Evaluating Factors Impacting Fallen Tree Detection from Airborne Laser Scanning Point Clouds

Heinaro, Einari

Multidisciplinary Digital Publishing Institute
2023-01-08

Heinaro, E.; Tanhuanpää, T.; Vastaranta, M.; Yrttimaa, T.; Kukko, A.; Hakala, T.; Mattsson, T.; Holopainen, M. Evaluating Factors Impacting Fallen Tree Detection from Airborne Laser Scanning Point Clouds. Remote Sens. 2023, 15, 382.

http://hdl.handle.net/10138/353317

remote sensing

MDPI

Article

Evaluating Factors Impacting Fallen Tree Detection from Airborne Laser Scanning Point Clouds

Einari Heinaro [1,*], Topi Tanhuanpää [1,2], Mikko Vastaranta [3], Tuomas Yrttimaa [1,3], Antero Kukko [4], Teemu Hakala [4], Teppo Mattsson [1] and Markus Holopainen [1]

[1] Department of Forest Sciences, University of Helsinki, 00014 Helsinki, Finland
[2] Department of Geographical and Historical Studies, University of Eastern Finland, 80101 Joensuu, Finland
[3] School of Forest Sciences, University of Eastern Finland, 80101 Joensuu, Finland
[4] Department of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute FGI, The National Land Survey of Finland, 02150 Espoo, Finland
* Correspondence: einari.heinaro@helsinki.fi

Abstract: Fallen tree mapping provides valuable information regarding the ecological value of boreal forests. Airborne laser scanning (ALS) enables mapping fallen trees on a large scale. We compared the performance of line-detection-based individual fallen tree detection when using moderate point density ALS data (15 points/m^2) and high-point-density unmanned aerial vehicle-based laser scanning (ULS) data (285 points/m^2). Furthermore, we inspected the dataset and detection methodology-related factors impacting performance in each case. The results of this study showed that increasing the point density of the laser scanning dataset enables the detection of a larger proportion of fallen trees. However, based on our experiment, a line-detection-based fallen tree detection approach is sensitive to noise, thus generating a large number of false detections, especially with high-point-density data. Different types of filters, such as a simple height-based filter and machine-learning-based filters, can be used for reducing noise. However, using such filters is always a compromise, as in addition to reducing noise and thus false detections, they also reduce the number of true detections. Hence, a less noise-sensitive fallen tree detection method utilizing the finer details visible in high-density point clouds could be more suitable for high-point-density laser scanning data.

Keywords: airborne laser scanning; unmanned aerial vehicle; light detection and ranging; deadwood; fallen trees; biodiversity

Citation: Heinaro, E.; Tanhuanpää, T.; Vastaranta, M.; Yrttimaa, T.; Kukko, A.; Hakala, T.; Mattsson, T.; Holopainen, M. Evaluating Factors Impacting Fallen Tree Detection from Airborne Laser Scanning Point Clouds. *Remote Sens.* **2023**, *15*, 382. https://doi.org/10.3390/rs15020382

Academic Editor: Lin Cao

Received: 15 December 2022
Revised: 4 January 2023
Accepted: 6 January 2023
Published: 8 January 2023

1. Introduction

Deadwood provides a habitat for a wide variety of species and is thus an essential structural and functional element supporting biodiversity in boreal forests [1]. The amount of deadwood is among the most important indicators used for monitoring biodiversity in boreal forests together with the number of large living trees and the share of old growth forests and deciduous forests [2]. In Finnish forests, which are mainly boreal, fallen trees constitute approximately two-thirds of the deadwood [3]. As relatively large and distinguishable objects, standing and fallen dead trees serve as a measurable ecological indicator widely used in forest inventories. The ecological value of dead trees increases with tree size, as the number of species dependent on a decaying tree increases as the tree size increases [4,5]. Thus, from a biodiversity perspective, large trees are of the most interest.

In forest inventories, the amount of deadwood is typically measured only from ground plots covering a rather small area in geographical space. Then, the estimate over the whole area of interest is derived based on the ground sample [6–8]. If deadwood volume and quality maps are needed, those are typically created by generalizing the ground sample over the area of interest using remote sensing [9]. The former approach is widely used for

monitoring the amount and quality of deadwood in forest management areas, regions, and countries. However, the approach is dependent on the amount and quality of the ground sample and has rather limited capabilities for capturing the spatial arrangement and local deadwood hotspots. The weakness of the latter approach has been that the predicted amount of deadwood on each map location is dependent on the remote sensing features describing only the characteristics of the standing trees. Thus, the core assumption in this approach is that certain standing forest structures always include a certain number of dead trees. To be valid, this inference requires similar silvicultural practices over the whole area of interest, which is often an unrealistic assumption. Thus, to map and monitor deadwood, and especially to characterize its spatial arrangement and hotspots, a universal, spatially detailed, and automated method for mapping fallen trees from remote sensing data would provide an efficient way for monitoring this key biodiversity indicator on a large scale.

Airborne laser scanning (ALS) is an active remote sensing method that can be used for characterizing forest structures. ALS enables measuring canopy-related characteristics, such as canopy height, height variation, and density, but also properties of the subcanopy vegetation and topography [10,11]. Conventionally, ALS has been used for generalizing ground sample measurements for wider areas. This has proven to work well for forest characteristics that are correlated with canopy height, height variation, and density [12]. However, due to the limitations of the sampling-based approach, the most useful approach for mapping deadwood would be to detect and measure it directly from ALS data. This direct approach has become increasingly feasible with the advances in laser scanning technology.

Laser scanning is based on transmitting laser pulses to a target and determining the distance to the target from the time difference (pulse-based scanners) or phase difference (continuous wave scanners) between the transmitted and returning pulse [13]. The returning pulses or partial pulses are called echoes. When monitoring forests, the upper canopy layer generates the first echoes. Some pulses continue deeper into the canopy and ultimately reflect from the terrain [14,15]. Thus, these latter echoes describe the topography and near-ground objects. These echoes are of most interest in direct fallen tree detection. When a sufficient number of laser pulses have reflected from fallen trees, they stand out as line- or cylinder-like objects in the laser scanning point cloud. Presumably, increasing the point densities of laser scanning datasets increases the probability of detecting such objects.

Several earlier studies have addressed direct fallen tree mapping using either ALS data alone or a combination of ALS data and aerial imagery. Blanchard et al. [16] segmented individual fallen trees from a combination of raster layers generated from ALS data. Similarly, Mücke, et al. [17] segmented fallen trees from an ALS-based rasterized height model, and Nyström, et al. [18] performed line-template matching on a rasterized surface created from single and last ALS returns. In contrast to the aforementioned studies, which all operated on rasterized ALS data, Lindberg, et al. [19] and Polewski, et al. [20,21] operated directly on the point cloud. Lindberg, et al. [19] used a template matching method similar to Nyström, et al. [18] at the point cloud level, whereas Polewski, et al. [20,21] used 3D shape contexts [22] and machine learning to select representative fallen tree segments from a set of fallen tree candidates generated from ALS point pairs. Most recently, Heinaro, et al. [23] detected individual fallen trees using an iterative line-fitting method based on the Hough transform.

Unmanned-aerial-vehicle-based laser scanning (ULS) is a type of ALS where the scanner is mounted on a drone. It is still a rather new concept that has not yet seen many applications in forestry. Among the first to utilize ULS in forestry applications were Jaakkola, et al. [24], who tested its suitability for individual tree mapping. Later studies [25–28] applied ULS for a living tree inventory. However, thus far, studies addressing the feasibility of ULS for fallen tree mapping are non-existent. ULS opens new possibilities for fallen tree mapping. Firstly, the point densities of ULS datasets exceed those of ALS datasets. This potentially allows the detection of fallen trees with higher accuracy, as well as estimating their dimensions more precisely. Secondly, ULS is a more agile and

cost-efficient approach for small-scale mapping compared to ALS. ULS datasets can be collected at rather short notice, and it is easy to focus dataset collection on a specific area of interest. These advantages are useful, for example, when identifying biodiversity hotspots, planning forest conservation and restoration practices, or when multi-temporal data are required. Due to its agility, ULS-based fallen tree mapping is also suitable for mapping wind damage.

The aim of this study was to evaluate the general suitability of a line-detection-based approach for direct fallen tree mapping. Furthermore, the aim was to gain insight into what types of dataset-specific properties and filtering-related decisions affect the accuracy of fallen tree detection. This insight would hopefully provide information on the factors to consider when developing new, more sophisticated fallen tree detection methods. The study aimed to answer the following research questions:

- How do differences between laser scanning datasets affect the accuracy of fallen tree detection?
- How does algorithm parameter selection affect the detection of different types of fallen trees?
- What is the impact of machine-learning-based filters applied at different stages of the detection process?

To answer these questions, we performed a sensitivity analysis on the parameters of the line-detection-based method introduced by Heinaro, et al. [23] using both an ALS dataset (point density of approximately 15 points/m^2) and a ULS dataset (285 points/m^2). As the number of parameters of this method is rather large, we selected the parameters that are common to a variety of automatic shape detection methods, which were most relevant to the research questions, and which, based on observations made during method development and testing, had the largest impact on the performance of the method. The parameters and reasoning behind why each parameter was chosen are presented in the methods section. The impact of the selected parameters was inspected in general as well as for different types of trees, as tree type-specific sensitivity was of interest due to the varying ecological value of different types of trees.

