A Convex Hull-Based Feature Descriptor for Learning Tree Species Classification from ALS Point Clouds

Yanxing Lv, Yida Zhang, Suying Dong, Long Yang, Zhiyi Zhang, Zhengrong Li, Shaojun Hu

***Abstract*—Classifying tree species from point clouds acquired by LiDAR scanning systems is important in many applications including remote sensing, virtual reality, and forestry inventory. Compared with terrestrial laser scanning systems, airborne laser scanning (ALS) systems can acquire large-scale tree point clouds from only a single scan. However, ALS point clouds have the disadvantages of low density, uneven distribution, and unclear branch structure, making the classification of tree species from ALS point clouds a challenging task. Recently, deep learning-based classification approaches such as PointNet++, which can operate directly on three-dimensional point sets, have been intensively studied in scene classification. However, the classification precision of learning-based approaches for point clouds relies on point coordinates and features such as normals. Unlike the face normals of regular objects, trees have complex branch structures and detailed leaves which are difficult to capture using ALS systems. Hence, it might be inappropriate to use the normals of ALS tree points for classification. In this paper, we propose a novel convex hull-based feature descriptor for tree species classification using the deep learning network PointNet++. To evaluate the effectiveness of our approach, three additional feature descriptors (normal descriptor, alpha shape-based descriptor, and covariance descriptor) are also investigated with PointNet++. The results show that the convex hull-based feature descriptor can achieve 86.6% overall accuracy in tree species classification, which is notably higher than the other three descriptors.**

***Index Terms*—ALS point clouds, feature descriptor, deep learning, tree species, classification**

I. Introduction

TREE species information is important for modeling trees in virtual reality and biomass calculation [1] in forestry inventory. Recently, laser scanning has been widely applied to obtain tree point clouds, and many studies have been conducted on the classification of tree species from point clouds [2]. Laser scanning systems can be roughly categorized into airborne laser scanning (ALS) [3], terrestrial laser scanning (TLS) [2] and mobile laser scanning (MLS) [4, 5] systems. For densely covered forests or jungles that are difficult for humans to reach, ALS systems have the advantage of capturing panoramic views at a distance [6]. However, an ALS system mainly captures the top of the canopy, and has limitations in capturing the branches and trunks, which will cause uneven distribution of the tree point clouds. Compared with the hundreds or even thousands of points/m^2 obtainable with MLS and TLS, the density of tree point clouds obtained by ALS is relatively low [4]. Therefore, we intend to find a more suitable classification method for low-density ALS point clouds.

Recently, tree species classification methods for point clouds have been widely studied [2, 3, 7]. These methods include multi-view representation method [2, 8], feature-based method [3], and spectral-based method [9]. Multi-view representation method projects 3D point clouds onto 2D images, which increases the amount of data but consumes more time and space [10]. For feature-based method, the selected features are important to the classification results. However, for ALS point clouds, features such as topological structure, 3D texture, diameter at breast height and other structural features are difficult to extract directly. Spectral-based method requires obtaining additional spectral information for classification, which is inappropriate for classification cases that only have point information. Therefore, we employ the deep learning network PointNet++ for tree species classification from ALS point clouds because PointNet++ can operate directly on point coordinates without additional captured features or spectral information [10].

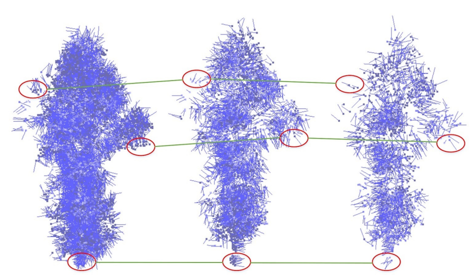


Fig. 1: Comparison of normals from gradually downsampled tree point clouds. (a) The normals of original tree point cloud with $7,000$ points; (b) The normals of downsampled point cloud of (a) with $3,500$ points; (c) The normals of downsampled point cloud of (a) with $1,750$ points.

Briechle et al. [11] first used PointNet++ to semantically label coniferous and deciduous trees on ALS point clouds. Based on $14,000$ tested tree datasets, their method achieved

The work is supported by the National Natural Science Foundation of China (61303124, 61702422), NSBR Plan of Shaanxi (2019JM-370) and the Fundamental Research Funds for the Central Universities (2452017343).

