Faitli, Tamás; Hyyppä, Eric; Hyyti, Heikki; Hakala, Teemu; Kaartinen, Harri; Kukko, Antero; Muhojoki, Jesse; Hyyppä, Juha

Integration of a Mobile Laser Scanning System with a Forest Harvester for Accurate Localization and Tree Stem Measurements

Published in:
Remote Sensing

DOI:
10.3390/rs16173292

Published: 01/09/2024

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Faitli, T., Hyyppä, E., Hyyti, H., Hakala, T., Kaartinen, H., Kukko, A., Muhojoki, J., & Hyyppä, J. (2024). Integration of a Mobile Laser Scanning System with a Forest Harvester for Accurate Localization and Tree Stem Measurements. *Remote Sensing*, *16*(17), Article 3292. https://doi.org/10.3390/rs16173292

remote sensing

MDPI

Article

Integration of a Mobile Laser Scanning System with a Forest Harvester for Accurate Localization and Tree Stem Measurements

Tamás Faitli [1,†], Eric Hyyppä [1,†], Heikki Hyyti [1,*], Teemu Hakala [1], Harri Kaartinen [1], Antero Kukko [1,2], Jesse Muhojoki [1] and Juha Hyyppä [1,2]

1 Department of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute FGI, The National Land Survey of Finland, Vuorimiehentie 5, FI-02150 Espoo, Finland; tamas.faitli@nls.fi (T.F.); eric.hyyppa@nls.fi (E.H.); teemu.hakala@nls.fi (T.H.); harri.kaartinen@nls.fi (H.K.); antero.kukko@nls.fi (A.K.); jesse.muhojoki@nls.fi (J.M.); juha.hyyppa@nls.fi (J.H.)
2 Department of Built Environment, School of Engineering, Aalto University, P.O. Box 11000, FI-00076 Aalto, Finland
* Correspondence: heikki.hyyti@nls.fi
† These authors contributed equally to this work.

Abstract: Automating forest machines to optimize the forest value chain requires the ability to map the surroundings of the machine and to conduct accurate measurements of nearby trees. In the near-to-medium term, integrating a forest harvester with a mobile laser scanner system may have multiple applications, including real-time assistance of the harvester operator using laser-scanner-derived tree measurements and the collection of vast amounts of training data for large-scale airborne laser scanning-based surveys at the individual tree level. In this work, we present a comprehensive processing flow for a mobile laser scanning (MLS) system mounted on a forest harvester starting from the localization of the harvester under the forest canopy followed by accurate and automatic estimation of tree attributes, such as diameter at breast height (DBH) and stem curve. To evaluate our processing flow, we recorded and processed MLS data from a commercial thinning operation on three test strips with a total driven length ranging from 270 to 447 m in a managed Finnish spruce forest stand containing a total of 658 reference trees within a distance of 15 m from the harvester trajectory. Localization reference was obtained by a robotic total station, while reference tree attributes were derived using a high-quality handheld laser scanning system. As some applications of harvester-based MLS require real-time capabilities while others do not, we investigated the positioning accuracy both for real-time localization of the harvester and after the optimization of the full trajectory. In the real-time positioning mode, the absolute localization error was on average 2.44 m, while the corresponding error after the full optimization was 0.21 m. Applying our automatic stem diameter estimation algorithm for the constructed point clouds, we measured DBH and stem curve with a root-mean-square error (RMSE) of 3.2 cm and 3.6 cm, respectively, while detecting approximately 90% of the reference trees with DBH > 20 cm that were located within 15 m from the harvester trajectory. To achieve these results, we demonstrated a distance-adjusted bias correction method mitigating diameter estimation errors caused by the high beam divergence of the laser scanner used.

Keywords: harvester; lidar; IMU; SLAM; forestry; stem curve

check for updates

Citation: Faitli, T.; Hyyppä, E.; Hyyti, H.; Hakala, T.; Kaartinen, H.; Kukko, A.; Muhojoki, J.; Hyyppä, J. Integration of a Mobile Laser Scanning System with a Forest Harvester for Accurate Localization and Tree Stem Measurements. *Remote Sens.* **2024**, *16*, 3292. https://doi.org/10.3390/rs16173292

Academic Editor: Brenden E. McNeil

Received: 9 July 2024
Revised: 25 August 2024
Accepted: 30 August 2024
Published: 4 September 2024

1. Introduction

Many forest operations, such as thinning and logging, are carried out by machines, the operators of which have to continuously make quick decisions with a high impact on the forest ecosystem and the economy of the operation. Currently, there is a need for better tools to plan and optimize the forest value chain due to trade-offs between different ecosystem services provided by forests, such as economic use of timber for structural and fibre materials, preservation of biodiversity, and the use of the forest as a carbon sink. Partial or full automation of forest machines, such as forest harvesters and forwarders, may provide one solution to enhance forest operations and harvesting quality while reducing human error and resulting environmental damage [1]. A first step towards smarter and more efficient forest operations is to equip forest machines with smart perception and computational aid [2], which has recently become feasible thanks to advancements in laser scanning sensors and data processing of Mobile Laser Scanning (MLS) systems [3–11].

Automation of forest machines would require accurate real-time localization and information retrieval from the surrounding forest scene to support decision making. However, Kaartinen et al. [12] demonstrated that current Global Navigation Satellite System (GNSS) is insufficient for sub-1-meter positioning accuracy in a forest environment due to signal obstruction caused by the forest canopy. The positioning accuracy of a forest machine can be improved with the integration of a MLS system including an Inertial Measurement Unit (IMU) and lidar sensors that enable Simultaneous Localization and Mapping (SLAM) approaches for improved positioning [5,13]. The MLS system can also construct a 3D model of the surrounding forest scene that enables automatic detection, classification, and measurement of trees.