2. Materials and Methods

2.1. Study Site

The study site (Figure 1) covered approximately 16 km2 and was located in the Region of Kainuu, Finland. The western part of the site was within Hiidenportti National Park, whereas the eastern part was a state-owned managed forest. The state-owned managed forest had ecological relevance, as it connected two conservation areas: Hiidenportti National Park and Teeri-Lososuo mire conservation area. Accurate information on deadwood is of high importance in such areas, as it allows planning of ecological corridors between the conservation areas.

The part of the site within Hiidenportti National Park mainly consisted of old-growth forests while the forest maturity in the eastern part was more varied. The forest of the study site as a whole can be described as a boreal forest mainly consisting of Norway spruces (*Picea abies* L. Karst), Scots pines (*Pinus sylvestris* L.), silver birches (Betula pendula, Roth), and downy birches (Betula pubescens, Ehrh). Some aspens (*Populus tremula*, L.) could also be found around the area. The topography varied from flat to steep with elevations ranging between 190 and 250 m above sea level.

2.2. Datasets

2.2.1. Airborne Laser Scanning Data

The airborne laser scanning dataset used in this study covered a 16 km^2 area (Figure 1) around the study site. The dataset was collected in May 2019 with a Riegl VQ1560i laser scanner (RIEGL Laser Measurement Systems GmbH, Horn, Austria). The scanner has two channels, both of which collect data using straight parallel scan lines that are tilted 28 degrees against each other. The resulting point cloud is rather uniform with occasional

linear patterns caused by straight scan lines. The timing of the data acquisition was specifically selected so that the snow in the study site had already melted but the trees had not yet come into leaf. The dataset was collected with five parallel flight lines with 30% overlap and a perpendicular flight line that tied all lines together. The point density of the ALS dataset was approximately 15 points/m^2, although the point density varied at different parts of the study site depending on flight line coverage.

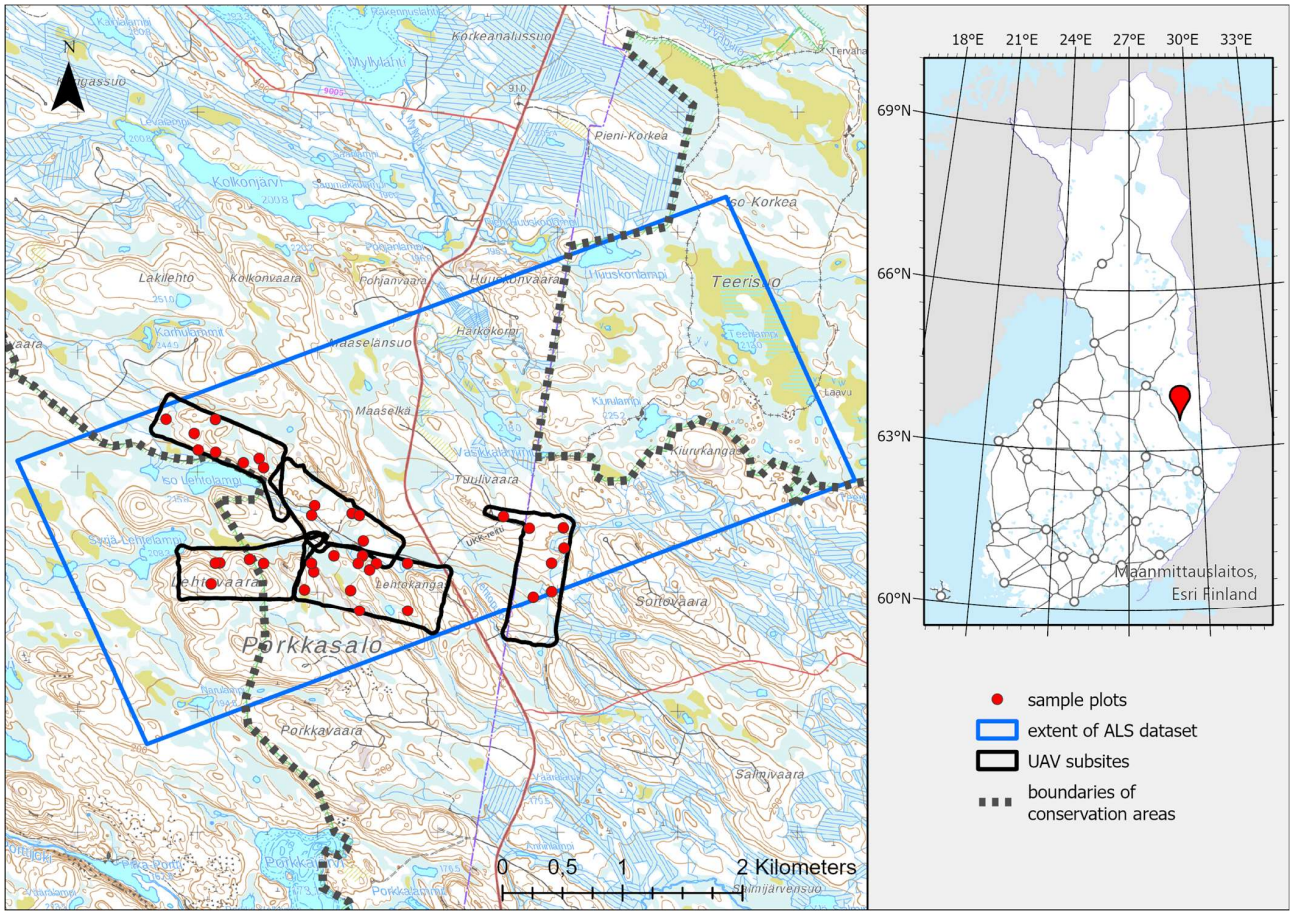

Figure 1. Map of the study site. The blue rectangle shows the boundaries of the airborne laser scanning (ALS) data, whereas the black polygons show the boundaries of the laser scanning data collected using an unmanned aerial vehicle (ULS). The red circles are sample plot locations. The western black and green dotted line depicts the east boundary of Hiidenportti National Park, whereas the eastern black and green dotted line is the west boundary of Teeri-Lososuo mire conservation area. The study area between these two natural parks is state-owned managed forest. Background maps: National Land Survey of Finland.

2.2.2. ULS Data

The ULS dataset was collected at the beginning of June 2020 in similar snow-free and leaf-off conditions as the ALS dataset. It was collected using a Riegl miniVUX-1DL laser scanner (RIEGL Laser Measurement Systems GmbH, Horn, Austria) coupled with a NovAtel CPT7 dual-antenna GNSS-IMU device. The scanner, specifically designed for UAV use, collects data using a virtually circular scan pattern on the ground. In total, the dataset covered an area of approximately 2.4 km^2, which consisted of five distinct but partially overlapping subsites (Figure 1). Each subsite was covered with six to nine parallel flight lines, depending on the shape and size of the subsite. The flight line overlap was 30%. The point density was approximately 285 points/m^2, although the point density varied depending on flight line coverage.

2.2.3. Reference Data

The reference dataset contained information on 197 fallen trees measured at 37 circular sample plots (radius 9 m). The inventory on 14 of these sample plots was taken between July and September 2019. These sample plots were part of a systematic sample plot grid that was aimed to capture the full variety of forest characteristics within the area covered by the ALS data (For more details, see Heinaro, et al. [23]). The remaining 23 sample plots were placed within the subsites covered by the ULS dataset and scrutinized in November 2020. These sample plots were placed at locations with a known abundance of fallen trees to ensure that the dataset would contain objects of interest. Table 1 presents a summary of the trees measured for reference.

Table 1. Summary statistics of the reference trees, including the minimum (min), mean, maximum (max), and standard deviation for length and diameter, and the proportion of trees in each decay class.

	Total	**Scots Pine**	**Norway Spruce**	**Silver and Downy Birch**	**Aspen**
Number of trees	197	22	148	20	7
Length (m)					
Min	1.6	2.1	1.6	3.1	5.9
Mean	11.9	9.3	12.4	10.2	13.1
Max	28.8	22.2	28.8	22.1	18.0
Standard deviation	5.1	4.7	5.0	5.4	4.6
Diameter (mm)					
Min	100.0	100.0	100.0	110.0	233.0
Mean	193.0	159.5	188.2	203.0	372.1
Max	450.0	296.0	404.0	418.0	450.0
Standard deviation	72.9	49.9	61.5	80.5	82.5
Decay class					
Proportion of class 1	52%	77%	47%	60%	57%
Proportion of class 2	16%	14%	18%	10%	0%
Proportion of class 3	13%	9%	12%	15%	29%
Proportion of class 4	14%	0%	17%	15%	0%
Proportion of class 5	6%	0%	7%	0%	14%

All fallen trees located partially or fully within the sample plots were considered part of the reference dataset. The information on each such tree included the location measured at the top and bottom ends of the tree and the diameter, state of decay, and tree species. The exact locations of the top and bottom ends of the fallen trees were measured using a Trimble R2 (Trimble Inc., Sunnyvale, CA, USA) real-time kinematic global navigation satellite system (GNSS) in FIX mode. The diameter of each tree was measured at a 1.3-m distance from the bottom end (i.e., diameter at breast height, DBH) using steel calipers. The bottom-end diameter was measured if the breast height could not be determined. The state of decay was determined based on the guidelines applied in the Finnish national forest inventory [29] and measured on a scale of one to five with one representing the least decayed trees and five representing the most decayed trees. Trees in decay class 1 were hardwood where the decay process had not yet begun, whereas the trees in decay class 5 were soft by the rot throughout the diameter. Please refer to Appendix A in Heinaro, et al. [23] for a more specific description of the different decay classes. In addition to the fallen tree characteristics, the dataset contained information on the forest characteristics at each sample plot, including the species-independent and species-specific mean DBH, mean height, basal area, and total volume of living trees, as well as the number of trees per hectare and the amount and size of undergrowth. These characteristics were obtained by recording the DBH and species of each living tree within the sample plot and measuring the height of a comprehensive sample that aimed to capture the variety of living trees within

the sample plot. A detailed description of the methodology used in collecting the reference dataset is presented in Heinaro, et al. [23].

