forests

MDPI

Article

A Tree Segmentation Algorithm for Airborne Light Detection and Ranging Data Based on Graph Theory and Clustering

Jakub Seidl [1], Michal Kačmařík [1,*] and Martin Klimánek [2]

[1] Department of Geoinformatics, Faculty of Mining and Geology, VŠB–Technical University of Ostrava, 708 00 Ostrava, Czech Republic; jakub.seidl@vsb.cz
[2] Department of Forest Management and Applied Geoinformatics, Faculty of Forestry and Wood Technology, Mendel University in Brno, Zemědělská 3, 613 00 Brno, Czech Republic; martin.klimanek@mendelu.cz
* Correspondence: michal.kacmarik@vsb.cz

Abstract: This paper presents a single tree segmentation method applied to 3D point cloud data acquired with a LiDAR scanner mounted on an unmanned aerial vehicle (UAV). The method itself is based on clustering methods and graph theory and uses only the spatial properties of points. Firstly, the point cloud is reduced to clusters with DBSCAN. Those clusters are connected to a 3D graph, and then graph partitioning and further refinements are applied to obtain the final segments. Multiple datasets were acquired for two test sites in the Czech Republic which are covered by commercial forest to evaluate the influence of laser scanning parameters and forest characteristics on segmentation results. The accuracy of segmentation was compared with manual labels collected on top of the orthophoto image and reached between 82 and 93% depending on the test site and laser scanning parameters. Additionally, an area-based approach was employed for validation using field-measured data, where the distribution of tree heights in plots was analyzed.

Keywords: segmentation; 3D point cloud; 3D graph; forest; clustering; LiDAR; UAV

check for updates

Citation: Seidl, J.; Kačmařík, M.; Klimánek, M. A Tree Segmentation Algorithm for Airborne Light Detection and Ranging Data Based on Graph Theory and Clustering. *Forests* **2024**, *15*, 1111. https://doi.org/10.3390/f15071111

Academic Editor: Eric Casella

Received: 1 May 2024
Revised: 18 June 2024
Accepted: 21 June 2024
Published: 27 June 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Light detection and ranging (LiDAR) is a device that measures the distance to objects using an emitted laser beam which is constructed using an optical system that focuses radiation into a very narrow beam. The device records the time from the emission of the beam (emitter) to the subsequent recording of the reflected radiation and its intensity (detector). As an active method, data collection by laser scanner is not dependent on sunlight and can therefore be performed at any time of day. Generally, it is possible to distinguish between three main areas of LiDAR deployment. Spaceborne LiDAR usually provides global coverage of moderate spatial resolution data that are used in oceanography [1], climate research [2], environmental monitoring [3], and the mapping of atmospheric conditions [4]. Wavelengths are selected to efficiently penetrate the atmosphere and usually include near-infrared (e.g., CALIOP—1064 nm) and green light (CALIOP, ICESat-2—532 nm). Another possibility is airborne LiDAR which provides greater detail while covering smaller areas. Typically, wavelengths from the near-infrared light between 900 and 1550 nm are used with this method. Lastly, there is terrestrial LiDAR, which offers highly accurate data (millimeter-level accuracy) and can be utilized in construction [5], geology [6], etc. Previously, very large and expensive instruments were required for LiDAR data collection, but they are now being transformed into smaller sizes and their affordability is enabling their use in new industries. This can be illustrated by the integration of laser scanners into mobile phones (iPhone 13), or their integration into unmanned aerial vehicles (UAV), with examples of sensors from Riegl, DJI, and other companies. These can be used for the very detailed mapping of relatively small areas.

Airborne LiDAR is used in many fields of work and study, mainly because of the accuracy of the data, level of acquired detail, and the ability to penetrate vegetation cover

and collect information about the vegetation itself and about the terrain underneath. This has led to heavy utilization in many areas like archeological surveys [7], urban planning [8], and forestry [9,10]. In forestry, LiDAR is used extensively for biomass estimation [11], forest carbon mapping [12], canopy structure analysis [13], and forest inventory efforts, where single trees can be detected and further parameters derived (height, species, crown, base, height, etc.). The forest inventory itself can be distinguished into either the area-based approach (ABA) or single tree detection (STD). ABA is usually used for large-scale applications with low-detail data using statistical sampling to derive carbon stock, wood volume, or biomass. STD provides more detailed data for each tree, but it is generally more time-consuming and requires more computationally and technically demanding methods.

Studies applying the STD approach frequently use raster-based approaches applied to canopy height models (CHMs). One of the commonly used methods is the watershed algorithm [14,15] which segments images into regions based on the topological concept of valleys and ridges which are identified by flooding simulations originating from local maxima. Another option is local maxima detection, which is a central aspect of various methods. Probably the most dominant method is using a sliding window, where identified local maxima are considered tree-tops [15–18]. This approach serves as the foundation for the polynomial fitting method (PFM) with a tree segmentation accuracy of 85%, [18] and individual tree crown segmentation (ITCS). An accuracy of 64% was initially reached by Dalponte [19]. Later, Wu [15] reported accuracies of 82% for ITCS and 77% for PFM.

Another option is to work directly with point cloud segmentation. Hu [20] presented a segmentation method based on regional growth and thresholding. Similarly, as in previously discussed methods, the process assumes that tree-tops are the highest points in the point cloud and are separated from each other by a certain distance. This method accuracy has been evaluated at 86% [20] and 82% [15].

It is also possible to convert the point cloud into a 3D graph. Strîmbu [21] presented graph-based segmentation on top of the multilevel raster (levels based on height above ground—HAG). Level patches are then formed from pixels with the same value and are hierarchized into an oriented weighted graph. Segmentation is then performed with the elimination of weak edges. This method showed an accuracy varying from 75% to 99% according to the forest type.

Deep learning techniques can also be utilized for point cloud segmentation. Two main approaches using convolutional neural networks can be distinguished: the first converts the point cloud into multiple rasters which are then processed [22], the second uses direct point cloud processing with various techniques like snapshot processing [23], voxelization [24], or a PointNet network [25]. Liu [26] used various deep learning techniques for single tree segmentation with an accuracy of 91% with PointNet++, 90% with Li2012, and 86% with layer-stacking segmentation.

This paper presents a new method for single tree segmentation using a 3D point cloud obtained from a UAV. The method employs clustering and indexing to minimize computational demands and transforms the 3D point cloud into a connected 3D graph upon which segmentation is applied. Some of the main advantages of this method are preserving the 3D information during processing (rather than being based on some type of raster derived from the point cloud as in the aforementioned methods) and evaluating each point within the context of other connected points in the same set. The novelty of this study lies in its usage of clustering in the 3D point cloud and its transformation into a 3D graph which is then used for segmentation based on Louvain partitioning. Initially, the forest area is segmented into relatively homogenous sections in order to refine the segmentation parameters. The proposed segmentation method focuses solely on the spatial properties of the points. The method was implemented using open-source software libraries and was successfully tested on two sites covered by commercial forest where multiple data collection campaigns with different flight parameters were performed.