Y. Lv, Y. Zhang, S. Dong, L. Yang, Z. Zhang and S. Hu are with the College of Information Engineering, Northwest A&F University, Yangling, Shaanxi, China, S. Hu is the corresponding author, e-mail: hsj@nwsuaf.edu.cn

Z. Zhang and S. Hu are with the Key Laboratory of Agricultural Internet of Things, Ministry of Agriculture, Yangling, China.

Z. Li is with Beijing New3S Technology Co. Ltd., Beijing, China.

Manuscript received August **, 2020; revised ** **, 2020.

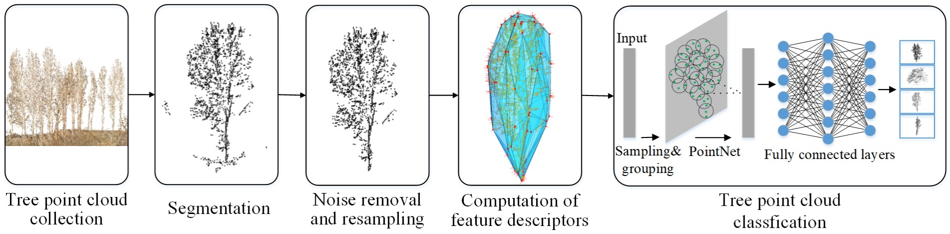


Fig. 2: Overview of tree species classification including segmentation, denoising and resampling, computation of feature descriptors, network training and tree species classification steps.

up to 90% classification accuracy for coniferous trees and 81% accuracy for deciduous trees. However, they only considered the point coordinates for broad tree species classification; employing additional features for detailed tree species classification remained a challenge. The classification accuracy of PointNet++ depends on the point coordinates and the characteristics of points such as normals [10]. Unlike regular objects, trees have intricate branching structures and tiny leaves that are difficult for ALS systems to capture [6]. Moreover, for the same tree with different sampling densities, the point normals at the same place will differ greatly as shown in the circles of Fig. 1. Therefore, it is necessary to find another feature to replace normals for detailed tree species classification. In this paper, we introduce a new feature descriptor based on convex hulls, which has the advantage of avoiding the sudden change of normal directions of tree points caused by incomplete and uneven sampling of ALS systems. The main contributions of our work are as follows:

- A novel convex hull-based descriptor that can represent the characteristics of ALS tree point clouds and improve the classification accuracy of ALS tree point clouds using PointNet++.
- We verify that PointNet++ can be used for more detailed classification of ALS tree point clouds, not just coniferous and deciduous trees.

II. Methodology

Fig. 2 shows the workflow of our tree species classification method. First, we segment the forest point cloud to individual trees. Next, we preprocess the individual tree by removing noise and resampling. Then, we compute the feature descriptors for each point to enrich the features of the tree point cloud. Finally, we input the point coordinates and features to PointNet++ to train network parameters and classify tree species.

A. Individual tree point clouds segmentation

We used the classic marker-controlled watershed algorithm to segment individual trees [12]. First, we projected the point cloud to a height map image. Then, the relative canopy height was calculated and interpolated into a raster Canopy Height Model (CHM) from the image. Next, we derived a variable window size in CHM by using a power law to describe the relation between crown radius and height [13], and the tree tops were detected by finding the local maxima within the window size. Using the computed CHM and the tree top positions, the marker-controlled watershed segmentation was used to generate 2D labeled matrices and mark different trees with different colors. Finally, we extracted the individual tree point cloud from the original dataset and the 2D labeled matrices with segmentation information as shown in Fig. 3.

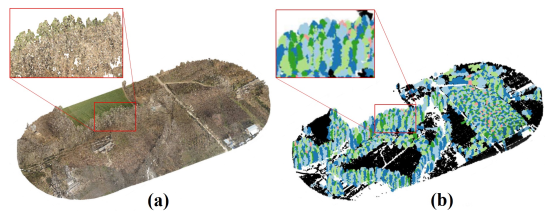


Fig. 3: The original ALS tree point clouds are segmented into individual tree point clouds by the marker-controlled watershed algorithm. (a) Original point clouds; (b) Segmented tree point clouds labeled with different colors.

B. Noise removal and resampling

Because of the complex forest environment, trees usually intersect and occlude each other; thus, the segmented individual tree point clouds are not clean. Moreover, the number of points for each tree is not identical, which is not appropriate for network training in PointNet++. Therefore, each tree point cloud was preprocessed before being applied to classification.