2.3. Methods

2.3.1. Pre-Processing the Laser Scanning Data

Both laser-scanning datasets were first normalized. In other words, topography was removed from the point clouds to access objects of interest near the ground surface. The normalization process started by extracting the ground points from the point clouds. Ground points were extracted using the method of Zhang, et al. [30], which was available in R's lidR package [31,32]. Note that the lidR implementation of this algorithm applies morphological operations directly on the point cloud as opposed to the original implementation that instead used a raster generated from the point cloud. After ground point extraction, a digital terrain model (DTM) was generated from the ground points and the corresponding DTM height was subtracted from each point in the point cloud.

2.3.2. Fallen Tree Detection

Fallen trees were detected using the method introduced by Heinaro, et al. [23]. The steps of this method are shown in Figure 2. Below, we present a summary of the method. Refer to the original article for a more comprehensive description of the method.

The method can be divided into three main parts:

1. Filtering.
2. Line detection.
3. Segment delineation and classification.

The filtering step consisted of two parts: Height range (HR)-based filtering and connected component classification (CCC). HR-based filtering restricted the analysis to point cloud points located close to the ground (detected in an earlier step), as most fallen trees lie close to the forest floor. CCC grouped neighboring points and removed point groups with roundish non-elongated shapes that are not typical for fallen trees. The grouping and classification were performed on a binary grid (cell size 0.2 m) in which values of 1 represented grid cells with points inside them and values of 0 represented empty cells. The 8-neighborhood was used for connecting non-empty grid cells together. A shallow neural network was used for classifying point groups as either belonging or not belonging to fallen trees. The neural network was trained on manually labeled point groups that were extracted from outside the sample plots. The purpose of CCC was to detect and then remove points originating from other near-ground objects such as rocks and stumps.

Table 2. A summary of the tested parameters: Height range (HR), connected component classifier (CCC), minimum number of points (MNP), and false tree detection (FTR) of the fallen tree detection method. Refer to Heinaro, et al. [23] for a detailed description of the method.

Parameter	Description	Range
HR	The height range to which fallen tree detection is applied. Fallen trees are detected from point cloud points falling within the given height range.	Lower height range limit: 0.1, 0.2 and 0.3 m Upper height range limit: 1 m 3 different height ranges in total.
CCC	A binary parameter that determines whether the connected component classifier is used. The connected component classifier is a shallow neural network that classifies point groups as either belonging or not belonging to fallen trees. Essentially, CCC removes roundish non-elongated point groups from the point cloud before line detection, as these point groups are not likely to belong to fallen trees.	0—the classifier is not used. 1—the classifier is used.
MNP	The number of point cloud points that must fall on the same line for a line segment to be detected.	3, 6, 9, . . . , 30 points. 10 different values in total.
FTR	A binary parameter that determines whether the false tree remover is used. The false tree remover is a convolutional neural network that inspects each detected fallen tree segment and removes the segments that do not resemble fallen trees.	0—the classifier is not used. 1—the classifier is used.

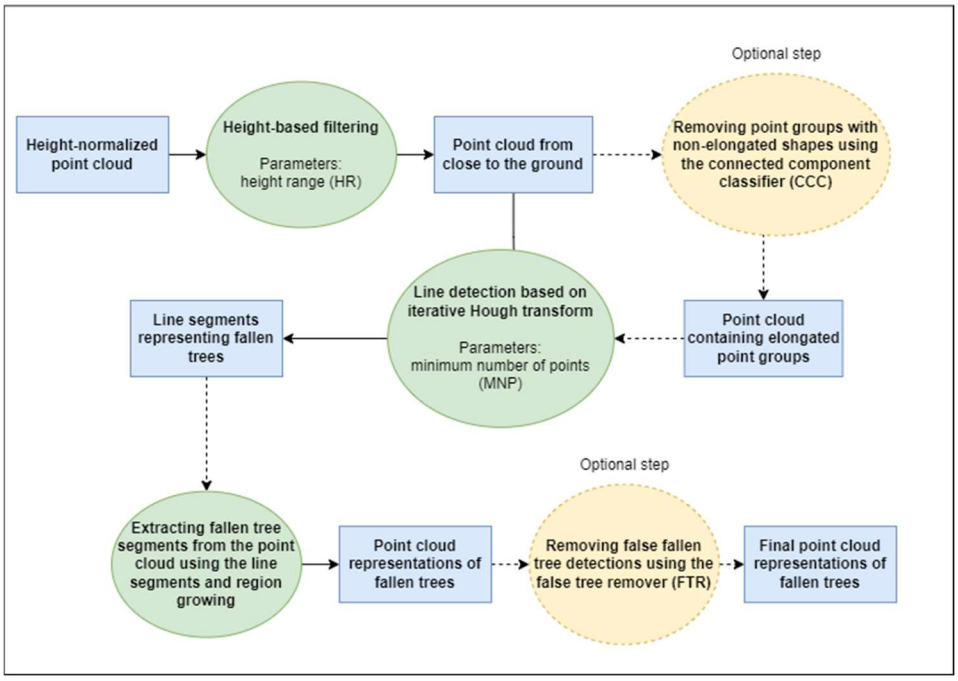

Figure 2. The steps of fallen tree detection. Refer to Table 2 for a detailed description of the parameters mentioned.

The line detection step was the core of the detection method and was based on the Hough transformation [33,34]. The points not removed in the filtering step were first projected on the xy-plane. Then, linear shapes were detected in the projected point cloud using iterative Hough transformation. Iterative Hough transformation detected a line segment in the point cloud and removed the points within 0.5 m from the line segment to ensure that the same line segment would not be detected again. The most notable parameter of this step was the minimum number of points (MNP) that must fall on the same line for a line segment to be detected.

The detected line segments represented fallen tree candidates. The segment delineation and classification step generated point cloud representations of these fallen tree candidates using a region-growing algorithm. The algorithm started from the longest line segment and assigned the points within 0.5 m from the segment to this segment. Then, points within 0.2 m from the assigned points were added to the segment iteratively until no more neighboring points were found. The same iterative process was repeated for all of the line segments. Then all of the generated point cloud representations of the detected fallen tree candidates were classified as fallen trees or false detections using a trained convolutional neural network denoted as the false tree remover (FTR) further in the text.

We ran the fallen tree detection for both the ALS and ULS datasets using 120 different parameter combinations to reveal the impact of each varied parameter (Table 2). The varied parameters were:

1. The HR used in height-based filtering. We kept the upper limit fixed at 1 m but varied the lower limit between 0.1, 0.2, and 0.3 m. Errors in ground extraction resulted in some ground points being classified as above-ground points. Furthermore, ground vegetation was often dense close to the forest floor. For these reasons, setting the lower height limit to exactly zero results in a dense point cloud and a large number of false fallen tree detections. Setting the lower height limit to a value slightly larger than zero mitigates this issue but prevents the detection of fallen trees located very close to or laying entirely on the forest floor.
2. Whether to use CCC or not for filtering the point cloud. The CCC step was originally created to remove point groups originating from objects other than fallen trees and in-

accuracies in ground classification, as these point groups resulted in many false fallen tree detections. However, using CCC also removes some point groups originating from fallen trees, resulting in a smaller number of true fallen tree detections.

3. The MNP that must fall on the same line for a line segment to be detected. The value of MNP varied between 3 and 30 points with 3-point increments. Smaller values of MNP should make the line detection process more sensitive, resulting in more true detections, but also more false detections, and vice versa. The optimal value of this parameter should depend on the point density, and thus the optimal value should be different for the ALS and ULS datasets.
4. Whether to use the FTR or not. The CCC step cannot remove all point groups not originating from fallen trees and thus iterative Hough transformation detects numerous false fallen trees. The segment delineation and classification steps were introduced to reduce these false observations. Similar to CCC, false fallen tree removal reduces false observations, but also true observations.

2.3.3. Validation

Due to varying combinations of the used parameters, in total, 120 different fallen tree detections were obtained. All of these were validated using the reference data collected from the sample plots. The automatic validation process (Figure 3) consisted of the following steps:

1. Line segment matches were searched for each reference fallen tree using distance- and angle-based criteria. All line segments whose distance to the reference was $\leq$1 m and whose angle differed $\leq$10 degrees from the reference tree were determined as matches.
2. If no such line segments were found, the number of false negatives (FN) was incremented by one. If one or more such line segments were found, the number of true positives (TP) was incremented by one and the matched line segments were removed from the data so that they would not be matched with another reference tree.
3. Once all reference trees had been inspected, the number of remaining line segments was set as the number of false positives (FP).
4. The precision (Equation (1)), recall (Equation (2)), and F1-score (Equation (3)) were calculated and used as measures of fallen tree detection performance. Precision represents the proportion of true detections of all detections (i.e., user's accuracy), recall represents the proportion of reference trees that were detected (i.e., producer's accuracy), and F1-score combines both precision and recall into a single metric, aiming to present the performance of the method as a single value.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 * (precision * recall)}{precision + recall} \tag{3}$$

Overall precision, recall, and F1-score were calculated using all reference trees. In addition, the fallen tree detection performance was assessed by deadwood dimensions and state of decay. The reference trees were divided into five different classes based on their length, diameter, and state of decay, and class-specific recalls were calculated for each of these classes. Class-specific precision could not be calculated, as it would have been impossible to determine the true length/diameter/decay class of false positives.