2. Materials and Methods

2.1. Study Areas

Two test sites were selected in the area around the village of Mostek in eastern Bohemia, Czech Republic (Figure 1). The first area (M1) is about 7.5 ha covered by various tree species and wood assortments and contains 4 circular sample plots with 211 trees measured during fieldwork described below. In this area, four flights were performed in total with varying flight/LiDAR setting parameters in order to allow for their consequent testing. The second area (M2) covers around 5 ha, where tree species and wood assortment distribution are much more homogenous. The area contains 3 circular sample plots with 224 measured trees. Two flights were performed. The parameters for all flights from both areas are listed in Table 1. Flight planning itself was performed using the Riegl RiPARAMETER software version 2.5.0. Data acquisition was completed on 13 June 2023 under favorable meteorological conditions.

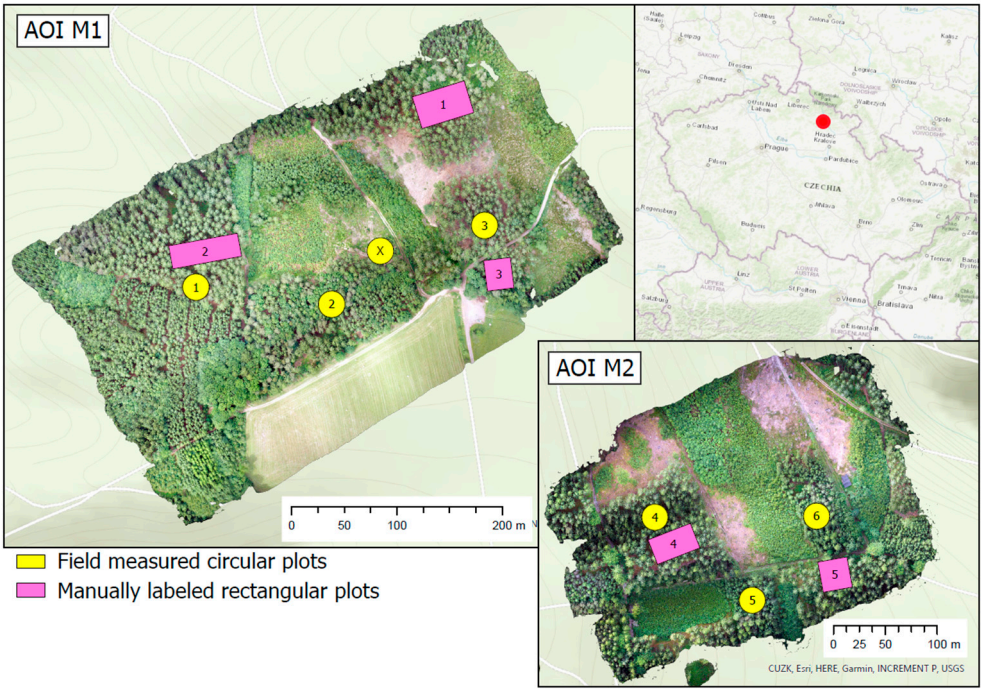


Figure 1. Visualization of areas of interest (AOIs) with reference plots.

Table 1. Flight parameters of individual realized flights in both AOIs.

Flight ID	Height (m)	Scan Angle (°)	Speed of Flight (m/s)	Scanning Line Dist. (m)	Scan Line Approx. Overlap Min/Max (%)	Scan Line Width (m)	Mean Point Density (Planned) (p/m^2)
M1_1	90	60	2.0	0.10	30/20	104	80
M1_2	90	60	2.7	0.13	30/20	104	59
M1_3	90	90	2.0	0.10	33/25	180	69
M1_4	110	60	2.0	0.10	43/37	127	65
M2_1	90	60	2.0	0.10	30/16	104	80
M2_2	90	90	2.0	0.10	33/20	180	69

For data collection, a RIEGL miniVUX-1UAV mounted on a hexacopter DJI Matrice 600 Pro was used. The sensor itself is a lightweight airborne laser scanner with a 360° field of view and multiple target capability allowing it to capture up to 5 echoes per pulse (http://www.riegl.com/products/unmanned-scanning/riegl-minivux-1uav/, accessed on 20 June 2024).

2.2. Data Preprocessing

Raw data were first processed to obtain georeferenced point clouds. Data acquired during all flights at both AOIs were processed using an identical workflow and settings. The laser scanner was equipped with its own localization unit; therefore, the first step was to calculate the exact flight trajectory using the POSPAC UAV software version 8.7. The solution was based on the global navigation satellite system (GNSS) differential technique combined with data from the inertial measurement unit. For this purpose, data were used from the GNSS permanent reference station, named CTRU, located approximately 15–17 km away from the areas of interest. The reported RMS errors of the resulting laser scanner position in all three coordinate axes were within 2 cm in the horizontal direction and around 2.5–3 cm in the vertical component. The subsequent processing of the raw scanner data was performed in the Riegl RiPROCESS software version 1.9 according to the standard workflow recommended by the manufacturer. It also included the mutual alignment of point clouds from individual flight lines and cloud coloring using photographs acquired by the RGB camera installed on the laser scanner. Before the export of point clouds into the standard LAS format, removal of noisy and isolated points was carried out. Since reference data from fieldwork were available in the Czech national coordinate system S-JTSK (EPSG:5514), point clouds from the laser scanning were transformed from the ETRF2000 coordinate system to the national one.