Such automatically derived information of the forest surroundings could be used to support and optimize decision making, e.g., during a thinning or logging process, which may ultimately enable the preservation of valuable habitats, species, plants, retention trees, decaying wood, and most vital trees while harvesting the economically most profitable trees and/or those with stem or canopy defects. Productivity improvements may also be expected even for partial automation of the decision making since the cutting quality and productivity of human operators have been shown to improve by prior markings of the trees to be harvested [14]. Potential other benefits of increased automation include energy efficiency as a result of more optimized operation of forest machines. These intended applications would require onboard and real-time processing of the MLS data that often sacrifices precision and accuracy. On the other hand, some other applications of harvester-based MLS data would favour accuracy and allow a time-consuming post-processing solution instead of real-time computation. Examples of such applications include the collection of reference measurements for large-scale Airborne Laser Scanning (ALS) surveys [15,16], updating existing forest inventories, and generating reports about the tree logs left on the ground after a harvesting operation to improve their logistics.

Currently, there are only a few scientific studies or industry reports in which laser scanners have been mounted on a forest harvester to study the automatic mapping of forests [7,17–19]. These previous studies have mostly focused on the localization of the harvester within a forest environment [7] or only proposed concepts without adequate evaluation of the presented methods in real experiments [17,18]. To date, none of these previous studies have focused on estimating tree attributes, such as the stem curve, for the trees surrounding the harvester using an onboard MLS system. Some early studies have utilized an all-terrain vehicle (ATV) or a mobile robot as a proxy of a forest harvester and developed methods for positioning the ATV and detecting the surrounding trees [20–23], but a demonstration of the methods with a real harvester system is lacking.

Beyond the context of forest machines, a multitude of studies have recently focused on improving lidar-based positioning of mobile platforms. Over the past years, lidar-based positioning has shifted to general feature-based or featureless approaches [24], often aided by additional sensors such as an IMU or camera. These methods are primarily developed in the urban scene for autonomous driving, with little to no experiments in forest environments. Some of the most popular frameworks include [25–28]. An alternative approach for positioning MLS systems under the forest canopy relies on using a prior map of the forest area of interest, to which features detected from the lidar data are matched as discussed, e.g., in [17,29–32].

During the past decade, efficient estimation of tree attributes has been demonstrated using various mobile laser scanning systems including, e.g., handheld systems [6,33–35], backpack-based systems [8], systems based on ground vehicles [5,10,36], under-canopy flying UAVs [9], and above-canopy flying UAVs [37]. Recent advancements in the processing of MLS point clouds have enabled accurate measurements of tree attributes, such as stem curve and volume, with a low RMSE of 5–10% even in the presence of decimeter-level point cloud distortions [9,38]. Thus, comparable performance may be achieved with an MLS system onboard a forest harvester if the localization of the harvester and real-time computation aspects can be solved.

Currently, there are thus no prior studies presenting a full processing flow for harvester-based MLS demonstrating tree attribute measurements with a sufficient accuracy for operational applications. Even the post-processing solutions providing high-quality maps of the harvester surroundings are largely missing to date. In typical MLS data acquisition, point clouds from plots are covered with a looped square-like figure providing loop closure detection possibilities for the SLAM algorithm, which results in improved positioning and modelling of tree stems. With a forest harvester, such looped paths are not typical, which provides challenges both for positioning and stem detection.

In late 2022, Ponsse, one of the world's leading manufacturers of cut-to-length forest machines, launched a technology concept introducing a world-first thinning density assistant based on mobile lidar onboard a harvester [2]. The current system aids the operator to cut down an optimal number of trees based on a target thinning density and the number of detected trees in the surroundings of the harvester. However, the system does not estimate the stem diameter or the volume of the detected trees which could be used to produce a more detailed estimate of the remaining basal area and biomass.

In this work, we present a full processing flow for an MLS system onboard a forest harvester starting from the localization and mapping of the harvester followed by accurate, automatic measurements of the surrounding trees. The methods are evaluated on real data collected using a forest harvester during a thinning operation on three test strips in a boreal spruce forest stand located in Finland. We study the positioning accuracy of the developed localization solution for two different modes, one representing real-time operation during thinning, and the other optimization of the full trajectory after the harvesting. The reconstructed 3D point clouds of the surroundings are further tested for algorithmic tree stem measurements conducted in a post-processing mode to assess the quality of the obtained point clouds and to estimate the potential of tree attribute measurements with the current sensors. We provide a detailed evaluation for the accuracy of the estimated tree attributes and study the dependence of errors on scanner-to-tree distance and tree size. Furthermore, we demonstrate a distance-adjusted correction for stem diameter bias, caused mainly by finite laser beam divergence, allowing a significant improvement in the bias and RMSE of the estimated stem curves and diameters. Importantly, further improvements and optimization of the algorithms may enable real-time tree attribute estimation in the future.

2. Materials and Methods

2.1. Harvester System

We studied a custom MLS system attached onto a Ponsse Ergo forest harvester (Ponsse, Finland) equipped with a C44 crane and an H7 harvester head as illustrated in Figure 1a. As shown in the inset A of Figure 1a, our custom MLS system consisted of an Ouster OS0-128 Rev C (Ouster, San Francisco, CA, USA) lidar sensor with properties detailed in Table 1 and a Lord Microstrain 3DM-GQ7 (Microstrain by HBK, Williston, VT, USA) used as an IMU. Furthermore, a prism of the Leica Nova TS60 robotic total station (Leica Geosystems, Heerbrugg, Switzerland) was attached to the harvester cabin for the acquisition of reference positioning measurements as explained in Section 2.4.1.

The forest harvester was composed of a cabin, a front link, and a crane, all of which could move with respect to each other, thus necessitating the use of different coordinate frames for each of the parts as schematically illustrated in Figure 1b. We mostly pay attention to the two main rigid bodies of the harvester, the first being the front link hosting the MLS sensor system and the second being the cabin equipped with the total station prism. These two bodies had different trajectories as they are not rigidly connected. Our real-time positioning method was primarily developed to estimate the trajectory of the MLS system at the front link.