To remove ground noise, we used the voxel-based upward filter proposed by Guan et al. [5]. Subsequently, we used statistical filters to remove over-segmented or under-segmented points. Given a set of tree point clouds $S = \{p_1, p_2, \cdots, p_n\} \in R^3$. For each point in the set, we found its $k (k = 32)$ neighborhood points through the nearest k search, and calculated the mean distance from this point to all its neighboring points. Assuming that the resulting distribution is Gaussian, we then calculated the mean μ and the standard deviation σ of all points on this tree. In the set S, the points with a mean neighborhood distance greater than $(\mu+\lambda\sigma)$ were regarded as outliers and then removed, where λ was set as 1.0 in our work.

Because the number of input points for PointNet++ was constant, we used the following resampling method to unify the number of points for each individual tree. When a tree had more than $2,048$ points, we used a downsampling method to reduce the points to $2,048$ by introducing a 3D voxel grid. The coordinates of all the points in each voxel were approximated by using their centroid. For trees with fewer than $2,048$ points, we used an upsampling method to increase the number of points by randomly duplicating points from the point cloud to reinforce coordinates until they reach $2,048$. Finally, all the individual tree point clouds in the dataset were scaled and normalized in a unit sphere space.

C. Compute feature descriptors

The basic input of PointNet++ is only coordinates. However, classification accuracy can be improved by integrating additional features such as laser intensity, surface normals, and multispectral features [11]. However, the distribution of an ALS tree point cloud is irregular with many holes; hence, traditional feature descriptors may not be appropriate for representing the characteristics of tree point clouds. In this paper, we present four feature descriptors, including a normal descriptor, convex hull-based descriptor, alpha-shape based descriptor, and covariance descriptor as additional input for the tree point cloud classification experiment.

a) Normal descriptor: Given a tree point cloud set $S = \{p_1, p_2, \cdots, p_n\} \in R^3$, the problem of determining the normal of a point can be transformed to a least-squares plane fitting estimation problem, which can be simplified to analyze the eigenvectors and eigenvalues of the covariance matrix at that point. This covariance matrix is created from the k-nearest neighbors of the point to be estimated. Specifically, for each point $p_i = (x_i, y_i, z_i)$, the corresponding covariance matrix $\mathbf{C}_i$ is represented by:

$$\mathbf{C}_i = \frac{1}{k}\sum_{j=1}^{k}(p_j - \bar{p})(p_j - \bar{p})^{\mathrm{T}}, \tag{1}$$

where $\bar{p}$ is the centroid of the k-neighborhood of p_i. By solving the following linear equation:

$$\mathbf{C}_i \cdot \mathbf{v^i_m} = \lambda^i_m \cdot \mathbf{v^i_m}, m \in \{1, 2, 3\}, \tag{2}$$

we can compute the m-th eigenvector $\mathbf{v^i_m}$ and the eigenvalue λ^i_m of p_i. The eigenvector corresponding to the minimum eigenvalue is then approximated as the normal vector of the point.

Through the function $estimate\_global\_k\_neighbor\_scale()$ from the open source library CGAL [14], we can find an appropriate global scale of the point cloud, and then compute normals using the CGAL function $pca\_estimate\_normals()$. The estimated normals of the tree point cloud from Fig. 2 are shown in Fig. 4(a).

b) Convex hull-based descriptor: Motivated by the observation that trees in the same species have similar crown shapes and the sampling density will not overly affect the macro-shape of a tree, we introduce a 3D convex hull to describe the crown shape of a tree point cloud, implemented with the quickhull algorithm [15].

Once the convex hull is computed, the feature descriptor of each point is represented by the orientation vector of the point relative to its convex hull, which can be categorized into three cases. (i) If the point is on the vertex of a triangular patch of the convex hull, the orientation vector of the point is represented by $\mathbf{v_i} = \sum_{j=1}^{k} \mathbf{n_j}/k$, where $\mathbf{n_j}$ is the normal

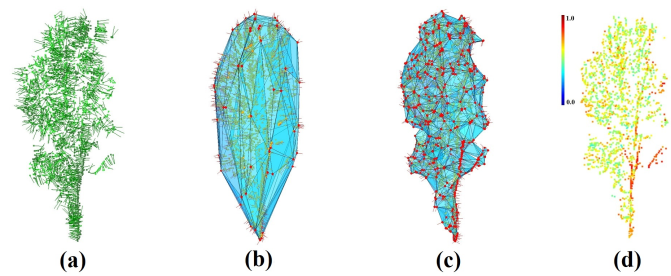


Fig. 4: Visualization results of four feature descriptors of the tree point cloud in Fig. 2. (a) Normal descriptor; (b) Convex hull-based descriptor; (c) Alpha shape-based descriptor; (d) Covariance descriptor.