2.3. Validation Datasets

2.3.1. Field Measurement

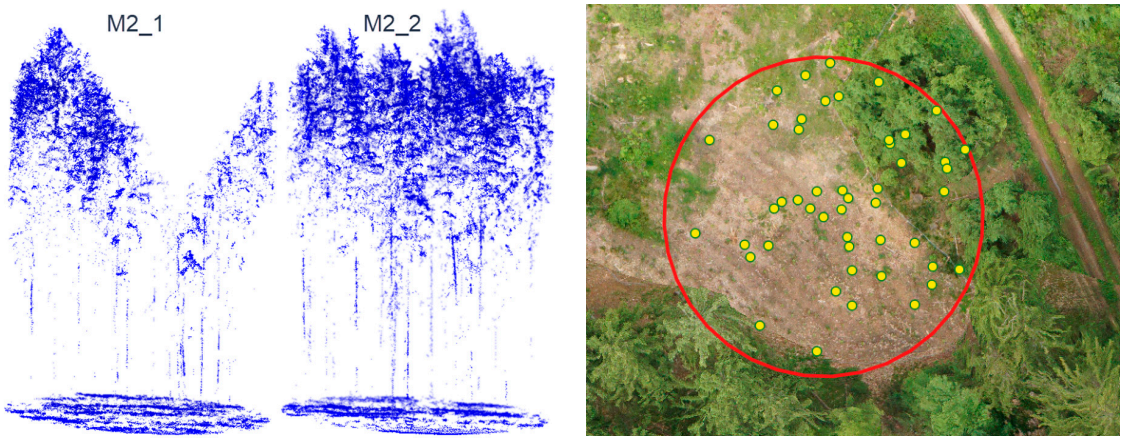
The field measurement of calibration data took place in July and August of 2022 on circular plots with a radius of 12.62 m (i.e., an area of 500 m^2). The center of each plot was marked by an RTK GNSS receiver connected to the Trimble VRSNow network (https://positioningservices.trimble.com/en/vrs, accessed on 20 June 2024); the centers of the plots were stabilized in the field. The accuracy of positioning under the tree canopy depends on the current observation parameters; however, our accuracy of position did not exceed an error of 10 cm. The test plots were laid out from the center using the Haglöf xScape ultrasonic range finder. The trees within the area were numbered in the field plot and were measured to find both their diameters at breast height (DBH) using a forestry caliper, and their heights using a laser rangefinder. The lower limit for DBH was set to 7 cm. The DBH of trees was measured by averaging the DBH for two perpendicular axes using the forestry caliper, and tree heights were measured twice from different locations using the TruPulse 360 laser rangefinder and then, averaged again. The values were recorded into the geodatabase on a field computer and were further used as field labels (FL all—all collected field labels; FL—measurements satisfying the required criteria).

Circular plot number 4 was left out of the validation phase due to an incorrect collection of LiDAR data caused by insufficient flight height above the canopy, meaning the laser could not correctly penetrate vegetation, which lead to no data coverage on top of the canopy in a particular area. Also, one of the circular plots defined in AOI M2 (visible in Figure 1 and marked by X) was not used in the validation phase because of harvesting operations performed between the times of field measurement and LiDAR data acquisition (Figure 2).

2.3.2. Orthophoto Labels

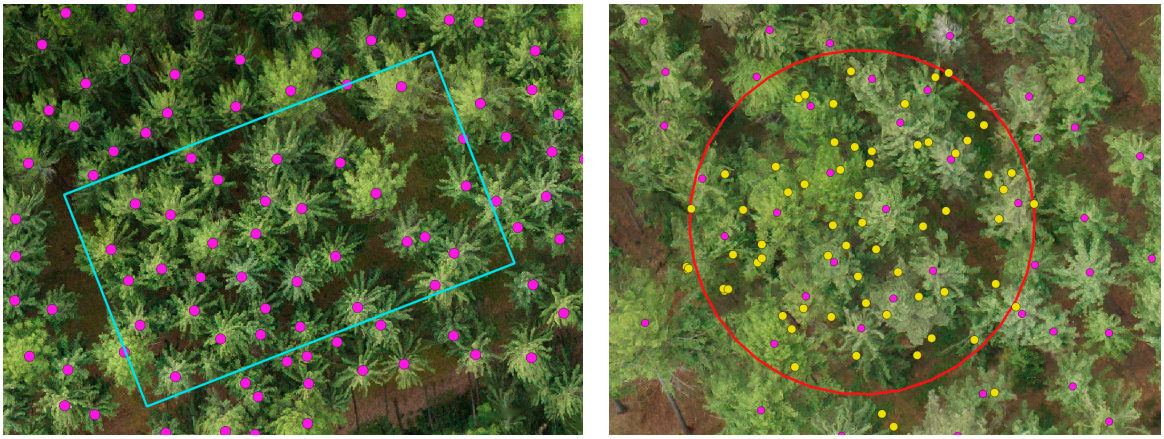
To extend the validation dataset, a manual label (ML) on top of an RGB orthophoto image was collected (Figure 3). The orthophoto was created from images collected during LiDAR scanning with a Sony Alpha 6000 and the usage of Agisoft Metashape Professional 1.8.4 software. The overall workflow required deriving a photogrammetric point cloud with the use of a structure-from-motion algorithm [27–29], which identifies key points inside images and then reconstructs 3D geometry whenever the point is visible on multiple images (three is the required minimum). Then, image stitching is employed and a true orthophoto is produced. In this case, two orthophotos were produced, one for each test area. Natural ground control points were used to georeference the orthophotos. The

collection of validation data was performed for defined rectangles randomly distributed in the forest areas, where all tree-tops were labeled with point geometry. In addition, trees nearby the defined rectangles were labeled to test the influence of modifiable area unit problem (MAUP). In total, five rectangular plots were created. Basic information about each rectangular and circular plot is provided in Table 2.



(**a**) (**b**)

Figure 2. The problem with captured laser data covering circular plot 4 (**a**). The final felled area (marked with X in Figure 1) of one circular plot covering AOI M1 with field labels portrayed by yellow points (**b**).



(**a**) (**b**)

Figure 3. Manual labels (purple dots) identified on top of the RGB orthophoto inside the rectangular plot area (turquoise). (**a**); Field inventory circular plot (red) with measured trees (yellow dots) and manual labels (**b**).

Table 2. General information about the test plots. S_plot: manually labeled plot; C_plot: circular plot from field measurements.

	AOI M1						AOI M2			
	S_plot1	**S_plot2**	**S_plot3**	**C_plot1**	**C_plot2**	**C_plot3**	**S_plot4**	**S_plot5**	**C_plot5**	**C_plot6**
Area (m^2)	1630	1510	750	490	490	490	840	1100	490	490
CHM mean (m)	24.5	26.3	21.8	26.5	21.3	22.6	28.3	25.1	20.5	25.8
ML (count)	47	40	19	14	28	17	36	24	18	48
FL all (count)	X	X	X	52	29	56	X	X	82	101
FL (count)	X	X	X	17	29	15	X	X	18	51

2.4. Method Description

The method used different approaches to segment individual trees. The main parts of the method were based on graph theory and various clustering algorithms. For segmentation, only the spatial properties (x, y, z coordinates) of the point cloud were considered (no usage of point intensity or RGB color). The overall process scheme is presented in Figure 4.

