, Nov. 2017.

[32] D. Wang, T. S. Momo, and E. Casella, “LeWoS: A universal leaf-wood classification method to facilitate the 3D modelling of large tropical trees using terrestrial LiDAR,” *Methods Ecol. Evol.*, vol. 11, no. 3, pp. 376–389, Dec. 2019.

[33] T. J. Pingel, K. C. Clarke, and W. A. Mcbride, “An improved simple morphological filter for the terrain classification of airborne LIDAR data,” *ISPRS J. Photogramm. Remote Sens.*, vol. 77, pp. 21–30, Jan. 2013.

[34] J. D’Errico, Inpaint_nans.m, 2004. [Online]. Available: http://www.mathworks.com/matlabcentral/fileexchange/4551. Accessed on: Oct. 18, 2021.

[35] R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, 2009, pp. 3212–3217.

[36] T. C. Lee, R. L. Kashyap, and C. N. Chu, “Building skeleton models via 3-D medial surface axis thinning algorithms,” *CVGIP, Graph. Models Image Process.*, vol. 56, no. 6, pp. 462–478, May 1994.

[37] M. Kerschnitzki *et al.*, “Architecture of the osteocyte network correlates with bone material quality,” *J. Bone Mineral Res.*, vol. 28, no. 8, pp. 1837–1845, Aug. 2013.

[38] K. Itakura and F. Hosoi, “Automatic tree detection from three-dimensional images reconstructed from 360 spherical camera using YOLO v2,” *Remote Sens.*, vol. 12, no. 6, Mar. 2020, Art. no. 988.

[39] A. Burt, M. Disney, and K. Calders, “Extracting individual trees from lidar point clouds using TreeSeg,” *Methods Ecol. Evol.*, vol. 10, no. 3, pp. 438–445, Nov. 2018.

[40] A. H. Lang, V. Sourabh, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast encoders for object detection from point clouds,” in *Proc. Comput. Vision Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 12697–12705.

[41] S. F. López-López, M. T. Tomás, B. M. Héctor, G. N. Moises, and M. Héctor, “Non-destructive method for above-ground biomass estimation of Fraxinus uhdei (Wenz.) Lingelsh in an urban forest,” *Urban Forestry Urban Greening*, vol. 24, pp. 62–70, May 2017.

[42] X. Ge and Q. Zhu, “Target-based automated matching of multiple terrestrial laser scans for complex forest scenes,” *ISPRS J. Photogramm. Remote Sens.*, vol. 179, pp. 1–13, Sep. 2021.

Kenta Itakura received the B.E. degree in agricultural and environmental engineering from Kyoto University, Kyoto, Japan, in 2017, and the M.E. degree in biological and environmental engineering from The University of Tokyo, Tokyo, Japan, in 2019, where he is currently working toward the Ph.D. degree in biological and environmental engineering with the Department of Agricultural and Life Sciences.

Satoshi Miyatani received the B.E. and M.E. degrees in aeronautics and astronautics from the University of Tokyo, Tokyo, Japan, in 2013 and 2015, respectively, and the advanced master’s degree from Institut Supérieur de l’Aéronautique et de l’Espace, Toulouse, France, in 2016.

He is currently the CEO with ScanX Company, Ltd., Tokyo, Japan.

Fumiki Hosoi received the M.E. degree in applied physics and Ph.D. degree in biological and environmental engineering from the University of Tokyo, Tokyo, Japan, in 1995 and 2008, respectively.

In 1995, he joined the Opto-Technology Laboratory, Furukawa Electric Company, Ltd., Ichihara, Japan. He is currently with the Graduate School of Agricultural and Life Sciences, The University of Tokyo.