of the j-th triangle adjacent to point p_i. (ii) If the point is inside a triangular patch, the orientation vector of the point is denoted by:

$$\mathbf{v_i} = \sum_{m\in\{1,2,3\}}(1 - \frac{d_m}{\sum_{j\in\{1,2,3\}} d_j})\mathbf{n_m}, \tag{3}$$

where $\mathbf{n_m}$ is the m-th orientation vector of the three vertices on the triangular patch and d_m is the distance from the current point to the m-th vertex. (iii) If the point is inside the convex hull, the ray from the centroid of the whole point cloud to the point is formed firstly, then the intersecting triangular patch of the ray is found, and the orientation vector formed at the intersection point according to (i) or (ii) is used to represent $\mathbf{v_i}$.

c) Alpha shape-based descriptor: In contrast to a convex hull descriptor, an alpha shape (α-shape) [16] descriptor is able to generate more compact hulls of input points to describe the structure of the input by controlling a parameter α. If the value of α is large, the created shape will degenerate into a convex hull. However, a small α value will generate a concave-like hull with holes. Therefore, it is crucial to choose a good α value to form a hull on the tree point cloud as tightly as possible. Moreover, the created hull should consist of a number of singular triangular facets with as few holes as possible. We found an optimal α value using the function $find\_optimal\_alpha()$ from CGAL [14]. Then we computed the triangle surfaces using the function $get\_alpha\_shape\_facets()$. Fig. 4(c) shows a connected α-shape of the tree point cloud in Fig. 2. The calculation of the orientation vector for each tree point is similar to the convex hull-based descriptor. However, if a point is inside an α-shape, the ray from the centroid of the point cloud to the point may intersect the α-shape several times. In this case, we compute the distances between the point and the intersected triangular patch, and the triangular patch with the minimum distance is set as the candidate surface and the corresponding intersection point is used to calculate the orientation vector.

d) Covariance descriptor: A covariance descriptor has the advantage of describing shape, location and color information for images and point clouds, and it is robust to changes in rotation and illumination. Therefore, we also consider the covariance descriptor in our tree species classification experiments. According to the covariance matrix of a point estimated in Eq. (1), we define the covariance descriptor $\sigma_m(p_i)$ from the eigenvalues of the covariance matrix by:

$$\sigma_m(p_i) = \frac{\lambda_m^i}{\sum_{j\in\{1,2,3\}} \lambda_j^i}, m \in \{1, 2, 3\}. \tag{4}$$

Fig. 4(d) illustrates the dominant covariance feature $\sigma_3(p_i)$ ($\sigma_3(p_i) > \sigma_2(p_i) > \sigma_1(p_i)$) represented by different colors, where the component of the covariance descriptor is linearly converted to a RGB color in a colormap. From the image, we can observe that yellow and green colors are mainly distributed on the area of branches and leaves, and red color is mainly distributed on the area of the trunk.

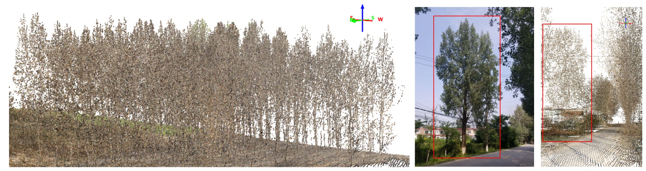
D. Deep learning network

We use the PointNet++ deep learning network as the classifier. Without rasterization, the network directly uses the coordinates of points as input to retain 3D spatial information. Moreover, it uses a density-adaptive layered network, which can combine features of different scale regions. Thus, good results can be achieved for classifying non-uniform sparse ALS point clouds. In this experiment, we set the number of object categories as 4, batch size as 8, and point number as 2048. The rest parameters such as initial learning rate, optimization algorithm, momentum value are identical to the work of Qi et al. [10]