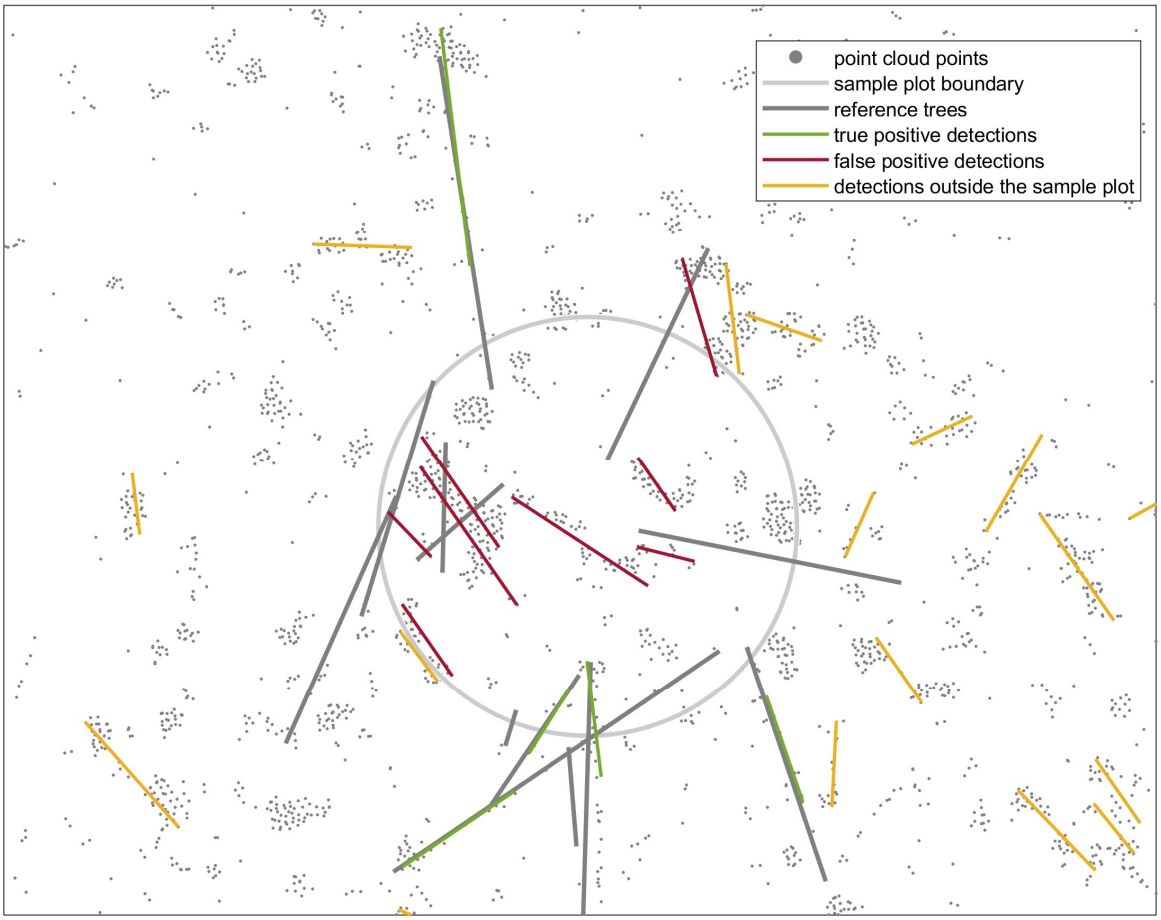

Figure 3. Validation on a single sample plot. The figure shows an example of reference trees and line segments detected by the fallen tree detection method on a single sample plot. Note that detection was performed on a buffered point cloud extending beyond the sample plot boundaries. This ensured that reference trees falling partially outside the sample plot could be detected. Using a buffered point cloud resulted in detections that could not be matched with a reference tree and that were located fully outside the sample plot boundaries. These detections (yellow lines in the figure) were discarded. Green lines depict detections that could be matched with a reference tree (true positives). In contrast, red lines depict detections that could not be matched with a reference tree and that were at least partially located within the sample plot (false positives).

2.3.4. Sensitivity Analysis

A sensitivity analysis was performed to reveal the impact of each varied parameter on the performance of the fallen tree detection method. Sobol's method [35,36] was used to examine how large a proportion of the total variance in the performance was accounted for by each of the parameters. Sobol's method is based on the observation that the variance in the output of a function can be expressed as the sum of the contributions to the variance by individual inputs and interactions between inputs (i.e., the variance can be decomposed into parts accounted for by individual inputs and their interactions). For example, in the case of four input variables, the variance in the output can be decomposed in the following way:

$$V_y = \sum V_i + \sum V_{ij} + \sum V_{ijk} + \sum V_{ijkh}, \quad (4)$$

where V_y is the total variance in the output variable, V_i represents the contribution of input variable i to the total variance, V_{ij} represents the contribution of the interaction between input variables i and j to the total variance, and so on. The contributions of individual

input variables and the interactions between two input variables can be calculated using (Equations (5) and (6)):

$$V_i = V[E_{x\sim i}[y|x_i]] \quad (5)$$

$$V_{ij} = V\left[E_{x\sim ij}\left[y|x_i, x_j\right]\right] - V_i - V_j. \quad (6)$$

In Equations (5) and (6), x_i and x_j represent individual input variables, E denotes the expected value, and $x\sim i$ represents that the value of x_i is fixed while the values of other variables vary. In practice, the values of one or more input variables were fixed while the other input variables were left to range within predetermined bounds, and the expected value of the output variable was then determined accordingly. The values of the fixed variables were changed between iterations to repeat the process for all possible combinations of the values of the fixed variables. The variance of the expected values was then computed. For variances including more than one fixed input variable, the smaller-order variances involving any combination of the fixed variables were subtracted to acquire the true interaction variance of the fixed variables.

The output of Sobol's method is a set of indices that represent the impact each input variable and each interaction of input variables has on the total variance (Equation (7)). These indices measure the proportional contribution of input variable i to the total variance in the output variable.

$$S_i = \frac{V_i}{V_y} \quad (7)$$

The sensitivity analysis was carried out separately for the ALS and ULS-based methods. Furthermore, in addition to examining how different parameters impacted fallen tree detection performance in general, Sobol's method was used for examining whether the impact of different parameters varied between fallen trees of different lengths, diameters, and decay states.

3. Results

3.1. Differences in Performance between the ALS and ULS Datasets

According to the sensitivity analyses carried out in this study, the highest performance (F1-score 0.39) of the ALS-based method for fallen tree detection was obtained using the following parameter combination: HR = 0.1 − 1.0 m, MNP = 3, CCC = Yes, FTR = Yes. Using this parameter combination, a recall of 0.33 denoted that 33% of the field-measured fallen trees could be detected, while a precision of 0.49 denoted that 49% of ALS-derived fallen trees were true observations. In contrast, the highest performance (F1-score 0.21) of the ULS-based method for fallen tree detection was obtained using the following parameter combination: HR = 0.2 − 1.0 m, MNP = 12, CCC = Yes, FTR = Yes. Applying this parameter combination resulted in a recall of 0.18 and a precision of 0.27 in fallen tree detection.

Figure 4 presents the precision and recall of the ALS and ULS datasets at different height ranges when CCC and FTR were not applied (left) and when both of these classifiers were applied (right). The figure shows that when the classifiers were not applied, the recall in fallen tree detection using the ULS dataset was higher than the recall using the ALS dataset, regardless of the values of MNP and HR. In contrast, precision was always higher for the ALS dataset. When both classifiers were applied (Figure 4, right), precision was still always higher for the ALS dataset, but with small values of MNP, the ALS dataset outperformed the ULS dataset also when recall was used as the measure of performance.

Figure 5 shows the fraction of variance in precision, recall, and the F1-score explained by different parameters and parameter combinations of the fallen tree detection method run on ALS data and ULS data. For the ALS dataset, MNP explained most of the variance in all accuracy metrics, but HR and the interaction between HR and MNP also had a notable impact on the accuracy of fallen tree detection. For the ULS dataset, the proportion of variance explained was divided more evenly among the parameters. HR was the most impactful parameter on precision and F1-score, but its impact on recall was smaller, although still notable. MNP still had a significant impact on recall and F1-score, but

the impact on precision was smaller. In contrast to the results on the ALS dataset, CCC explained a significant proportion of the variance in precision and recall, but the impact on the F1-score was negligible. This indicates that the impact on precision and recall evened out and thus the F1-score remained relatively constant regardless of whether CCC was used. FTR and the interaction of HR and CCC also had some impact on the performance.