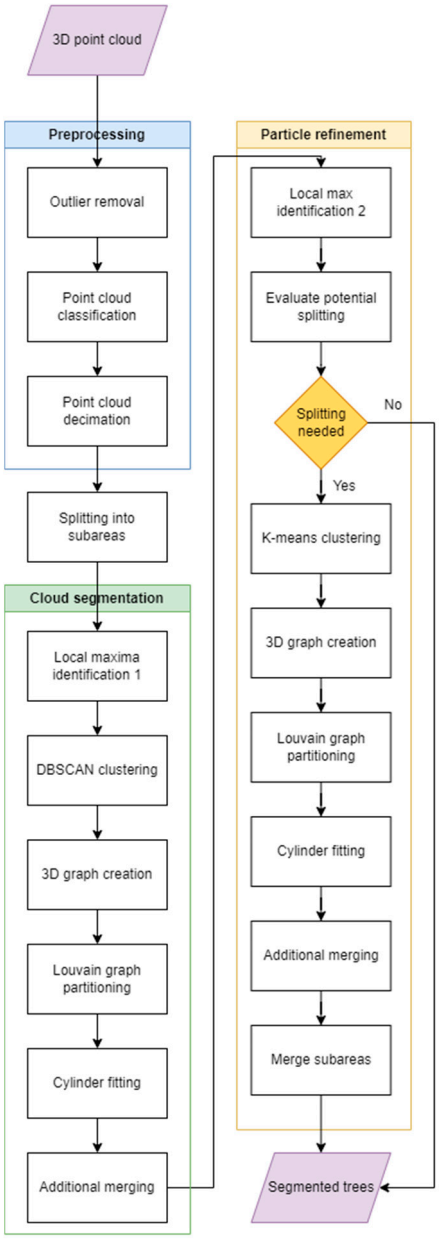


Figure 4. The whole process scheme of the method.

Preprocessing

First, the outlier points were removed using PDAL *filters.outlier*. This method was based on Rusu [30] statistical filtering (comparing points with calculated global statistics like mean and standard deviation of point clouds) or radius filtering (number of neighboring points in a given radius). In this case, statistical filtering was used with 15 neighbors and the multiplier was set to 2. Further, the classification of points representing the ground was applied. For this, a cloth simulation filter presented by Zhang [31] was used (PDAL *filters.csf* function). As the scan lines could be unevenly spaced, a point cloud decimation approach was used. Knowledge of scanning parameters is crucial at this stage to correctly

set the decimation and prevent unnecessary data loss. In this case, based on laser data acquisition settings, decimation was set to 20 cm.

Cloud segmentation

The forest can be a very complex structure in terms of mixed tree height, which could strongly affect the final tree segmentation accuracy. It could be helpful to divide the processed forest into smaller relatively homogeneous areas (in terms of tree height) to set up more precise segmentation parameters for each area and reduce computational time. To that end, an image segmentation method is applied to the CHM created with the usage of PDAL and GDAL libraries. Then, a simple flood fill algorithm (*scikit-image flood fill*) with additional refinement was triggered. The final refinement was performed according to the minimum area. If the area is too small (bellow 50 m^2), it was then connected to the neighboring area with the most similar properties. The method used 4-neighbor connectivity (shared edges) to form more compact polygons.

The next step of processing is local maxima identification. For this study, a sliding circular window with a 2 m radius was used to identify more local maxima rather than omit them. In many cases [15–19], a sliding window was used for direct tree-top detection. This process is heavily influenced by the size of the window and is therefore not optimal for all forest types. The proposed approach does not consider a local maxima as a tree-top, but as a generally significant point.

After that, a density-based spatial clustering of applications with noise (DBSCAN) [32–34] was applied via *sklearn.cluster.DBSCAN*, as it does not require a number of expected clusters and is robust to noise. Clustering operates under the presumption that a point neighborhood, defined by a certain radius, must contain a minimal number of points. The distance function influences the final shape of the neighborhood (Manhattan—rectangular shape, etc.) [32]. The main purpose of a DBSCAN is to identify point groups (clusters) in a point cloud and at the same time limit the number of points for further processing. Centroids for each group are created, and those centroids are then used as nodes. Noise points are temporarily set aside. The importance of each node is based on a number of points clustered into a point group. A check for local maxima presence is also performed and then written as a true/false attribute for each node. The nodes are then connected to the 3D graph. The creation of graph edges is performed on two levels. The first level enables connection of node Px (x_x, y_x z_x) only to the nearest local maxima P_{max} (x_{max} y_{max} z_{max}). The limiting criterion is the height and distance threshold (T).

$$z_x < z_{max}, \tag{1}$$

$$\sqrt{(x_x - x_{max})^2 + (y_x - y_{max})^2} \leq T, \tag{2}$$

These edges are considered primary edges. Secondary edges are then constructed between the node and all other points in a given distance threshold. The distance (D) is calculated as a Manhattan distance, which is less influenced by distance in the z-axis. For the trees, we anticipated that they would be taller than they were wide. Then, for each node, a number of all reachable nodes was calculated and written as an additional attribute.

$$abs(x_1 - x_2) + abs(y_1 - y_2) + (abs(z_1 - z_2) * 0.5) \leq D \tag{3}$$

A division into node communities [24] was then completed using Louvain partitioning [35–37] with the NetworkX function *community_louvain.best_partitioning*, as it is considered one of the most cited and most effective algorithms for community detection [38]. As a result, each node was assigned to a community/partition. Then, an iteration over all partitions and their direct neighbors (no transitivity neighborhood allowed) was executed and cylinder fitting was applied. The cylinder fitting was split into two levels. Firstly, the best-fitting cylinder was created around the main particle. The cylinder must be vertically oriented, and its height must be from the highest point particle to the lowest in the z-axis. Then, an intersection of the cylinder and the compared particle was evaluated.

If a sufficient number of points from the compared particle lie inside the cylinder, then the compared particle was considered part of the main particle and was merged. If the merging criteria were not met, an additional cylinder was constructed. This additional cylinder preserved the orientation of the original best-fitting cylinder, but this time the cylinder must contain all points of the main particle. Then, the intersection of the compared particle points was executed again with the cylinder. This time, at least 80% coverage was needed. Whenever the merging was performed, a neighborhood matrix was modified. A neighborhood matrix is a mathematical representation of the relationships between nodes of a graph which describes whether an edge exists between two nodes.

Next, a merging of small particles was performed. During merging, a centroid of a small particle was created and a set of closest points (Euclidean distance) of neighboring particles was found. Then, the particle with the highest number of closest points was selected and a small particle was connected to it. As DBSCAN also produces noise points, points that do not belong to any group, it was necessary to eliminate them. A suitable solution is merging noise points to the same particle as the nearest points. In this case, a Euclidean distance was used. To prevent brute-force calculation, KDTree indexation was applied. As a result of this processing, a segment of points was constructed. In many cases, the constructed segments either present a single tree or contain multiple trees, necessitating additional refinement.

Particle refinement

At this stage, the process goes through all segments. In each segment, local maxima were identified. If a segment contains more than one maxima, a line connecting both maxima was created. Then, all peak points lying under the line were selected. With a defined sampling frequency along the line, the mean z-value of selected points was calculated. When the sampling was completed, a pattern of z values between local maxima was classified. If the pattern is convex (a significant decrease between local maxima is identified) or the pattern is not able to be classified as convex or concave (Figure 5), additional splitting is applied. In case only one maxima is found, or the pattern is recognized as concave, the group is considered to be a solo tree and is written into output.