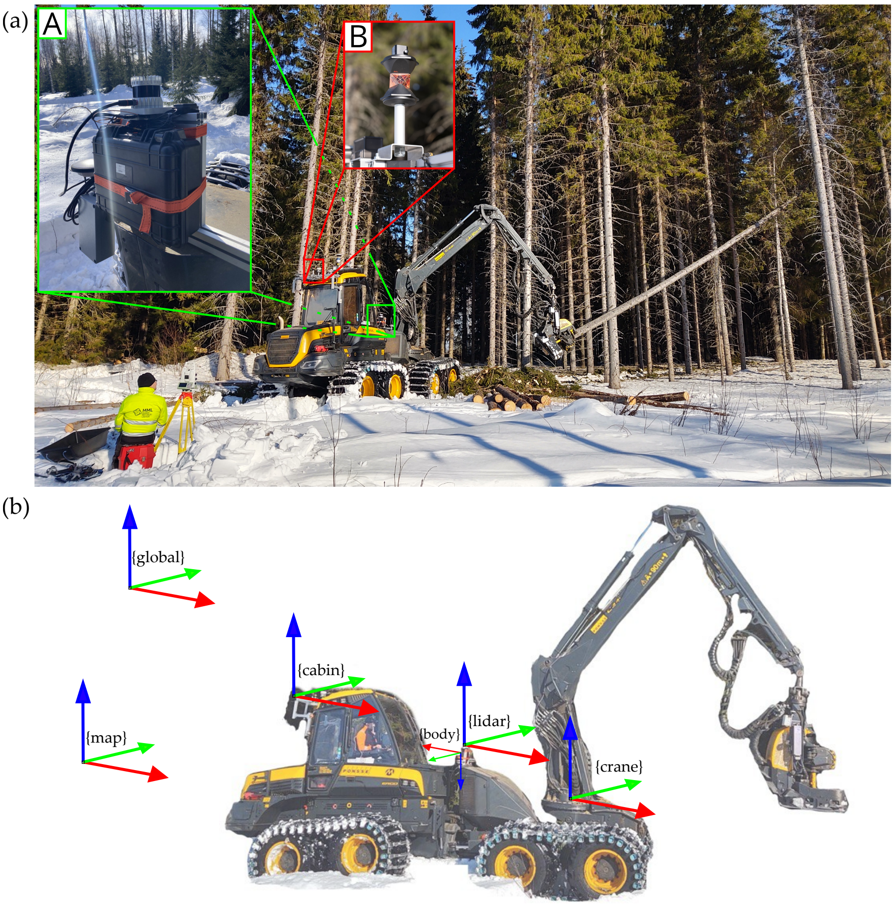

Figure 1. (**a**) Ponsse Ergo forest harvester with the attached mobile laser scanning system (inset **A**) consisting of an Ouster OS0-128 lidar and a Lord Microstrain 3DM-GQ7 GNSS/INS unit. (Inset **B**) shows the prism of the Leica Nova TS60 total station. The total station itself is located on the lower left corner of the image. (**b**) Schematic illustration of the harvester system and the coordinate frames. We label the point of reference positioning measurements as cabin frame, the origin of the lidar sensor on the front link as lidar frame, and the origin of the IMU sensor also on the front link as body frame, which is the frame whose trajectory is estimated firsthand by our SLAM. Furthermore, map frame denotes an arbitrary stationary coordinate frame constructed at the start of a measurement according to an initial fix of the frame, while the global frame is a georeferenced stationary frame.

In addition to the MLS data, our positioning approach relied on the availability of rotation angle measurements between the harvester crane and the front link, enabling us to remove the moving crane from the lidar data. The rotation angle measurements could be provided by an additional built-in angular position sensor in the harvester (e.g., an encoder), but in this work, we manually derived the angles from the lidar data by annotating the position of the harvester head, from which the angle was computed. Furthermore, our system also relied on the angle between the front link and the cabin when observed from the top view. This angle was automatically computed by registering a point-based model of the cabin onto the lidar data.

Table 1. Important properties of the Ouster OS0-128 Rev C lidar sensor as presented in the sensor datasheet [39]. Here, vertical resolution refers to the number of channels in the vertical direction, whereas the horizontal resolution denotes the number of fired laser beams for a given channel during a single rotation. The pulse repetition rate is a product of the vertical and horizontal resolutions and the rotation rate. The reported beam divergence corresponds to the full width at half maximum.

Property	Value for Ouster OS0-128 Rev C
Range	45 m for >90% detection probability
	50 m for >50% detection probability
Range accuracy	±3 cm for lambertian targets
	±10 cm for retroreflectors
Field of view (vertical)	90% (−45° to +45°)
Field of view (horizontal)	360°
Vertical resolution	128
Horizontal resolution	1024
Rotation rate	10 Hz
Pulse repetition rate	1.31 MHz
Laser wavelength	865 nm
Beam diameter exiting sensor	5 mm
Beam divergence	6.1 mrad (0.35°)

2.2. Test Site

The experiments with the harvester MLS system were conducted in March, 2022 on a managed Finnish forest stand located in Tuusniemi, Eastern Finland (62.7°N, 28.5°E); see Figure 2a. In the experiments, the harvester conducted the last commercial thinning operation before the final felling planned to take place in a few decades, and simultaneously data were collected with the MLS system. The 1.7 ha forest test site had two stands, one on each side of a gravel road going through the area. The thinning operation with the data collection was performed on six (6) test strips harvested individually as shown in Figure 2b. On the day of the experiments, the weather was clear and sunny with the temperature ranging from −10 °C to slightly over zero Celsius. In the forest, there was about 80 cm of snow on the ground.

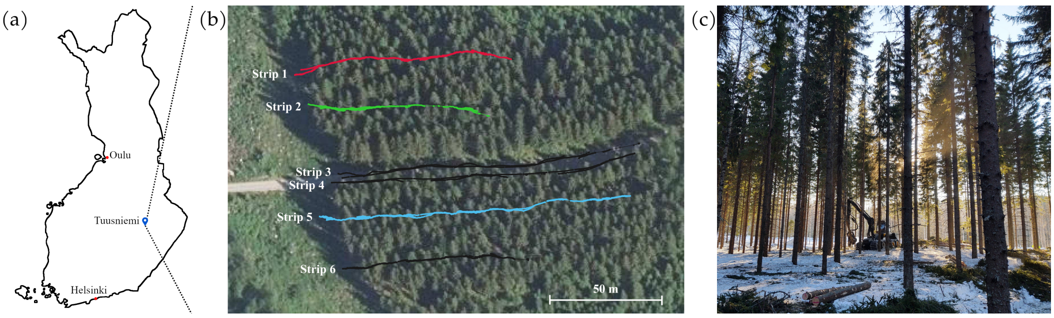

Figure 2. (**a**) The test site is located in Tuusniemi, Finland. (**b**) Six test strips on the site. The test strips 1 (red), 2 (green), and 5 (blue) were used to study the harvester localization and stem attribute estimation. (**c**) Harvester working on the test site (photo by Ville Kankare).