III. Results and discussion

We used an airborne LiDAR scanning system to acquire tree point clouds. The system consisted of a SwissDrones Dragon35 unmanned aerial vehicle (UAV) and a scanning device (Riegl Vux-Sys). The point cloud used in this paper was collected from a countryside in Chang'an District, Xi'an, Shanxi Province, China. The UAV has a flying height of about $200m$ and the sampling density of point clouds was approximately 40points/m^2. Fig. 5(a) shows a grove of tree point clouds scanned by our ALS system. The main tree species in this area are: T1: Poplar, T2: Birch, T3: Camphor tree, T4: Purple Leaf Plum, as shown in Fig. 6. After deploying LiDAR scanning, we conducted a field investigation to confirm the local tree species and collect field data. The comparison between the field data and the point cloud data is shown in Fig. 5. Fig. 5(b) shows the real scene photograph captured by a camera, and Fig. 5(c) shows the corresponding point cloud scene captured by the ALS system. We subsequently conducted PointNet++ training on an Ubuntu 16.0 OS with an Nvidia Quadro K620 graphics card. The computing platform was CUDA8.0 with the cuDNN6.1 acceleration library.

After the segmentation of individual tree point clouds, we used the voxel-based upward-growing filtering [5] to remove ground points. We divided the point cloud space into $21\times21\times17$ voxel blocks, and set the height threshold of 9 and distance threshold of 0.4. Next, we manually labeled species of the tree point clouds according to the field investigation. In our experiment, we annotated $1,330$ tree point clouds including 582 poplar, 446 birch, 90 camphor and 212 purple-leaf plum trees. Among them, 890 tree samples were used for the training set and 440 tree samples were used for the test set. We then



(a) (b) (c)

Fig. 5: Comparison of photograph and point cloud data in the same place. (a) A grove of trees scanned by our ALS system; (b) Photograph taken during the field investigation; (c) Point cloud correspondence to the similar viewpoint in scene (b).

T1 T2 T3 T4

Fig. 6: Four kinds of tree species in our study. Left is the real photo of each tree species; right is the point cloud of each tree species. (T1: Poplar, T2: Birch, T3: Camphor tree, T4: Purple-leaf plum).

input the training set to PointNet++; the average training time was approximately 6 hours. We set the input number of points as $2,048$ because it was closest to the average number of points in an ALS tree point cloud in our data.

TABLE I: Classification results of four feature descriptors

Method	Input features	Accuracy
PointNet++	Coordinate	72.7%
PointNet++	Coordinate + Normal descriptor	77.3%
PointNet++	Coordinate + Convex hull-based descriptor	86.6%
PointNet++	Coordinate + Alpha shape-based descriptor	78.4%
PointNet++	Coordinate + Covariance descriptor	82.0%

In our experiment, we tested four kinds of feature descriptors with the PointNet++ network for classifying ALS tree point clouds. As shown in Table I, the overall classification accuracy when using only the point coordinates as input was 72.7%, which is clearly lower than the accuracy with additional feature descriptors that enrich the features of input tree point clouds. The accuracy using the normal descriptor was 77.3%, which was lower than the accuracies obtained with the other feature-based descriptors. For a sparse tree point cloud with no obvious branch details, the normals are disordered at the details, which may provide incorrect structure information to the neural network. Therefore, the normals of ALS tree point clouds might not be good features for tree species classification. The accuracy of the alpha shape-based descriptor was 78.4%, an improvement over the normal descriptor because the α-shape could alleviate the error of normals caused by unevenly distributed tiny branches and leaves at small scales. However, the alpha-shape of a tree point cloud is sensitive to changes in crown shapes and the α value, and quite different hulls can be generated even for the same tree species. In contrast to the normal and alpha-shape based descriptors, the classification accuracy of the covariance descriptor was 82.0%, which is a relatively good result, since this descriptor has the potential to identify trunks, branches and leaves as shown in Fig. 4(d).

The convex hull-based descriptor achieved a classification accuracy of 86.6%, which was 13.9% higher than that of input point coordinates only, and 9.3% higher than that using the normal descriptor. This is because the convex hull represents a tree point cloud's crown shape, while ignoring the cloud's density and the details of branches and leaves. Since the shapes of the same tree species are similar, the convex hulls generated from different individuals of the same tree species will also be similar. Moreover, the change in shape of a convex hull is relatively small even for the same tree with different sampling density, and the orientation vectors of points based on a convex hull are more stable than vectors generated from the normal or alpha-shape based descriptors. Therefore, our convex hull-based descriptor can be used as an effective feature descriptor for ALS tree point cloud classification.