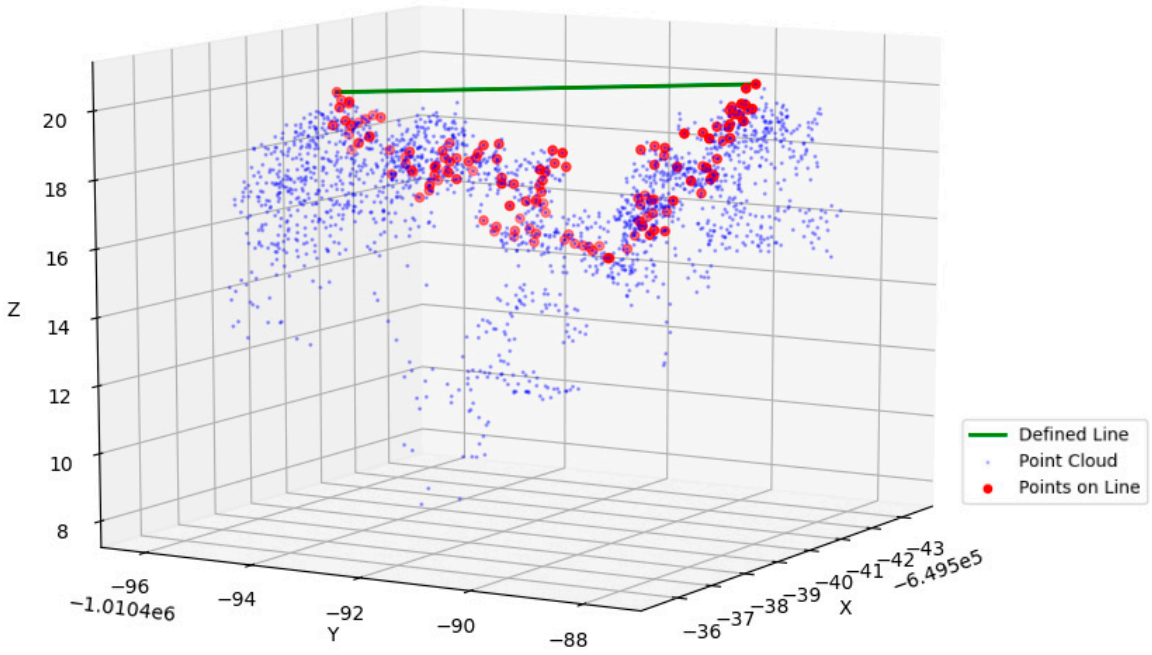


Figure 5. Visualization of convex pattern between local maxima.

Additional splitting works very similarly to the first part of the processing. The points were split into groups, only this time the K-means clustering method was used.

The expected number of groups was defined as the count of local maxima times two. Then, Louvain partitioning, cylinder fitting, and small particle merging were applied with modified parameters to prefer smaller groups. As a result, each partition should represent an individual tree. The last step consisted of merging all subareas into the final output, and calculating statistical parameters for each tree (height—max HAG in a given point partition, crown diameter—mean diameter of partition in X, Y plane). The example of segmentation results can be found in Figure 6.

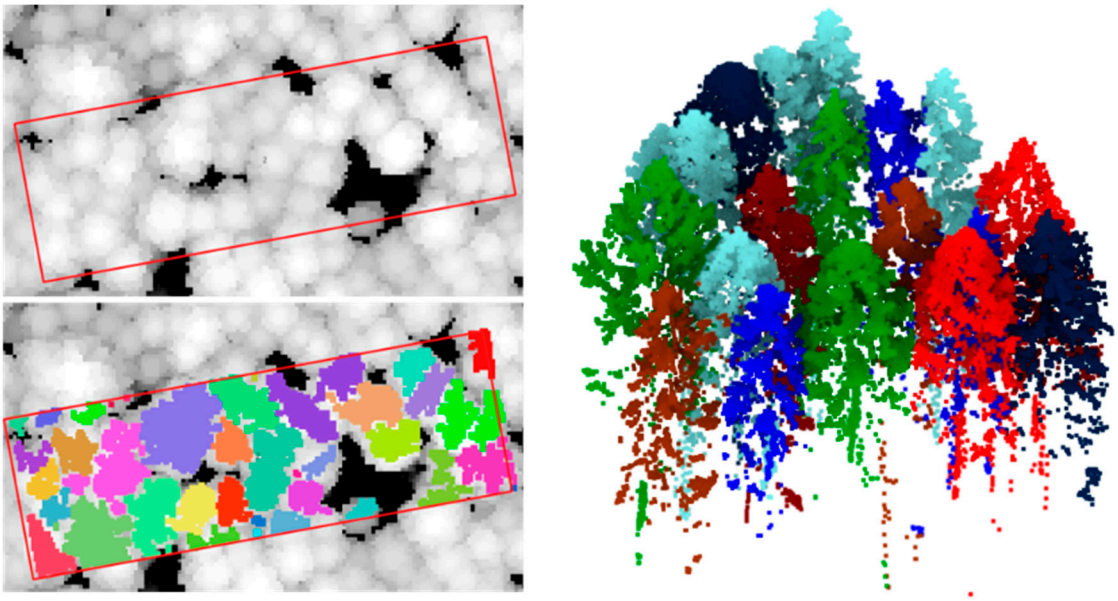


Figure 6. Example of segmentation results in raster format on top of CHM (**left**) and in 3D point cloud format (**right**) from flight M1_4.

The whole processing chain was implemented in Python 3.10.8 (https://www.python.org/, accessed on 6 March 2024) with usage of libraries:

- Numpy 1.22.3 (https://numpy.org/, accessed on 6 March 2024);
- Laspy 2.3.0 (https://laspy.readthedocs.io/, accessed on 6 March 2024);
- GDAL 3.5.1 (https://gdal.org/, accessed on 6 March 2024);
- Shapely 1.8.2 (https://shapely.readthedocs.io/, accessed on 6 March 2024);
- PDAL 3.1.2 (https://pdal.io/en/latest/, accessed on 6 March 2024);
- NetworkX 2.8.4 (https://networkx.org, accessed on 6 March 2024);
- SciPy 1.7.3 (https://scipy.org, accessed on 6 March 2024);
- scikit-learn (https://scikit-learn.org, accessed on 6 March 2024);
- scikit-learn (https://scikit-learn.org, accessed on 6 March 2024).

For image attachments, a Matplotlib 3.2.2 (https://matplotlib.org, accessed on 6 March 2024) was used.