In this work, the test strips 1, 2, and 5 were used to evaluate the accuracy of our positioning approach and the automatic stem attribute measurements since these test strips were fully located in the forest with sufficient distance from the gravel road. As can be seen from Figures 1a and 2c, Norway spruce (*Picea abies* (L.) H.Karst.) was the dominant tree species on the test site. Descriptive statistics of the trees on the test strips are provided in Table 2.

Table 2. Descriptive statistics of the test strips and reference trees located within a distance of 15 m from the harvester trajectory. The statistics of reference trees were obtained using a ZEB Horizon laser scanner (see Section 2.4.2). The test strip length refers to the shortest distance between the starting location of the measurement and the location, at which the harvester turned back. For DBH, tree height and stem volume, we report the standard deviation within the parentheses.

Test Strip	Length of Strip (m)	Number of Trees	Stem Density (1/ha)	Mean DBH (cm)	Mean Tree Height (m)	Mean Stem Volume (m^3)
1	93	252	630	22.3 (±7.6)	18.3 (±4.2)	0.47 (±0.36)
2	76	190	540	27.0 (±7.9)	21.4 (±3.9)	0.75 (±0.45)
5	139	216	490	27.9 (±5.7)	22.7 (±2.4)	0.79 (±0.34)

2.3. Data Acquisition with the Harvester System

In our tests, the harvester was driven into the forest starting from the edge of the forest, following an approximately straight line from west to east direction. The cutting was planned to be as similar as possible to a standard commercial thinning operation but still allowing good reference measurements. On its way, the harvester was cutting trees, and it was driven as long as the reference position could be measured using the total station (see Section 2.4). Once reaching too far, the operator was called to turn back, after which trees were no longer cut and the machine was driven straight back in reverse to the starting point. Typically, the data collection on each test strip took 25–40 min, of which only the last few minutes corresponded to the reverse direction. When a tree was cut, primarily the crane was moving and the rest of the machine remained stationary; however, at those times the sensors were subjected to large vibrations due to the falling trees.

We recorded the data into rosbag files, using a Jetson Xavier AGX unit (NVIDIA, Santa Clara, CA, USA) running the ROS drivers provided by the manufacturers of the sensors. The sensors were synchronized on hardware level using a pulse-per-second (PPS) signal and NMEA messages. The point clouds obtained as a result of the data acquisition after subsequent positioning correction (see Section 3.1) had an average point density of approximately 2×10^4 pt/m^2 within 15 m from the harvester trajectory, as discussed in more detail in Section 4.2.

2.4. Reference Data

2.4.1. Reference for Harvester Positioning

Reference position measurements were collected using a Leica Nova TS60 robotic total station with a distance accuracy of 3 mm + 1.5 ppm according to the manufacturer [40], sampled at 3 Hz on average. The reference position was measured using a prism located at the top of the harvester cabin as shown in Figure 1a. As the position was measured on each of the test strips until the total station was no longer capable of measuring the location of the cabin, there was a short missing segment with no reference position measurement at the eastern end of each strip, which is ignored for the evaluation of the positioning accuracy.

2.4.2. Reference for Tree Attributes

A handheld ZEB Horizon scanner (GeoSLAM, Nottingham, UK) was utilized to collect high-quality point cloud data across the studied forest area to conduct reference measurements of stem attributes. The ZEB Horizon system includes a rotating Velodyne VLP-16 scanner and an integrated SLAM system that enable accurate mapping of the forest. Several previous studies have shown that the ZEB Horizon system and its predecessors enable measurements of the stem diameter with a low RMSE on the order of 1–2 cm, thus rendering it as an efficient alternative for collecting reference data for stem attributes, such as DBH or stem curve [3,6,33,34,38,41–43].

Data collection with the ZEB Horizon system was conducted in two parts to avoid excessively large files in SLAM processing. As the test site was naturally divided to the northern and southern part by the road running through the test site (see Figure 2), these parts were scanned separately. To minimize the drift in SLAM processing, the data were collected such that there were crossings in the traversed path and the data collection was finished at the start location to produce a closed loop. GeoSLAM Hub (version 6.0.0) software was used to process the collected raw data to point clouds in laz-format.

We analyzed the point clouds using the automatic algorithm described in Hyyppä et al. [38] to obtain reference values for the locations, DBHs, stem curves, tree heights, and stem volumes of the trees on the test strips. Here, the algorithm for the automatic detection of the reference stems consists of the same steps as that used for the analysis of the harvester-derived lidar data (see Section 3.2.2), but the algorithm for DBH and stem curve estimation differs from Sections 3.2.3 and 3.2.4. In our previous study using the ZEB Horizon system and a similar algorithm [38], the resulting RMSE of DBH estimation was 0.9–1.3 cm while detecting 77–93% of trees in comparable boreal forest conditions. For the stem curve estimation, the reported RMSE increased from 3–4% at the breast height to 7–8% at the height of 6 m above ground. Based on these previous results with a similar algorithm in comparable forest conditions, we regard the DBHs and stem curves acquired with the ZEB Horizon system as accurate enough to be used as a reference for the current study while noting that the expected error of the reference diameters is on the order of 1 cm.

3. Algorithms for Real-Time Localization and Offline Tree Attribute Estimation

In this section, we present our algorithms utilizing the harvester MLS data for real-time positioning of the harvester (see Section 3.1) and automatic measurements of the surrounding trees (see Section 3.2). We study two modes for the positioning, one targeted for real-time onboard operation named as *online trajectory with initial fix*, and the second based on the final state of the trajectory with offline-computed georeferencing, named as *final trajectory with optimized fix*. For the tree attribute measurements, we also investigate two parameter modes named as *tree map* and *accurate attributes*, the first of which enables a high detection rate of trees, while the latter favours accuracy of the estimated stem attributes. See also Section 5.3 for discussion on the connection between potential applications of harvester-based MLS and the studied modes of positioning and tree attribute estimation.