We have also implemented Deep Belief Network (DBN) method [2] as a comparison to our method. First, We projected 2D images of the four kinds of tree species from 3D tree point clouds using the rotation angle $10°$ [2]. Then, we trained the DBN network and the overall classification accuracy of the ALS tree point cloud is 71.8%, which is 0.9% lower than the classification accuracy of coordinates and 14.8% lower than the classification accuracy of convex-hull based descriptor using PointNet++. Moreover, the training time of DBN is more than 30 hours, that is about 5 times longer than PointNet++.

IV. Conclusion

In this paper, we propose a convex hull-based feature descriptor for tree species classification, which can represent the features of low-density tree point clouds very well. In our experiment, we test four kinds of feature descriptors with the PointNet++ network. The experimental results show that the convex hull-based feature descriptor achieves the best result and that the normal descriptor is possibly poor for representing ALS tree point clouds. In addition, we verify that the PointNet++ is a promising network for classifying more specific tree species from ALS tree point cloud, not just for coniferous and deciduous trees. In future work, we intend to expand the tree dataset by considering more tree species and annotating more tree point clouds to augment the imbalanced dataset, and improve the accuracy of tree species classification. Furthermore, the marker-controlled watershed segmentation method will lead to under or over segmentation problem and more advanced segmentation algorithm should be considered.

References

[1] L. Terryn, K. Calders, M. Disney, N. Origo, Y. Malhi, G. Newnham, P. Raumonen, H. Verbeeck, and M. Akerblom, "Tree species classification using structural features derived from terrestrial laser scanning," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 168, pp. 170–181, 2020.

[2] X. Zou, M. Cheng, C. Wang, Y. Xia, and J. Li, "Tree classification in complex forest point clouds based on deep learning," *IEEE Geoscience & Remote Sensing Letters*, vol. 14, no. 12, pp. 2360–2364, 2017.

[3] J. Li, B. Hu, and T. Noland, "Classification of tree species based on structural features derived from high density LiDAR data," *Agricultural and Forest Meteorology*, vol. 171-172, pp. 104–114, 2013.

[4] E. Puttonen, A. Jaakkola, P. Litkey, and J. Hyyppä, "Tree classification with fused mobile laser scanning and hyperspectral data," *Sensors*, vol. 11, pp. 5158–5182, 2011.

[5] H. Guan, Y. Yu, Z. Ji, J. Li, and Q. Zhang, "Deep learning-based tree classification using mobile LiDAR data," *Remote Sensing Letters*, vol. 6, no. 10-12, pp. 864–873, 2015.

[6] S. Hu, Z. Li, Z. Zhang, D. He, and M. Wimmer, "Efficient tree modeling from airborne lidar point clouds," *Computers & Graphics*, vol. 67, pp. 1–13, 2017.

[7] S. Hartling, V. Sagan, P. Sidike, M. Maimaitijiang, and J. Carron, "Urban tree species classification using a WorldView-2/3 and LiDAR data fusion approach and deep learning," *Sensors*, vol. 19, no. 6, p. 1284, 2019.

[8] A. Fujimoto, C. Haga, T. Matsui, T. Machimura, K. Hayashi, S. Sugita, and H. Takagi, "An end to end process development for uav-sfm based forest monitoring: Individual tree detection, species classification and carbon dynamics simulation," *Forests*, vol. 10, no. 8, p. 680, 2019.

[9] X. Shen and L. Cao, "Tree-species classification in subtropical forests using airborne hyperspectral and LiDAR data," *Remote Sensing*, vol. 9, p. 1180, 2017.

[10] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," 2017.

[11] S. Briechle, P. Krzystek, and G. Vosselman, "Semantic labeling of ALS point clouds for tree species mapping using the deep neural network PointNet++," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-2/W13, pp. 951–955, 2019.

[12] M. Parkan, "Digital forestry toolbox," mparkan.github.io/Digital-Forestry-Toolbox, 2018.

[13] Q. Chen, D. Baldocchi, P. Gong, and M. Kelly, "Isolating individual trees in a savanna woodland using small footprint lidar data," *Photogrammetric Engineering and Remote Sensing*, vol. 72, pp. 923–932, 2006.

[14] CGALOpenSourceProject, "The computational geometry algorithms library (cgal-4.13.1)," www.cgal.org, 2019.

[15] C. B. Barber, D. P. Dobkin, and H. Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Trans. Math. Softw.*, vol. 22, no. 4, pp. 469–483, 1996.

[16] H. Edelsbrunner and E. P. Mücke, "Three dimensional alpha shapes," vol. 13, no. 1, pp. 43–72, 1994.