3. Results

The method was applied to all datasets from both AOIs. For rectangular sample plots, the accuracy of the method was compared with manual labels. In circular plots, the results were compared against manual labels and field measurements. By manual visual inspection, it was found that the displacement of tree-tops can differ up to 4 m between the orthophoto and the point cloud. The reason for this is the tilt of the trees on the vertical axes in the orthophoto, the inaccuracy of the orthophoto georeferencing, and inaccuracy of tree-top locations derived from the point cloud (a detected tree-top does not necessarily represent the real one). Therefore, an automated method for validation was not applied. The following metrics were established [39,40]:

1. True Positive (TP)—number of correctly segmented trees;
2. False Positive (FP)—number segments covering multiple tree labels + number of segments without any tree label;
3. False Negative (FN)—the tree label is not covered by any segment area.

The accuracy of the proposed tree segmentation method was evaluated with these metrics: recall (R)—correctly detected trees in relation to actual trees; precision (P)—correctly detected trees in relation to entire detection result; and F1 score ($F1$)—overall accuracy of segmentation [26].

$$R = \frac{TP}{TP + FN} * 100, \tag{4}$$

$$P = \frac{FP}{TP + FP} * 100, \tag{5}$$

$$F1 = 2 * \frac{P * R}{P + R} \tag{6}$$

The comparison of the achieved method performance with manually collected labels resulted in satisfying results for almost all flights (Table 3). The best overall accuracy (F1 score) was reached in flight M1_4 (92%). The poorest overall accuracy was reached with the segmentation applied on flight M2_2 (82%), mainly because of the results for circular plot number 6. This plot was covered by dense forest with a vast number of trees which the method was not able to fully or correctly separate. From all flights performed in the AOI M1 area, M1_2 had the lowest overall accuracy (84%). This is thought to be related to it having the lowest point density of all the flights. For all flights, except M1_1, the method put out worse results for circular plots than for rectangular plots, which was likely influenced by the larger area of the rectangular plots. The source of the decrease in accuracy was not caused by omitting trees (MS value), but in almost all cases, the accuracy was decreased by a high FP. This proves that the method tends to merge multiple tree segments together rather than simply not detecting the trees.

Table 3. Accuracy evaluation of method performance for different flights and plots.

	Rectangle Plots			**Circular Plots**			**All Plots**		
	R (%)	P (%)	$F1$ (%)	R (%)	P (%)	$F1$ (%)	R (%)	P (%)	$F1$ (%)
M1_1	91.9	84.0	87.7	89.7	86.2	88.0	91.1	84.8	87.8
M1_2	89.5	81.9	85.6	91.3	76.4	83.2	90.2	79.8	84.7
M1_3	97.0	91.4	94.1	86.3	76.0	80.1	94.7	86.5	90.0
M1_4	98.9	93.2	96.0	93.8	80.6	86.7	97.2	88.5	92.6
M2_1	100	93.3	96.5	91.1	77.3	83.6	95.5	84.9	89.9
M2_2	92.9	85.4	89.1	97.7	60.5	74.8	95.0	72.1	82.1

Validation with field measurements was then performed. In this phase, direct single tree detection was not possible due to position inaccuracy between the datasets (vegetation was too dense to correctly assign points to specific tree). Therefore, a statistical area-based approach (ABA) of the whole plot was executed. The field measurement also considered younger trees in the plots which were hidden under the canopy and had a DBH equal or greater than 7 cm. These proved to be impossible to detect in the point cloud with the current laser scanning settings for such dense vegetation cover. Therefore, a subset of data from the field measurement was derived with selection according to DBH and tree height. The minimum threshold for the tree height filter was set as a value 30% lower than the calculated mean value in the circular plot measured on top of CHM (pixels with height of less than 1 m were not considered). The minimum DBH value was set to 20 cm.

To further evaluate the proposed method, a comparison with commercial software was performed. Software Lis Pro 3D in version 2024-06-06-3000 (https://laserdata.at/lis_pro_3d.html, accessed on 23 May 2024) was selected which represents a complex solution

for point cloud processing and analysis and includes a module for forestry applications. The single tree derivation tool, which is based on CHM derived from the point cloud, was used. Several variants with different settings were created to reach the best results. A single dataset from one selected flight from each test area (M1, M2) was processed with the commercial solution (Table 4). Datasets were selected based on the highest accuracy reached by the proposed method. Again, the ABA approach was used and an observed metric, overall accuracy (OA) [40], was used. In all validation plots, the proposed method reached a higher accuracy than outputs from the Lis Pro 3D. The most significant difference was observed in circular plots where almost a 40% difference in accuracy was measured.

Table 4. Accuracy evaluation of proposed method against Lis PRO 3D results.

	Rectangle Plots (OA%)		Circular Plots (OA%)		All Plots (OA%)	
	Lis Pro 3D	**Proposed**	**Lis Pro 3D**	**Proposed**	**Lis Pro 3D**	**Proposed**
M1_4	80.5	88.8	67.2	83.8	73.9	86.3
M2_1	76.0	86.4	41.7	80.5	58.9	83.5

The tree height distribution across all flights is shown in Figure 7. It indicates the same maximum value for all flights, which corresponds with using local maxima during processing. Minimum tree heights vary due to false detections. Median tree height value exhibited minimal variance among all flights. In comparison to the field measurement, the tree height calculated from the LiDAR data was generally higher. This could be partly attributed to the almost one-year gap between dataset collection during which the trees grew. Depending on the tree species, age, and conditions of the forest habitat, this height increase could be 0.5 m.

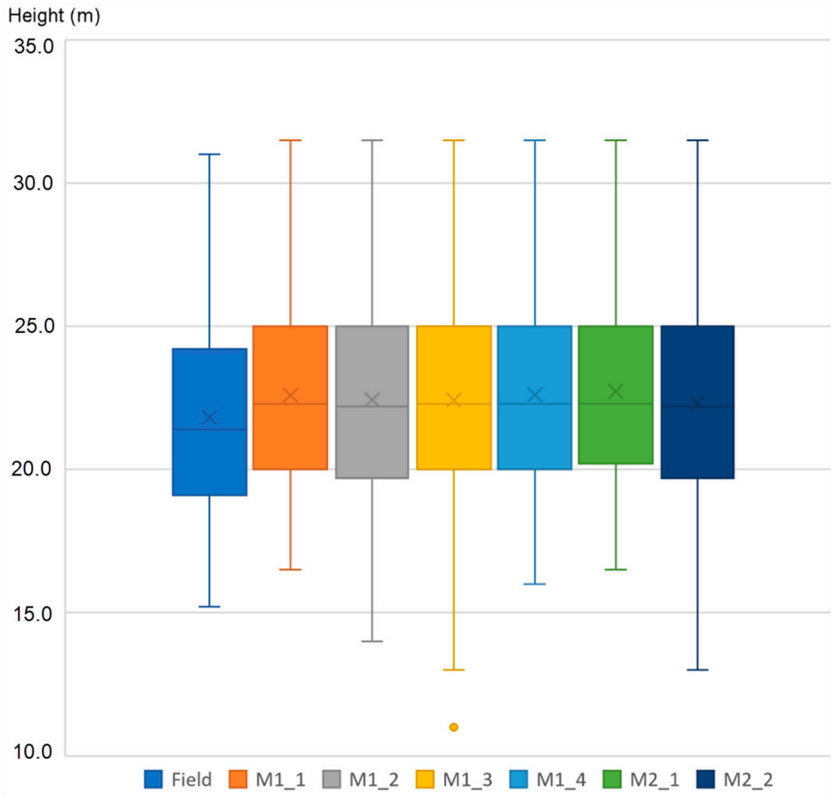


Figure 7. Graph of tree heights in circular square plots measured in the field and derived from laser scanning data.