3.1. Algorithms for Real-Time Localization and Mapping from MLS Data

Our SLAM algorithm takes the lidar and inertial measurements from the sensor drivers as inputs and maintains estimates of position, orientation, velocity, and IMU biases while correcting lidar observations. Lidar measurements are first expressed in the *lidar coordinate frame*, which we denote using the symbol $\mathcal{L}$. Furthermore, we use the term *scan* for points accumulated throughout one whole revolution of the scanner. The k-th incoming scan can therefore be formulated as a set of 3D points and their corresponding timestamps as follows:

$$^{\mathcal{L}}\mathcal{S}_k := \{(\mathbf{p}_i, t_i) | \mathbf{p}_i \in \mathbb{R}^3, t_i \in \mathbb{R}^+\}_{i=1,\dots,N_{\text{scan}}}\ , \tag{1}$$

where N_{scan} is the number of points in one scan. We further denote the body frame using the symbol $\mathcal{X}$ that is chosen to coincide with the IMU sensor frame. Similarly to the set of 3D points, we also group together IMU measurements collected during the kth scan, which we denote as $^{\mathcal{X}}\mathcal{I}_k$. We further use the symbol $\mathcal{C}$ to denote the cabin coordinate frame that coincides with the total station prism attached on the forest harvester. During the initialization of the SLAM algorithm, we align the map frame to the body frame and use it as a local map frame for positioning and mapping purposes. Finally, we also utilize a georeferenced global frame, mainly to interpret the reference data used during the performance analysis.

Our SLAM solution has been briefly presented previously in a conference paper by Faitli et al. [44]. Here we provide a more comprehensive description, extended with new elements considering the forest harvester as a special platform carrying the MLS system. Additional differences arise in the analysis of the positioning performance, presented in Section 3.1.3.

3.1.1. Scan Pre-Processing

Each incoming scan first goes through a pre-processing step. In this step, we compensate for the motion of the lidar during the scan, filter out points collected from the forest harvester itself, transform the scan into the body frame, and downsample the scan. In order to remove the motion distortion, we first estimate the trajectory during the scan by integrating the available IMU measurements as described in [45]. The measurement rate of the IMU is lower than that of the laser scanner, and hence, we compute a pose $\mathcal{X}_{t_i} \in \mathrm{SE}(3)$ for each point within the scan by interpolating between the closest trajectory elements $(\mathcal{X}_{m-1}, t_{m-1})$ and $(\mathcal{X}_m, t_m)$ in time as

$$\mathcal{X}_{t_i} = \mathcal{X}_{m-1} \oplus \left[\frac{t_i - t_{m-1}}{t_m - t_{m-1}} (\mathcal{X}_m \ominus \mathcal{X}_{m-1}) \right], \tag{2}$$

where operations $\oplus$ and $\ominus$ are used as defined in [46], and they are essentially plus and minus operators defined on the elements of Special Euclidean Group (SE(3)) with the help of their Lie Algebra. Finally, we compute the pose difference between $\mathcal{X}_{t_i}$ and the pose at the beginning of the scan to transform the point accordingly. Repeating this for each point in the scan, we obtain a new reconstructed scan, which is represented with a unified coordinate frame. In practice, it was enough to compute a pose for each vertical column of points as they were all assigned the same timestamp on the hardware level.

Due to the kinematics of the forest harvester and the scanner position, there were numerous lidar points recorded from the moving harvester itself. These points are not part of the static environment, and mapping them would result in a lot of noise. We filter them out by observing the scan from the top view and align rectangles surrounding the cabin, front link, and crane as illustrated in Figure 3. Then, we remove all points that are bounded by any of the rectangles. For this, we utilize the angle α between the front link and the crane, and the angle β between the front link and the cabin. The sides of the rectangles were set by manually measuring the dimensions of the machine components from the scans, with an additional safety margin. As the crane can expand during operation in order to grab the trees, the ideal bounding rectangle size varies a lot from scan to scan. However, given we only know the angle α, we always set its bounding rectangle assuming it expanded as far as possible.

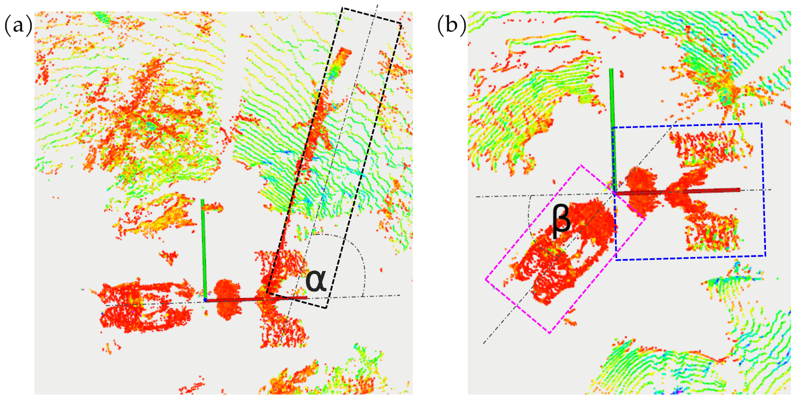

Figure 3. Top view of a single lidar scan illustrating the removal of points corresponding to the forest harvester. The black, blue, and magenta rectangles show bounding boxes for the harvester crane, the front link, and the cabin, respectively, all of which are filtered out from the scan. (**a**) We align the black rectangle to bound the points obtained from the crane with the help of the angle α between the crane and the front link. (**b**) We fit a static blue rectangle around the front link (as it does not move with respect to the lidar) and a magenta rectangle around the cabin with the help of the angle β.

In addition to removing the harvester points, we also remove all points that are closer than 1 m to the scanner origin or further away than 50 m. The remaining reconstructed scan is then transformed into the body frame by using a fixed calibration transformation, which in our case was obtained by measuring the distance and orientation between the lidar and IMU sensors with hand tools. Finally, the scan is downsampled using a voxel filter, where the points within a voxel are approximated with their centroid to reduce computational time.