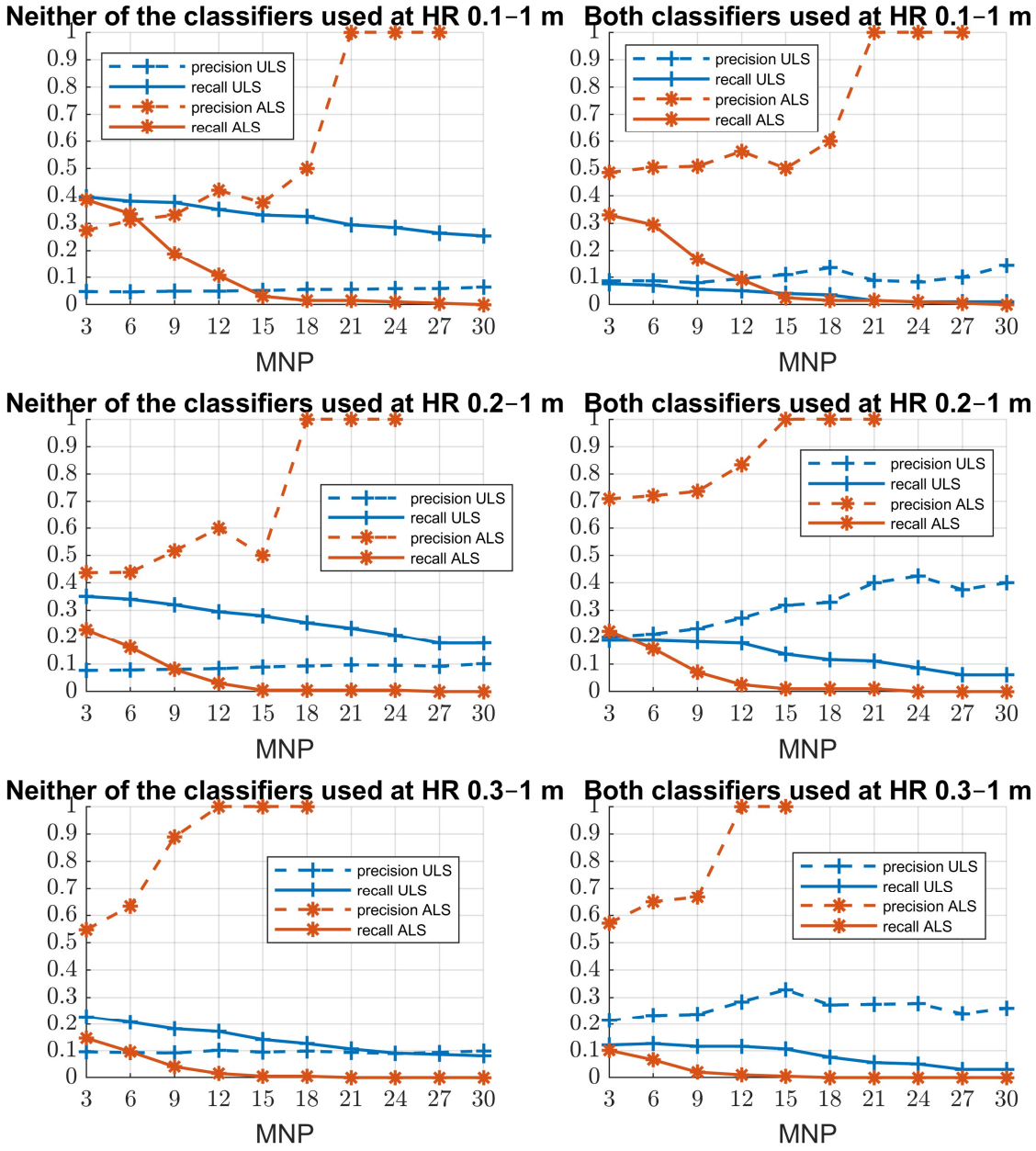

Figure 4. The precision and recall of the fallen tree method on the ALS and ULS datasets at different height ranges when the connected component classifier and the false tree remover were not used (**left**) and when both of these classifiers were used (**right**). MNP denotes the minimum number of points parameter.

Figure 5. Sobol's first-order indices for varied parameters and parameter combinations in ALS and ULS data. MNP denotes the minimum number of points that must fall on the same line for a line to be detected, HR denotes the height range from which fallen trees are searched, CCC denotes the connected component classifier, and FTR denotes false tree removal. For example, the row labeled ccc/mnp/ftr represents the fraction of variance in the output variable explained by the interaction between the CCC, MNP, and FTR parameters.

Figure 6 presents how the two machine-learning-based filters, CCC and FTR, impacted the results for ALS and ULS data. In the figure, each boxplot presents the distribution of the proportional difference in precision or recall between the case when no filters were used and the case when one or both of the filters were used. Figure 6 (orange) shows that, for ALS data, the median increase in precision was 18% when using CCC, 9% when using FTR, and 20% when using both filters. In contrast, the impact on recall was slightly smaller, with CCC, FTR, and the combination of both filters causing a median decrease of 10%, 0%, and 12%, respectively. Note that with several values of MNP, recall was zero when no filters were used, and thus using the filters did not change recall. As a result, the negative impact on recall caused by using the filters is somewhat underestimated. Based on Figure 6 (blue), the impact of the filters was larger on ULS data compared to the impact on ALS data. The median increases in precision caused by using CCC, FTR, and the combination of both filters were 93%, 54%, and 161%, whereas the median decreases in recall were 54%, 28% and 56%, respectively.

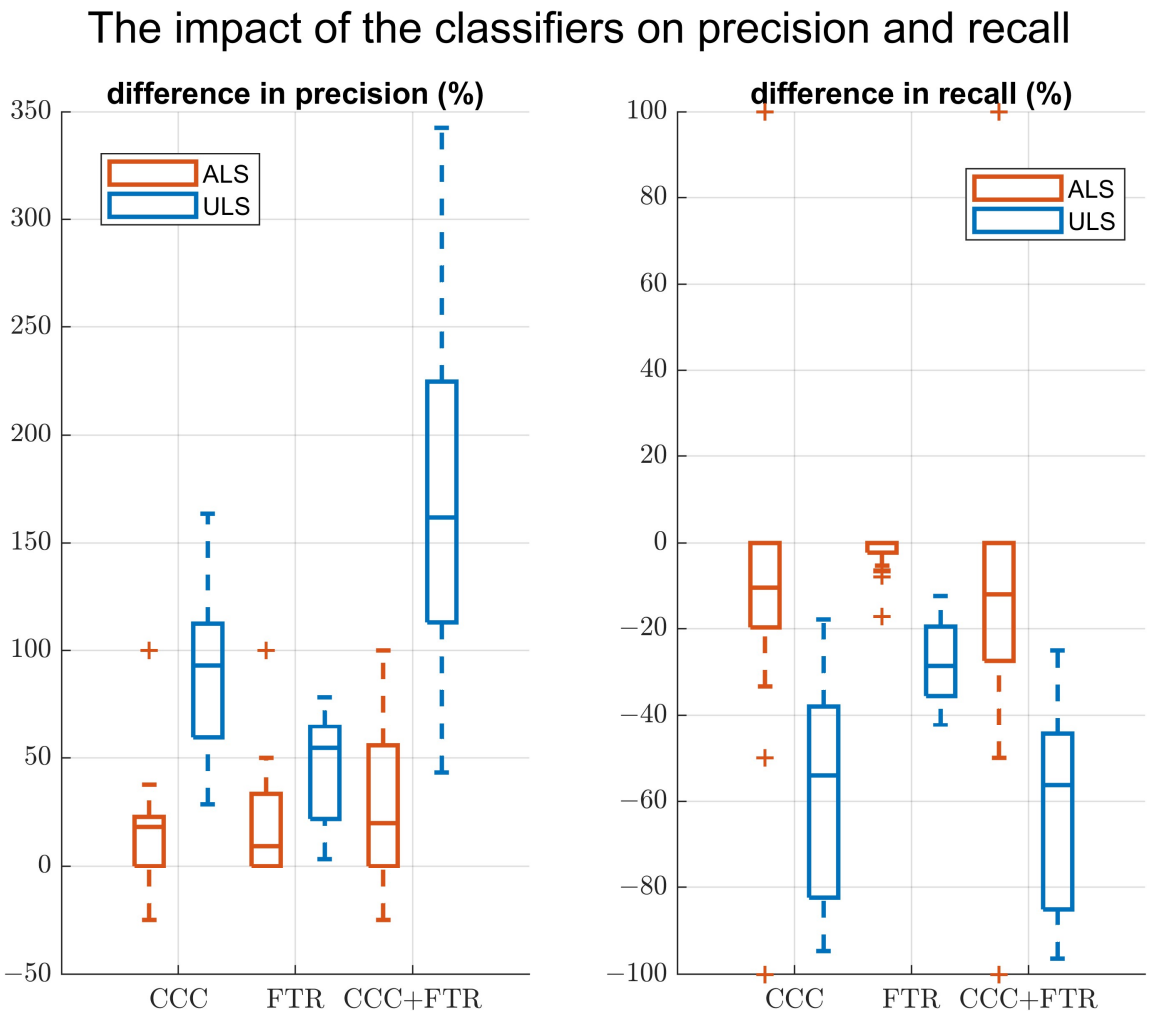

Figure 6. Boxplots of the proportional differences in precision and recall for the ALS and ULS datasets between the base case (no filters used) and cases when filters were used. CCC denotes that the connected component classifier was used, FTR denotes that the false tree remover was used and CCC+FTR denotes that both of the filters were used. For example, the leftmost boxplot on the left subplot shows that, on average, using the connected component classifier improved precisions of ALS data by approximately 18% compared to the case when no filters were used.