4. Discussion

The method proved to be more efficient for mature trees and sparser forest areas. The segmentation in both areas shows the lowest accuracy for missions with low point density, but the direct connection between higher density, and thus higher accuracy, was not proven. To evaluate the results in the context of other studies, those that used a similar approach were selected. The first was the study of Strîmbu [21] which used graph-based segmentation on three forest areas in Louisiana (US). The accuracy for the forests of the same age, species, and height reached 92% and 99% (regularly spaced trees). For forest areas with a higher variability in species, height, etc., accuracy was about 75%. The second study, by Neuville [41], used machine learning with HDBSCAN clustering and principal component analysis (PCA) in a deciduous closed-canopy forest area situated in Germany. The accuracy (F1 score) reached 82% for data collected during the leaf-off season, and 50% for data collected during the leaf-on season. Dersch [39] used a watershed algorithm for stem detection and graph-cut-based clustering for single-tree segmentation. In the test area, plots for deciduous, coniferous, and mixed forests were selected. For the mixed area, an accuracy (F1 score) of 73% was achieved, for deciduous, 77% was achieved, and for coniferous, 74% was achieved.

One potential area of method improvement could be dealing with areas with dense vegetation where the accuracy of the current approach is visibly lower than in other areas. The process still uses some fixed parameters (circular window radius, minimum number of point thresholds, etc.). It could be beneficial not to set each parameter individually, but rather define a category of forest (high dense forest, high sparse forest, etc.) that is expected in each area, and each category would have a specific setting for those parameters. Another source of potential improvement could be the usage of additional information, like point intensity or RGB color, during the segmentation phase.

5. Conclusions

In this paper, we presented a novel method for single tree segmentation applied to 3D point cloud data with the usage of clustering methods and graph theory. Firstly, multiple datasets with different scanning parameters were acquired and pre-processed. Afterwards, a segmentation process was applied and the results were validated with manually collected labels and field inventory measurements. The method proved to be suitable overall for tree segmentation. The accuracy of the proposed method reached a mean value of about 88% through all validation plots and datasets. The highest accuracy, 93%, was reached for flight AOI M1 with the second sparsest dataset (65 p/m^2 and highest flight height (110 m). The lowest accuracy, 82%, was measured on datasets from AOI M2, which is covered by denser forest. From AOI M1, the lowest accuracy of segmentation was 85%. Validation with field reference was performed using the ABA approach, which also supported the overall promising results. The proposed approach outperformed a commercial solution from the LIS Pro 3D software 9.5.0. A potential for further refinement of the method still exists. The algorithm can be provided by contacting the authors.

Author Contributions: Conceptualization, J.S., M.K. (Michal Kačmařík) and M.K. (Martin Klimánek); methodology, J.S.; software, J.S.; validation, J.S.; formal analysis, J.S.; resources, J.S.; data acquistion, J.S., M.K. (Michal Kačmařík) and M.K. (Martin Klimánek); writing—original draft preparation, J.S. and M.K. (Michal Kačmařík); writing—review and editing, J.S., M.K. (Michal Kačmařík) and M.K. (Martin Klimánek); visualization, J.S.; supervision, M.K. (Michal Kačmařík) and M.K. (Martin Klimánek). All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by a grant SGS No. SP2024/057, from the Faculty of Mining and Geology, VŠB-Technical University, Ostrava. The field data acquisition was funded by the Internal Grant Agency FFWT MENDELU, specifically by project No. LDF_TP_2021002.