3.1.2. Smoothing and Mapping Framework

The block diagram of our SLAM solution is presented in Figure 4. The processing is triggered by receiving a new scan from the lidar. The raw lidar measurements first go through a *scan preprocessing* step as detailed previously. Then, in blocks *scan registration* and *IMU preintegration,* we are computing observations from the reconstructed scan and raw IMU data about the current state. We define the updated state estimate at the k-th incoming scan as

$$\mathbf{x}_k := \left[{}^{\text{map}}\mathcal{X}_k, {}^{\text{map}}\mathbf{v}_k, {}^{\mathcal{X}}\mathbf{b}_k\right], \tag{3}$$

where $\mathcal{X}_k$ is the pose, $\mathbf{v}_k$ is the velocity of the body frame, and the IMU biases are denoted by $\mathbf{b}_k$. Furthermore, we define the map as a set of selected historical trajectory elements $\mathbb{X} = \{\mathcal{X}_l\}_{l=0,\ldots,L}$ and a set of corresponding pre-processed scans $\mathbb{S} = \{{}^{\mathcal{X}}\bar{\mathcal{S}}_l\}_{l=0,\ldots,L}$. Within the core of the algorithm, two factor graphs are maintained in parallel. The first one (in block *graph optimization*) is responsible for maintaining the online estimate $\mathbf{x}_k$. It is built by lidar and IMU factors and has a more rigid structure due to the lidar factors acting as anchors on each node individually. Meanwhile, the second graph (in block *map graph optimization*) is responsible for the mapping, and it only has constraints that connect two nodes together, providing a much more flexible structure for long term corrections after loop closures. This structure is essential for the real-time computation, with the intention of executing the second (more expensive) graph as few times as possible. To decide when to execute the mapping graph, we implement a keyframe selection strategy. At times when a new node is added to the mapping graph, we first check for a loop closure observation in the block *loop closure check*, and we then optimize the map graph using the lidar and loop closure factors and finally recompute the local map with the current best estimates in block *local map update*.

Keyframe selection: The keyframe selection strategy decides whether to execute the mapping graph within a given update or not. We apply a strategy based on travelled distance from the last node added to the graph. Both positional and angular movements contribute to this travelled distance, while their contribution was tuned such that 10 degrees of rotation is equivalent to 1 m lateral distance. Finally, when a preset threshold distance is surpassed (0.2 m in our case), a new node is added to the mapping graph.

Node selection for local map: The local map is reconstructed every time a new keyframe is added to the mapping graph. It samples historical scans with their corresponding pose estimates and concatenates them in the map coordinate frame. The sampling is performed by considering the last N ($N = 16$ in our experiments) nodes. We first apply a voxel filter on the position component of these N nodes, followed by selecting the closest original node to each of the voxel center points.

Lidar factor: Constraints are generated for the factor graphs by registering the scan to the local map. The registration is performed using the Normal Distributions Transform (NDT) algorithm with point-to-distribution error terms. The constraints are modelled as Gaussians where the mean is in the SE(3) and the uncertainties are expressed on its corresponding tangent space. The mean observation for the first graph is obtained from the registration as it is. Meanwhile, the mean observation for the map graph is computed as the difference between the latest pose in the map graph and the registration result. The noise models are computed by scaling a preset diagonal vector using the fitness score of the registration, where the fitness score is evaluated using the default implementation of the PCL library.

IMU factor: We generate additional constraints for the first factor graph by preintegrating the time corresponding IMU observations as proposed by Dellaert and Kaess [47].

Bias factor: In order to constrain the bias values in the first factor graph, we model the change of bias values between two consecutive scans. We assume it does not change; however, we leave some room for correction and model it as a zero-mean Gaussian noise model with preset parameters.

Loop closure factor: We check the availability of a loop closure when a new node is introduced to the mapping graph, before optimizing the graph or rebuilding the local map. First, we identify a single candidate node which is the closest to the one currently being added to the map. We exclude the 30 most recently mapped nodes from this search, and in case the closest node is further away than 5.0 m, loop closure is skipped within the ongoing update. In case a candidate has been found, its corresponding scan is registered to the current local map to compute a corrected pose. Finally, a new constraint to the map graph is added also modelled as a Gaussian distribution with a mean in SE(3), which is computed as the transform between the registration result of the candidate node and the current scan pose. In order to keep the computations in real time, we restrict the loop closure evaluation to maximum one time per 4 s.

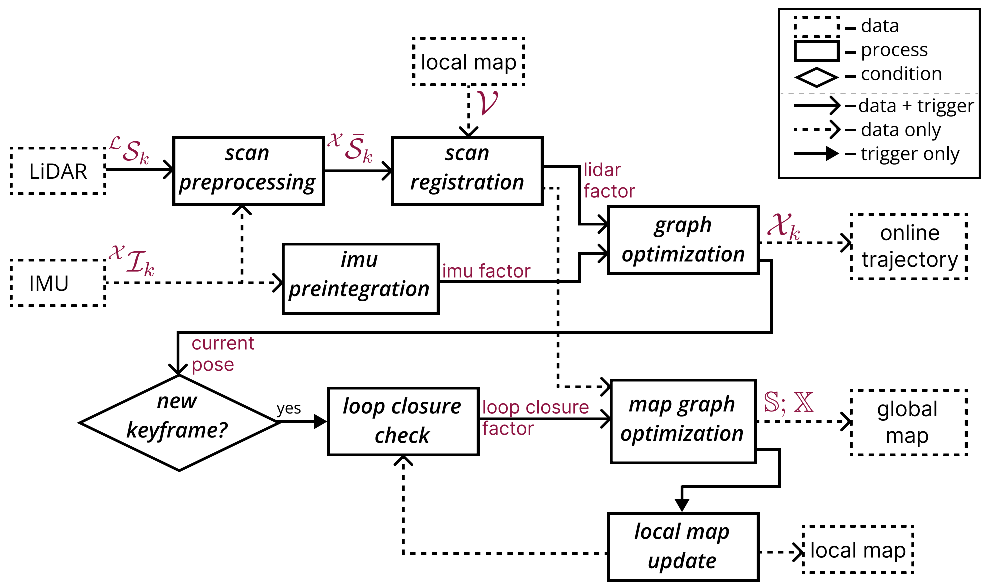

Figure 4. Block diagram of the proposed SLAM solution [44].