3.2. The Impact of Parameters on the Detection of Different Types of Fallen Trees

Figure 7 presents the fraction of variance in recall of different types of fallen trees explained by the parameters of the fallen tree detection method (i.e., Sobol's indices). The tree-type-specific Sobol indices largely reflected the general results (Figure 5), but there were some differences in how much changes in the parameters of the fallen tree detection method impacted trees belonging to different lengths, diameters, and decay classes. For ALS data (Figure 7, orange), the detection of large trees (length and diameter-wise) was more impacted by the MNP parameter than the detection of small trees. In contrast, the detection of small trees seemed to be more sensitive to HR and the interaction between HR and MNP. Heavily decayed trees were less sensitive to changes in the MNP parameter and more sensitive to the interaction between HR and MNP compared to trees in earlier stages of decay. For ULS data (Figure 7, blue), the trends between different types of fallen trees were more subtle compared to ALS data. The impact of CCC seemed to increase with tree length, indicating that long trees are more sensitive to this parameter than short trees. Long trees were more sensitive to MNP compared to short trees, whereas tree diameter had an opposite trend with this parameter. In contrast, long trees were less sensitive to the impact

of the interaction between HR and CCC compared to short trees, with tree diameter, again, showing an opposite trend. The impact of HR was larger for trees with small diameters compared to large-diameter trees. There seemed to be no clear trends with decay class apart from the impact of HR being larger for heavily decayed trees.

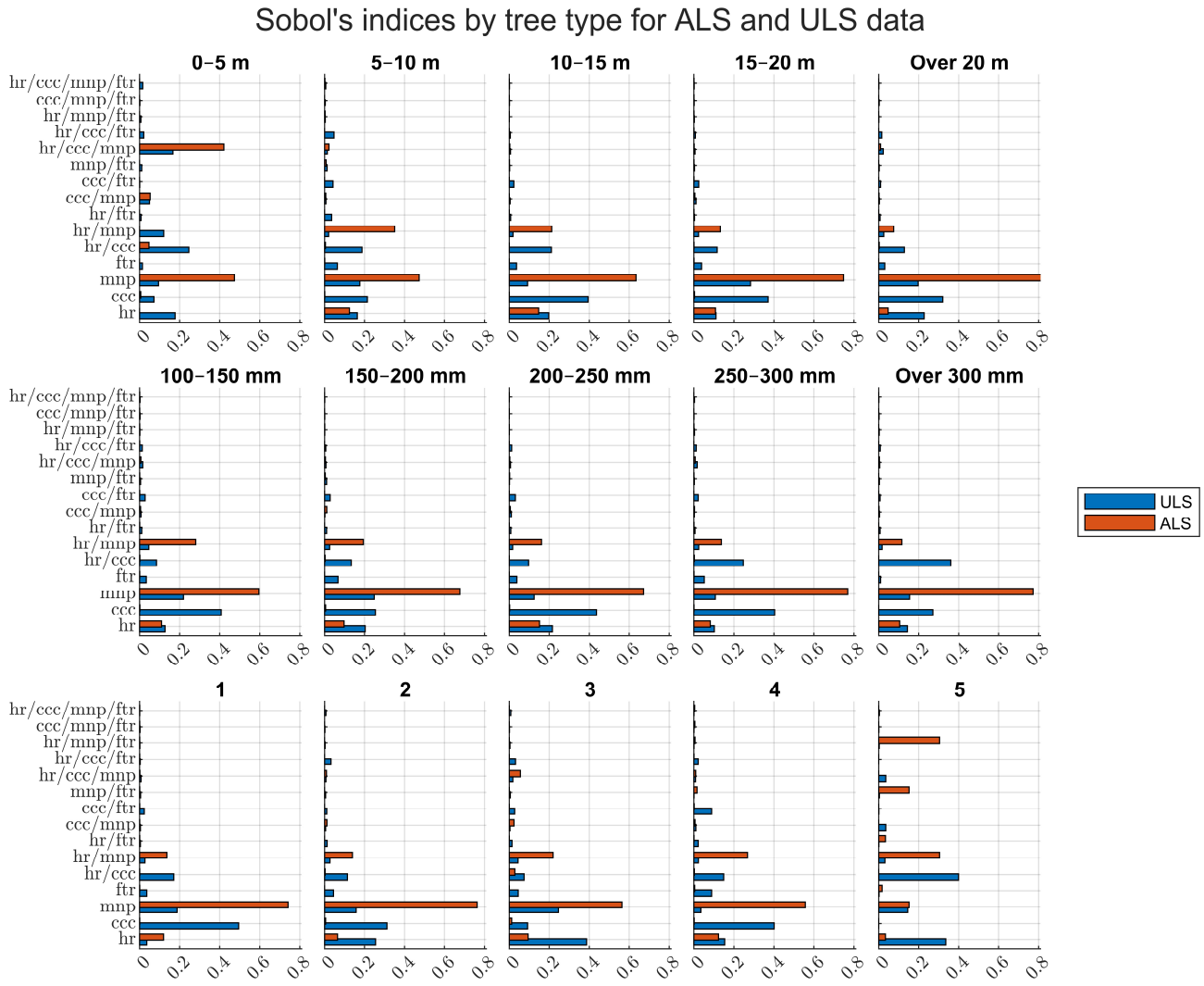

Figure 7. Sobol's indices by reference tree type for ALS data and ULS data. MNP denotes the minimum number of points that must fall on the same line for a line to be detected, HR denotes the height range from which fallen trees are searched, CCC denotes the connected component classifier, and FTR denotes false tree removal. For example, the row labeled hr/ccc/mnp in the top-left plot represents the fraction of variance in recall of trees in length class 0–5 m explained by the interaction between the HR, CCC, and MNP parameters.

4. Discussion

This study inspected how the performance of fallen tree detection varies between two types of laser scanning data: ALS data with a moderate point density and ULS data with a high point density. The inspection was based on a sensitivity analysis of the most important parameters of a line-detection-based fallen tree detection method. Based on the results, the fallen tree detection methodology should be adjusted based on the type of laser scanning dataset available.

The results of this study showed that, generally, the performance of fallen tree detection was actually better for the sparser ALS dataset than the denser ULS dataset. The best F1-score for ALS data was 0.39 (precision 0.49, recall 0.33) and 0.21 (precision 0.27, recall 0.18)

for ULS data. However, the results showed that the methodology used in this study favored the ALS dataset. The simplistic line-detection-based approach tested was rather sensitive to signs of linearity in the point cloud, allowing the detection of fallen trees even when they were represented by only a small linear group of points in the point cloud. As a tradeoff, the method was sensitive to noise (points originating from other sources than fallen trees, e.g., undergrowth) and generated a large number of false detections when the amount of noise was significant. For the ALS dataset, the method seemed to be rather suited, as it was sensitive enough to detect the linearities in the point cloud originating from fallen trees, but due to the moderate point density, the amount of noise was small. For the ULS dataset, the amount of noise was significantly higher resulting in a large number of false detections. The sensitivity of line detection was adjusted by varying the MNP parameter, but this mainly affected the ALS dataset, possibly due to the range of tested values of MNP not being wide enough.

One way to reduce noise in the near-ground point cloud is to use a lower HR threshold slightly above zero. The purpose of this threshold is to filter out falsely classified ground points that would otherwise generate false fallen tree detections. The tradeoff when using such a threshold is that trees left below this threshold cannot be detected and thus the optimal threshold value should filter out as many points not belonging to fallen trees as possible while retaining a majority of the fallen tree points. As expected, increasing the lower HR threshold decreased both the number of true and false detections in most cases (Figure 4). Based on the results, large trees were slightly less sensitive to the increase in the lower HR threshold (Figure 7), especially when coupled with an increase in the MNP parameter. This is logical, as large trees are more likely to occur higher from the ground than small trees. A similar result was reported by Mücke, et al. [17], who noticed that using a lower height threshold to exclude falsely classified ground points prevented the detection of small fallen trees. The ULS dataset was more sensitive to the HR threshold compared to the ALS dataset (Figure 5), as noise was a larger issue due to the higher point density. The inaccuracies in ground extraction are a result of small-scale variations in topography and ground vegetation that make distinguishing ground points from above-ground points a challenging task. The ground extraction method used in this study was based on only the geometric properties of points and their neighborhoods. Including point type and reflectance-related information in ground extraction could have perhaps increased the accuracy of ground extraction and thus decreased the number of false detections.