Data Availability Statement: The LiDAR dataset presented in this study area is available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Lu, X.; Hu, Y.; Trepte, C.; Zeng, S.; Churnside, J.H. Ocean subsurface studies with the CALIPSO spaceborne lidar. *J. Geophys. Res. Ocean.* **2014**, *119*, 4305–4317. [CrossRef]
2. Chand, D.; Anderson, T.L.; Wood, R.; Charlson, R.J.; Hu, Y.; Liu, Z.; Vaughan, M. Quantifying above-cloud aerosol using spaceborne lidar for improved understanding of cloudy-sky direct climate forcing. *J. Geophys. Res. Atmos.* **2008**, *113*, 14. [CrossRef]
3. Behrenfeld, M.J.; Hu, Y.; Hostetler, C.A.; Dall'Olmo, G.; Rodier, S.D.; Hair, J.W.; Trepte, C.R. Space-based lidar measurements of global ocean carbon stocks. *Geophys. Res. Lett.* **2013**, *40*, 4355–4360. [CrossRef]
4. Winker, D.M.; Pelon, J.R.; McCormick, M.P. CALIPSO mission: Spaceborne lidar for observation of aerosols and clouds. In *Lidar Remote Sensing for Industry and Environment Monitoring III*; SPIE: online, 2003; pp. 1–11. Available online: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/4881/1/CALIPSO--global-aerosol-and-cloud-observations-from-lidar-and/10.1117/12.462519.short#_=_ (accessed on 6 March 2024).
5. Cha, G.; Park, S.; Oh, T. A terrestrial LiDAR-based detection of shape deformation for maintenance of bridge structures. *J. Constr. Eng. Manag.* **2019**, *145*, 04019075. [CrossRef]
6. Buckley, S.J.; Howell, J.A.; Enge, H.D.; Kurz, T.H. Terrestrial laser scanning in geology: Data acquisition, processing and accuracy considerations. *J. Geol. Soc.* **2008**, *165*, 625–638. [CrossRef]
7. Fisher, C.T.; Cohen, A.S.; Fernández-Diaz, J.C.; Leisz, S.J. The application of airborne mapping LiDAR for the documentation of ancient cities and regions in tropical regions. *Quat. Int.* **2017**, *448*, 129–138. [CrossRef]
8. Yu, B.; Liu, H.; Wu, J.; Hu, Y.; Zhang, L. Automated derivation of urban building density information using airborne LiDAR data and object-based method. *Landsc. Urban Plan.* **2010**, *98*, 210–219. [CrossRef]
9. Dubayah, R.O.; Drake, J.B. Lidar remote sensing for forestry. *J. For.* **2000**, *98*, 44–46. [CrossRef]
10. Mazlan, S.M.; Wan Mohd Jaafar, W.S.; Muhmad Kamarulzaman, A.M.; Saad, S.N.M.; Mohd Ghazali, N.; Adrah, E.; Abdul Maulud, K.N.; Omar, H.; Teh, Y.A.; Dzulkifli, D.; et al. A Review on the Use of LiDAR Remote Sensing for Forest Landscape Restoration. In *Concepts and Applications of Remote Sensing in Forestry*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 49–74.
11. Gleason, C.J.; Im, J. Forest biomass estimation from airborne LiDAR data using machine learning approaches. *Remote Sens. Environ.* **2012**, *125*, 80–91. [CrossRef]
12. Mascaro, J.; Detto, M.; Asner, G.P.; Muller-Landau, H.C. Evaluating uncertainty in mapping forest carbon with airborne LiDAR. *Remote Sens. Environ.* **2011**, *115*, 3770–3774. [CrossRef]
13. Lovell, J.L.; Jupp, D.L.; Culvenor, D.S.; Coops, N.C. Using airborne and ground-based ranging lidar to measure canopy structure in Australian forests. *Can. J. Remote Sens.* **2003**, *29*, 607–622. [CrossRef]
14. Vincent, L.; Soille, P. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 583–598. [CrossRef]
15. Wu, B.; Wang, Z.; Zhang, Q.; Shen, N.; Liu, J. Evaluating and modelling splash detachment capacity based on laboratory experiments. *CATENA* **2019**, *176*, 189–196. [CrossRef]
16. Hyyppä, J.; Kelle, O.; Lehikoinen, M.; Inkinen, M. A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners. *IEEE Trans. Geosci. Remote Sens.* **2001**, *39*, 969–975. [CrossRef]
17. Popescu, S.C.; Wynne, R.H.; Nelson, R.F. Measuring individual tree crown diameter with lidar and assessing its influence on estimating forest volume and biomass. *Can. J. Remote Sens.* **2003**, *29*, 564–577. [CrossRef]
18. Cao, L.; Gao, S.; Li, P.; Yun, T.; Shen, X.; Ruan, H. Aboveground biomass estimation of individual trees in a coastal planted forest using full-waveform airborne laser scanning data. *Remote Sens.* **2016**, *8*, 729. [CrossRef]
19. Dalponte, M.; Coomes, D.A. Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data. *Methods Ecol. Evol.* **2016**, *7*, 1236–1245. [CrossRef] [PubMed]
20. Hu, T.; Sun, X.; Su, Y.; Guan, H.; Sun, Q.; Kelly, M.; Guo, Q. Development and Performance Evaluation of a Very Low-Cost UAV-Lidar System for Forestry Applications. *Remote Sens.* **2020**, *13*, 77. [CrossRef]
21. Strîmbu, V.F.; Strîmbu, B.M. A graph-based segmentation algorithm for tree crown extraction using airborne LiDAR data. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 30–43. [CrossRef]
22. Zhang, L.; Li, Z.; Li, A.; Liu, F. Large-scale urban point cloud labeling and reconstruction. *ISPRS J. Photogramm. Remote Sens.* **2018**, *138*, 86–100. [CrossRef]
23. Boulch, A.; Guerry, J.; Le Saux, B.; Audebert, N. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Comput. Graph.* **2018**, *71*, 189–198. [CrossRef]
24. Tchapmi, L.; Choy, C.; Armeni, I.; Gwak, J.; Savarese, S. Segcloud: Semantic segmentation of 3d point clouds. In Proceedings of the 2017 International Conference on 3D Vision (3DV), Qingdao, China, 10–12 October 2017; pp. 537–547.
25. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
26. Liu, Y.; You, H.; Tang, X.; You, Q.; Huang, Y.; Chen, J. Study on Individual Tree Segmentation of Different Tree Species Using Different Segmentation Algorithms Based on 3D UAV Data. *Forests* **2023**, *14*, 1327. [CrossRef]
27. Ullman, S. The interpretation of structure from motion. *Proc. R. Soc. Lond. Ser. B Biol. Sci.* **1997**, *203*, 405–426. [CrossRef]
28. Snavely, N.; Seitz, S.M.; Szeliski, R. Skeletal graphs for efficient structure from motion. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008. [CrossRef]

29. Brostow, G.J.; Shotton, J.; Fauqueur, J.; Cipolla, R. Segmentation and Recognition Using Structure from Motion Point Clouds. In Proceedings of the 10th European Conference on Computer Vision: Part I, Marseille, France, 12–18 October 2008; pp. 44–57. [CrossRef]
30. Rusu, R.B.; Marton, Z.C.; Blodow, N.; Beetz, M. Learning informative point classes for the acquisition of object model maps. In Proceedings of the 2008 10th International Conference on Control, Automation, Robotics and Vision, Hanoi, Vietnam, 17–20 December 2008; pp. 643–650.
31. Zhang, W.; Qi, J.; Wan, P.; Wang, H.; Xie, D.; Wang, X.; Yan, G. An easy-to-use airborne LiDAR data filtering method based on cloth simulation. *Remote Sens.* **2016**, *8*, 501. [CrossRef]
32. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the KDD96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, USA, 2–4 August 1996; Volume 34, pp. 226–231.
33. Kriegel, H.-P.; Kröger, P.; Sander, J.; Zimek, A. Density-based clustering. *WIREs Data Min. Knowl. Discov.* **2011**, *1*, 231–240. [CrossRef]
34. Schubert, E.; Sander, J.; Ester, M.; Kriegel, H.P.; Xu, X. DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN. *ACM Trans. Database Syst.* **2017**, *42*, 1–21. [CrossRef]
35. Newman, M.E.J.; Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **2004**, *69*, 026113. [CrossRef]
36. Blondel, V.D.; Guillaume, J.-L.; Lambiotte, R.; Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, *10*, P10008. [CrossRef]
37. Lancichinetti, A.; Fortunato, S. Community detection algorithms: A comparative analysis. *Phys. Rev. E* **2009**, *80*, 056117. [CrossRef]
38. Traag, V.A.; Waltman, L.; Van Eck, N.J. From Louvain to Leiden: Guaranteeing well-connected communities. *Sci. Rep.* **2019**, *9*, 5233. [CrossRef]
39. Dersch, S.; Heurich, M.; Krueger, N.; Krzystek, P. Combining graph-cut clustering with object-based stem detection for tree segmentation in highly dense airborne lidar point clouds. *ISPRS J. Photogramm. Remote Sens.* **2021**, *172*, 207–222. [CrossRef]
40. Yu, J.; Lei, L.; Li, Z. Individual Tree Segmentation Based on Seed Points Detected by an Adaptive Crown Shaped Algorithm Using UAV-LiDAR Data. *Remote Sens.* **2024**, *16*, 825. [CrossRef]
41. Neuville, R.; Bates, J.S.; Jonard, F. Estimating forest structure from UAV-mounted LiDAR point cloud using machine learning. *Remote Sens.* **2021**, *13*, 352. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.