3.1.3. Analysis of Positioning Estimation

In this section, we present our evaluation method of the positioning accuracy. During our evaluation, we differentiate between the *online* and *final* trajectories resulting from the SLAM processing. The *online* one captures the ongoing best positioning estimate as it would happen onboard the machine in real time. Since the SLAM is updated after each incoming scan, its internal state, including its estimation of the trajectory, is different after each update. The *final* trajectory encapsulates the state of the SLAM after the last update within a test strip and represents the optimization of the entire trajectory.

To compare the estimated sensor trajectory represented in a local map frame to the reference cabin trajectory in a global frame, we first estimate the relative transformation between the sensor and the cabin frames in each scan in order to obtain a direct estimation of the cabin trajectory. This transformation is estimated by registering a pre-built 3D point-based model of the cabin to each scan using the NDT algorithm. Additionally, we apply an Extended Kalman Filter (EKF) to avoid inconsistent registration results due to visibility issues of the cabin in some scans. In the EKF, we use a prediction model assuming no motion with high noise values, and a measurement model constructed from the registration result the same way as of the lidar factor described previously. The initial guess for the registration was also fed by the EKF's prediction, while the very first initial guess was given manually for each measurement.

After obtaining the estimated cabin trajectory, we map it to the same coordinate frame as the reference measurements to compute the error between them. To obtain this map ${}^{\mathrm{g}}\mathbf{T}_{\mathrm{m}} \in \mathrm{SE}(3)$, we first resample the cabin trajectory to the same timestamps as the reference one by interpolating on manifold between the original SE(3) trajectory elements weighted by their respective timestamps (similarly as in Equation (2)). Then, we minimize the sum of the distances between the time-corresponding transformed and reference trajectory elements, described as:

$$ {}^{\mathrm{g}}\mathbf{T}_{\mathrm{m}}^{*} = \underset{{}^{\mathrm{g}}\mathbf{T}_{\mathrm{m}}}{\operatorname{argmin}} \sum_{k} \left\| \mathbf{p}_{\mathrm{ref},k} - \left({}^{\mathrm{g}}\mathbf{T}_{\mathrm{m}} \, {}^{\mathrm{m}}\mathbf{T}_{l_k} \, {}^{l_k}\mathbf{T}_{\mathrm{c}} \, \mathbf{1}^{T} \right) \right\| , \tag{4} $$

where $\mathbf{p}_{\mathrm{ref},k} \in \mathbb{R}^4$ is the k-th reference cabin position in homogeneous form, ${}^{\mathrm{m}}\mathbf{T}_{l_k} \in \mathrm{SE3}$ is the k-th lidar pose in local map frame, while ${}^{l_k}\mathbf{T}_{\mathrm{c}} \in \mathrm{SE3}$ is the corresponding cabin pose in lidar sensor frame. The vector $\mathbf{1}^T \in \mathbb{R}^4$ is a column vector containing ones that helps us obtain only the translational part from the estimated pose (in homogeneous form), as we only have reference positions available and no orientations.

We solve Equation (4) with two different ranges of k elements to evaluate our positioning. In the case *initial fix*, we only compute the error for elements from the beginning of the measurement until the first element that is further away than 10 m from the starting point. This imitates an absolute evaluation approach, where we assume our system has an ideal global fix during initialization and see how much it drifts as it progresses. In the other case, named *optimized fix*, we evaluate Equation (4) for the whole trajectory, computing an ideal global fix, to understand how much error can be accounted to trajectory errors within the tree measurements. Furthermore, it is important to note that we only evaluate the solution map ${}^{\mathrm{g}}\mathbf{T}_{\mathrm{m}}$ using the *final* trajectory, and we use that to transform both the *online* and *final* trajectories, as it has better quality and they are both represented in the same local map frame.

Once we obtain the resampled and re-mapped estimated trajectory as detailed above, it is then comparable to the reference one. We compute the translational root-mean-square errors along each axis between corresponding reference ($p_{j,\mathrm{ref},k}$) and estimated positions ($p_{j,k}$) on the trajectory, evaluated as

$$ \mathrm{RMSE}_j = \sqrt{\sum_{k=1}^{N_{\mathrm{traj}}} \frac{(p_{j,k} - p_{j,\mathrm{ref},k})^2}{N_{\mathrm{traj}}}} , \tag{5} $$

where $j \in \{x, y, z\}$ represents the axis along which we define the error, k is the element index, and N_{traj} is the number of trajectory elements in the given measurement. Furthermore, we compute a combined *norm* error by computing the distance using the Euclidean-norm $\|.\|$ between the reference and estimated positions and compute its root-mean-square error, defined as

$$ \mathrm{RMSE}_{norm} = \sqrt{\sum_{k=1}^{N_{\mathrm{traj}}} \frac{\|\mathbf{p}_k - \mathbf{p}_{\mathrm{ref},k}\|^2}{N_{\mathrm{traj}}}} . \tag{6} $$

3.2. Algorithms for Tree Stem Measurements from the SLAM-Corrected Point Cloud

We use the SLAM-corrected point clouds obtained as a result of Section 3.1 to conduct automatic measurements of trees near the harvester. In this work, we focus on the detection of tree stems and the estimation of DBH and stem curve. As explained in the following subsections, we use and extend the algorithms developed in [8,9] that have previously provided state-of-the-art accuracy for automatic measurements of stem diameter (RMSE = 3–10%) and stem volume (RMSE = 10–15%) of individual trees from MLS data collected in boreal forests with multiple different mobile platforms [38]. The key idea of the algorithms is to analyze the data in temporal order to detect arc-shaped point clusters that can be used to estimate the stem diameter accurately even in the presence of decimeter-level drifts in the SLAM-corrected trajectory. Importantly, we extend the algorithms developed in [9] to mitigate the bias of the stem diameter estimates caused by the finite laser beam width using a linear model that relates the bias to the measurement distance.