In addition to height-based filtering, this study utilized machine-learning-based filters/classifiers for reducing noise (CCC) and identifying false detections (FTR). CCC was applied after height-based filtering to remove point groups not originating from fallen trees, whereas FTR was used for filtering out false detections after detecting fallen tree candidates. Similar filters have been used before. Nyström, et al. [18] used linear discriminant analysis to remove false detections after detecting fallen trees. Polewski, et al. [20,21] used machine learning at three stages of the detection process. Firstly, they used a classifier for identifying point cloud points belonging to stems. Secondly, they used a classifier for classifying detected partially fallen tree segments as either true or false detections. Thirdly, they used machine learning to learn optimal parameter values when merging the partially fallen tree segments. However, neither of these studies inspected the individual effects of these machine-learning-based phases of the detection process, but rather inspected the performance of the detection method as a whole. Our goal was to inspect how machine-learning-based filters affect fallen tree detection at different stages of the detection process and in general. Both filters were, in fact, able to reduce the number of false detections, but as a tradeoff, they also reduced the number of true detections (Figures 4 and 6). CCC was more aggressive in reducing detections in general compared to FTR (Figure 6). Note, however, that it is easy to tune the sensitivity of machine-learning-based filters by simply changing the classification threshold. For example, a detection was classified as a true detection if FTR output a value above 0.5. Setting this threshold to a lower value would have resulted in fewer false detections being removed, but also fewer true detections being

removed. More interestingly, the best performance was achieved when using FTR after using CCC. In this case, precision improved, but recall remained almost constant. This effect was clearer with ULS data where using both filters resulted in an improvement in precision that was larger than the combined effect of the individual filters while recall decreased only slightly compared to the case when only CCC was used. This implies that reducing noise before detection somehow improved the performance of FTR. Perhaps CCC was able to remove certain types of point groups (e.g., small point groups) that would have generated a false detection, which FTR would have had a hard time detecting.

Both machine-learning-based filters used in this study were originally trained on the ALS dataset (see [23]). This shows in the results, as the performance of the filters on the ULS dataset was significantly more varied compared to the performance on the ALS dataset. Moreover, with ULS data, using CCC had a larger negative impact on the detection of large trees compared to small trees (Figure 7). This is possibly due to large trees forming large and dense groups of points in ULS data that look like non-tree point groups for the classifier that was trained on sparser ALS data. Both observations raise an important point: Machine-learning-based filters/classifiers do not directly generalize to different types of data and should thus be trained separately for each dataset. If this is not possible, another option is to only extract features that should be consistent between different datasets and build the filter using these features.

The results showed that machine-learning-based filters can improve the performance of fallen tree detection, but if such filters are to be used, they should be trained with examples extracted from the same or at least a very similar dataset to the dataset used for fallen tree detection. Furthermore, the performance of such filters was suboptimal even with the original ALS data on which they were trained. Perhaps machine-learning-based filters would perform better with higher point density datasets in which the smaller details of objects become visible. Machine-learning-based object detection (e.g., [37,38]) could be an interesting option that would reduce the need for the numerous human-made decisions related to the line-detection-based approach. In addition, the filters used in this study only used features related to the geometric properties of the point cloud. Including intensity- or echo-width-related features (see [17]) could perhaps be useful for distinguishing fallen trees from other near-ground objects.

When comparing the results when neither of the machine-learning-based filters was used (Figure 4, left), we see that the recalls for ULS data were higher than the corresponding recalls for ALS data. This is in line with what could be expected, as by keeping the values of other parameters fixed, an increase in point density would always result in more (or as many) detections, some of which are likely to be true detections. The downside is that a higher point density also results in a higher number of false positives, which, in turn, decreases precision. This was especially evident in this study, where the precision for ULS data was very low. Precision could be improved by filtering out false detections, but the machine-learning-based filters used in this study performed suboptimally on ULS data due to the filters being trained using the sparser ALS dataset. To summarize, using a higher-point-density dataset can potentially increase the proportion of detected trees, but the increasing false detection problem must be addressed for a better solution.

In addition to point density, the ALS and ULS datasets differed from each other in their scan pattern. The scan pattern of the ALS dataset was rather uniform, whereas the ULS dataset had a circular pattern. Visual inspection revealed a phenomenon with the ULS data that generated a significant number of false detections. In the areas where the diameter of the circular scan pattern was rather large, the pattern formed almost linear point groups, which were falsely detected as fallen trees by the line detection method. This induced a loss in precision, which, coupled with the negative impact of point density on precision, amplified the difference in precision between the ALS and ULS datasets. This might warrant an a priori filtering/harmonization of the UAV point cloud. Another approach could be to consider the scan parameter settings carefully to acquire a more

uniformly distributed point cloud in the first place based on the flight line placement and flight parameters (speed and altitude).

The sensitivity analysis performed in this study was not fully comprehensive. To limit the amount of computation and ensure interpretability, we used knowledge gained during method development and testing for restricting the sensitivity analysis in several ways. Firstly, we selected only a subset of the parameters for sensitivity analysis and fixed the other parameters of the line detection method to values found suitable during method development. Including all method parameters would have yielded a rather complicated analysis that would have been challenging to interpret altogether. Thus, we only included parameters that we, based on prior knowledge and experience, believed to have the largest impact on the performance of the method. Secondly, we used experimental knowledge to select the bounds of the parameter values used in the sensitivity analysis. For the binary parameters (whether to use either of the machine learning filters), the bounds were self-evident, but for the other parameters (MNP and HR), the parameter bounds could have been different. Especially with the MNP parameter, increasing the upper bound could have yielded interesting results for the ULS dataset. Now the performance on this dataset was not very sensitive to changes in the MNP parameter as opposed to the ALS dataset, where MNP was the most influential parameter.

A critical step in the fallen tree detection process is ground extraction, as inaccuracies in this hamper the detection of fallen trees and generate false detections. Several ground extraction methods were tested, and their parameters were varied to find the method and parameters that yielded virtually the best results. The best method–parameter combination was selected based on visual inspection. A more comprehensive search for the best method-parameter combination could have improved the results, but this would have been a tedious process, which possibly advocates for further investigations for operative purposes.

In the validation phase, reference trees were automatically matched with detected fallen tree segments. The automatic matching process was rather simplistic, as it was only based on the angle and distance between reference trees and detected segments. Thus, the matching procedure might have generated false matches in some cases. Furthermore, the matching procedure did not consider the length of the matched segment, and thus a reference tree was determined as found even if the matched segment covered only a small portion of it. A manual matching process would have likely been more accurate. However, this would have been a highly tedious task, as there were 120 different sets of segments to be matched with the reference trees (one for each parameter combination used for the fallen tree detection). With 197 reference trees, this would have meant manually inspecting $120 \times 197 = 23{,}640$ trees.

The ALS dataset was acquired in 2019, whereas the ULS dataset was acquired in 2020. Furthermore, some of the sample plots were scrutinized in 2019 while others were scrutinized in 2020. Between these two years, some new trees might have fallen on the sample plots. These trees would not be visible in the ALS dataset nor be included in the sample plots measured in 2019, which would distort the results. However, to our knowledge, there were no forestry operations or major storms in the study site during this time period, and thus the conditions in the forest likely remained rather stable.

From an ecological perspective, the results of this study provide some guidelines on how the sensitivity of direct fallen tree detection methodology should be adjusted to acquire the most reliable information regarding biodiversity hotspots possible. Direct fallen tree detection is a challenging task and reaching high detection accuracies for all types of fallen trees is virtually impossible, at least with the current laser scanning systems. A detection method aiming to detect fallen trees of all sizes will generate a large number of false detections, as the sensitivity of the method needs to be adjusted for the smallest trees. As a consequence, the reliability of the biodiversity-related information gained from mapping fallen trees is poor. To maximize reliability, direct fallen tree detection methods should be optimized for detecting large trees, which can be detected from laser scanning

data with relatively high accuracy [23]. This also makes sense from an ecological viewpoint, as large trees have the most ecological value.

5. Conclusions

This study investigated factors affecting the performance of a line-detection-based fallen tree detection approach with different types of ALS data. Investigated factors included point cloud characteristics, such as the point density and scan pattern, and methodological considerations related to parameter selection in a machine-learning-based line detection algorithm.

The results of this study imply that using a higher-point-density dataset potentially increases the proportion of fallen trees that can be detected. However, at least with the method used in this study, this comes with a significant loss of precision (i.e., the proportion of true detections of all detections is decreased significantly), as the line detection approach was sensitive to noise and data-acquisition-related patterns in the dense point cloud. Furthermore, the simplistic line detection approach was not able to take advantage of the finer details made visible by the higher point density. Thus, instead of a line detection approach, a more advanced method based perhaps on more complex shapes or patterns and laser scattering properties (reflectance, deviation, and echo type) in the point cloud could improve the results significantly when using high-density point clouds. One such method could be machine-learning-based object detection.

Author Contributions: Conceptualization, E.H., T.T. and M.V.; methodology, E.H., T.T. and T.M.; software, E.H.; validation, E.H. and T.T.; formal analysis, E.H. and T.M.; investigation, E.H., T.T. and T.H.; resources, A.K., T.H. and M.H.; data curation, E.H. and A.K.; writing—original draft preparation, E.H.; writing—review and editing, E.H., T.T., M.V., T.Y., A.K. and T.M.; visualization, E.H.; supervision, T.T.; project administration, T.T. and M.H.; funding acquisition, E.H., T.T., A.K. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Doctoral Program in Sustainable Use of Renewable Natural Resources (AGFOREE) at the University of Helsinki, the LIFE financial instrument of the European Union (Beetles LIFE (LIFE17/NAT/FI/000181)), the Academy of Finland's Strategic Research Council (IBC-Carbon, project number 312559), the Academy of Finland's Flagship Program (UNITE, project number 337127), and the Academy of Finland's Research Council for Natural Sciences and Engineering (Tandem Forest Values II: Estimating Forest Resources and Quality-related Attributes Using Automated Methods and Technologies, project number 334829). Open access funding provided by University of Helsinki.

Data Availability Statement: The source code of the fallen tree detection method used in this study is available in GitHub (https://github.com/Eikka12/Fallen_tree_detection; accessed on 7 January 2023). The laser scanning datasets used are not publicly available due to the authors only having partial ownership of the data. The datasets can be requested from the corresponding author.

Acknowledgments: The authors would like to thank Osmo Suominen and Aleksi Ritakallio for helping with field data collection. Open access funding provided by University of Helsinki.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Stokland, J.N.; Jonsson, B.G.; Siitonen, J. *Biodiversity in Dead Wood*; Cambridge University Press: Cambridge, NY, USA, 2012; pp. 1–509, ISBN 9781139025843.
2. Mönkkönen, M.; Aakala, T.; Blattert, C.; Burgas, D.; Duflot, R.; Eyvindson, K.; Kouki, J.; Laaksonen, T.; Punttila, P. More wood but less biodiversity in forests in Finland: A historical evaluation. *Memo. Soc. Fauna Flora Fenn.* **2022**, *98*, 1–11.
3. Ihalainen, A.; Mäkelä, H. Kuolleen puuston määrä Etelä-ja Pohjois-Suomessa 2004–2007. *Metsätieteen Aikakauskirja* **2009**, *2009*, 35–36. [CrossRef]
4. Andersson, L.I.; Hytteborn, H. Bryophytes and decaying wood—A comparison between managed and natural forest. *Ecography* **1991**, *14*, 121–130. [CrossRef]

5. Bader, P.; Jansson, S.; Jonsson, B.G. Wood-inhabiting fungi and substratum decline in selectively logged boreal spruce forests. *Biol. Conserv.* **1995**, *72*, 355–362. [CrossRef]
6. Ducey, M.J.; Williams, M.S.; Gove, J.H.; Roberge, S.; Kenning, R.S. Distance-limited perpendicular distance sampling for coarse woody debris: Theory and field results. *Forestry* **2012**, *86*, 119–128. [CrossRef]
7. Ståhl, G.; Gove, J.H.; Williams, M.S.; Ducey, M.J. Critical length sampling: A method to estimate the volume of downed coarse woody debris. *Eur. J. For. Res.* **2010**, *129*, 993–1000. [CrossRef]
8. Ståhl, G.; Ringvall, A.; Fridman, J. Assessment of Coarse Woody Debris: A Methodological Overview. *Ecol. Bull.* **2001**, *49*, 57–70.
9. Pesonen, A.; Maltamo, M.; Eerikäinen, K.; Packalèn, P. Airborne laser scanning-based prediction of coarse woody debris volumes in a conservation area. *For. Ecol. Manag.* **2008**, *255*, 3288–3296. [CrossRef]
10. Kraus, K.; Pfeifer, N. Determination of terrain models in wooded areas with airborne laser scanner data. *ISPRS J. Photogramm. Remote Sens.* **1998**, *53*, 193–203. [CrossRef]
11. Næsset, E.; Gobakken, T. Estimating forest growth using canopy metrics derived from airborne laser scanner data. *Remote Sens. Environ.* **2005**, *96*, 453–465. [CrossRef]
12. Næsset, E. Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data. *Remote Sens. Environ.* **2002**, *80*, 88–99. [CrossRef]
13. Wehr, A.; Lohr, U. Airborne laser scanning—An introduction and overview. *ISPRS J. Photogramm. Remote Sens.* **1999**, *54*, 68–82. [CrossRef]
14. Lefsky, M.A.; Cohen, W.B.; Acker, S.A.; Parker, G.G.; Spies, T.A.; Harding, D. Lidar Remote Sensing of the Canopy Structure and Biophysical Properties of Douglas-Fir Western Hemlock Forests. *Remote Sens. Environ.* **1999**, *70*, 339–361. [CrossRef]
15. Lefsky, M.A.; Cohen, W.B.; Parker, G.G.; Harding, D.J. Lidar Remote Sensing for Ecosystem Studies: Lidar, an emerging remote sensing technology that directly measures the three-dimensional distribution of plant canopies, can accurately estimate vegetation structural attributes and should be of particular interest to forest, landscape, and global ecologists. *Bioscience* **2002**, *52*, 19–30. [CrossRef]
16. Blanchard, S.D.; Jakubowski, M.K.; Kelly, M. Object-Based Image Analysis of Downed Logs in Disturbed Forested Landscapes Using Lidar. *Remote Sens.* **2011**, *3*, 2420–2439. [CrossRef]
17. Mücke, W.; Deák, B.; Schroiff, A.; Hollaus, M.; Pfeifer, N. Detection of fallen trees in forested areas using small footprint airborne laser scanning data. *Can. J. Remote Sens.* **2013**, *39*, S32–S40. [CrossRef]
18. Nyström, M.; Holmgren, J.; Fransson, J.E.S.; Olsson, H. Detection of windthrown trees using airborne laser scanning. *Int. J. Appl. Earth Obs. Geoinf.* **2014**, *30*, 21–29. [CrossRef]
19. Lindberg, E.; Hollaus, M.; Mücke, W.; Fransson, J.E.S.; Pfeifer, N. Detection of lying tree stems from airborne laser scanning data using a line template matching algorithm. *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.* **2013**, *II-5/W2*, 169–174. [CrossRef]
20. Polewski, P.; Yao, W.; Heurich, M.; Krzystek, P.; Stilla, U. Detection of fallen trees in ALS point clouds using a Normalized Cut approach trained by simulation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 252–271. [CrossRef]
21. Polewski, P.; Yao, W.; Heurich, M.; Krzystek, P.; Stilla, U. Learning a constrained conditional random field for enhanced segmentation of fallen trees in ALS point clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *140*, 33–44. [CrossRef]
22. Frome, A.; Huber, D.; Kolluri, R.; Bülow, T.; Malik, J. Recognizing Objects in Range Data Using Regional Point Descriptors. In Proceedings of the Computer Vision—ECCV 2004, Berlin, Heidelberg, 11–14 May 2004; pp. 224–237.
23. Heinaro, E.; Tanhuanpää, T.; Yrttimaa, T.; Holopainen, M.; Vastaranta, M. Airborne laser scanning reveals large tree trunks on forest floor. *For. Ecol. Manag.* **2021**, *491*, 119225. [CrossRef]
24. Jaakkola, A.; Hyyppä, J.; Kukko, A.; Yu, X.; Kaartinen, H.; Lehtomäki, M.; Lin, Y. A low-cost multi-sensoral mobile mapping system and its feasibility for tree measurements. *ISPRS J. Photogramm. Remote Sens.* **2010**, *65*, 514–522. [CrossRef]
25. Chisholm, R.A.; Cui, J.; Lum, S.K.Y.; Chen, B.M. UAV LiDAR for below-canopy forest surveys. *J. Unmanned Veh. Syst.* **2013**, *1*, 61–68. [CrossRef]
26. Jaakkola, A.; Hyyppä, J.; Yu, X.; Kukko, A.; Kaartinen, H.; Liang, X.; Hyyppä, H.; Wang, Y. Autonomous Collection of Forest Field Reference—The Outlook and a First Step with UAV Laser Scanning. *Remote Sens.* **2017**, *9*, 785. [CrossRef]
27. Wallace, L.; Lucieer, A.; Watson, C.; Turner, D. Development of a UAV-LiDAR System with Application to Forest Inventory. *Remote Sens.* **2012**, *4*, 1519–1543. [CrossRef]
28. Wallace, L.; Musk, R.; Lucieer, A. An Assessment of the Repeatability of Automatic Forest Inventory Metrics Derived From UAV-Borne Laser Scanning Data. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 7160–7169. [CrossRef]
29. Metsäntutkimuslaitos. VMI11. Available online: http://www.metla.fi/ohjelma/vmi/vmi11-maasto-ohje09-2p.pdf (accessed on 20 May 2019).
30. Zhang, K.; Chen, S.-C.; Whitman, D.; Shyu, M.-L.; Yan, J.; Zhang, C. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. *IEEE Trans. Geosci. Remote Sens.* **2003**, *41*, 872–882. [CrossRef]
31. Roussel, J.-R.; Auty, D.; Coops, N.C.; Tompalski, P.; Goodbody, T.R.H.; Meador, A.S.; Bourdon, J.-F.; de Boissieu, F.; Achim, A. lidR: An R package for analysis of Airborne Laser Scanning (ALS) data. *Remote Sens. Environ.* **2020**, *251*, 112061. [CrossRef]
32. Roussel, J.-R.; Auty, D. Airborne LiDAR Data Manipulation and Visualization for Forestry Applications. Available online: https://cran.r-project.org/package=lidR (accessed on 10 September 2022).
33. Duda, R.; Hart, P. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM* **1972**, *15*, 11–15. [CrossRef]

34. Hough, P.V.C. Method and Means for Recognizing Complex Patterns. US Patent 3,069,654, 18 December 1962.
35. Sobol, I. On sensitivity estimation for nonlinear mathematical models. *Mat. Model.* **1990**, *2*, 112–118.
36. Sobol, I. Sensitivity analysis for non-linear mathematical models. *Math. Model. Comput. Exp.* **1993**, *1993*, 407–414.
37. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars Fast Encoders for Object Detection from Point Clouds. *Comput. Res. Repos.* **2018**, *abs/1812.05784*, 12697–12705. [CrossRef]
38. Zhou, Y.; Tuzel, O. VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *Comput. Res. Repos.* **2017**, *abs/1711.06396*, 4490–4499. [CrossRef]