We perform the stem attribute estimation using two parameter modes designed for different applications: the first parameter mode, named as *tree map*, is designed to observe tree stems with a high detection rate while providing stem attribute estimates, such as DBH, with a reasonable accuracy (RMSE = 3–4 cm). This parameter mode is closely related to the concept of thinning density assistant by Ponsse [2] with the difference that we derive the stem attributes with a reasonable accuracy in addition to the tree locations. The second parameter mode, named as *accurate attributes*, is designed for accurate stem diameter measurements (RMSE $=$ 2 cm) at the cost of a lower detection rate. This scenario is more forward-looking as we expect that such accurate estimates combined with other observations, such as tree species, may enable a more thorough optimization of the harvesting process. The goal of the optimization may be, for example, to preserve valuable habitat or to leave sufficient amount of wood in the forest based on the remaining basal area while felling the economically most valuable trees. Such accurate estimates may also serve as training data for large-scale lidar-based forest inventories at the level of individual trees. See also Section 5.3 for further discussion on the applications related to each parameter mode.

Note also that we perform the stem attribute estimation using the final result of the SLAM computation. Real-time estimation and update of stem attributes is left as a subject of future work, though our methods analyze the data in temporal order and may therefore be extended to real-time analysis, as discussed in Section 5.

3.2.1. Digital Terrain Model

Before the generation of the digital terrain model (DTM), we automatically remove potential noise points from the SLAM-corrected point cloud by dividing the data into cubic voxels with sides of 0.5 m and filtering out points from voxels with less than 3 points. Subsequently, we determine the DTM using a voxel-based approach based on [8]. Here, we divide the point cloud into voxels with a side length of 1.5 m in the z direction and a side length of 1.0 m in the x and y directions. For each pixel in the xy plane, the ground elevation is estimated by first finding the lowest voxel with a point count exceeding 0.5% of the overall point count within the pixel and then computing the mean z coordinate of points within the found voxel. Finally, the DTM is smoothed with a Gaussian kernel. After computing the DTM, the z coordinate of each point is normalized by subtracting the ground elevation.

3.2.2. Detection of Tree Stems

The detection of tree stems is achieved through two main steps. In the first step, we go through the point cloud data in temporal order to search for arc-shaped structures potentially corresponding to tree stems. In the second step, the detected stem arcs are grouped into trees using a clustering approach. For both of the steps, we utilize the algorithms proposed by Hyyppä et al. [9] since the algorithms are robust against drifts of the SLAM-corrected trajectory. The parameter values of the algorithms have been chosen based on prior studies, heuristics, and a test of a dozen different parameter value combinations on the test strip 1. Below, we provide the numerical parameter values without parentheses if the same value was used for both studied parameter modes. In the case of different values for *tree map* and *accurate attributes*, the value for *tree map* is reported first as a normal text and then the value for *accurate attributes* (in parentheses).

To provide more details of the stem-arc detection step, we first divide the points into disjoint sets based on their time stamps and the z coordinates using time intervals of 2.0 s (0.8 s) and height intervals of 0.3 m between $z = 0.5$ m and $z = 7.5$ m. For each disjoint set of points, we subsequently detect connected point groups potentially corresponding to stem arcs by applying density-based clustering for applications with noise (DBSCAN) [48].

We use a search radius of 7.5 cm and a minimum point number threshold of four (5) points. To discard DBSCAN clusters with a non-circular shape, we apply circle fitting based on the random sample consensus (RANSAC) framework [49] and retain only such clusters, for which more than 75% of the points are located within 3.5 cm from the circular fit. We further filter out noise points separated by more than 20° (15°) from their neighboring point using an arc division algorithm [9]. For the accepted clusters, we evaluate the goodness of the circular fit based on heuristically chosen quality criteria, including a minimum radius of 5.0 cm, maximum radius of 50.0 cm, minimum point count threshold of 14 (20), maximum allowed standard deviation of radial residuals of 1.75 cm (1.3 cm), and a minimum central angle of 108°. The clusters fulfilling the quality criteria are regarded as stem arcs and saved for further processing.

Having detected point groups qualifying as stem arcs, we apply DBSCAN clustering for the xy center coordinates of the arcs to detect tree stems. To this end, we use a point number threshold of 3 and a DBSCAN search radius of 30 cm corresponding to the typical DBH of trees. As a result, a tree stem must be associated with at least three arcs to be detected. Furthermore, we require that the stem arcs of each tree must span a height range exceeding 1.0 m in the z direction. Subsequently, each of the accepted DBSCAN clusters is regarded as a tree stem, the growth direction of which is further estimated with principal component analysis according to Hyyppä et al. [8]. This enables us to re-fit a circle to each of the stem arcs in a plane perpendicular to the growth direction, which corrects diameter estimation errors caused by the inclination of a tree.

3.2.3. Calibration and Correction of Stem Diameter Bias

In this section, we describe our approach for correcting the bias of stem diameter estimates caused by the large beam width of the Ouster laser scanner schematically illustrated in Figure 5a. This bias correction step mostly follows our recent work in [50] and represents an extension to the algorithmic workflow presented in [8,9]. In these previous works, the bias was small due to the small beam divergence of the laser scanners used. It is a well-known issue that a large beam width of a laser scanner can result in diameter overestimation for cylindrical structures [51]; see Figure 5b. The beam width increases with distance for a given beam divergence, and hence the diameter bias is high for large distances or for scanner with a large beam divergence. In previous experiments, a nearly linear relation between the bias and distance has been observed [52,53]. Simulations by Forsman et al. [51] demonstrated a near quadratic distance–bias relation. However, the deviation from the linear model is minimal and significantly smaller than the measurement noise for divergences of modern scanners and typical measurement distance ranges (6.1 mrad and 1–20 m, respectively, in this study).

Previously, Ringdahl et al. [52] proposed that the bias of a circular fit can be reduced by moving the points within a stem arc towards the center of the arc according to a beam-width-dependent correction angle. Here, we instead model the diameter bias D_{bias} of an individual stem arc as a linear function of the scanner-to-arc distance as

$$D_{\text{bias}}(d) = a + bd, \tag{7}$$

where d denotes the distance between the scanner and the center of the stem arc, and a and b are parameters to be calibrated. The linear model is motivated by its robustness against over-fitting in the presence of noise points and supported by experimental observations in the previous works and our study.

We calibrate the parameters of the linear model in Equation (7) using data on the test strip 1. For each arc, we compute the distance to the scanner at the time of observing the arc and evaluate the diameter estimation error compared to the reference diameter at the given height. Subsequently, we fit a linear model in Equation (7) to the diameter errors as a function distance to calibrate the parameters.

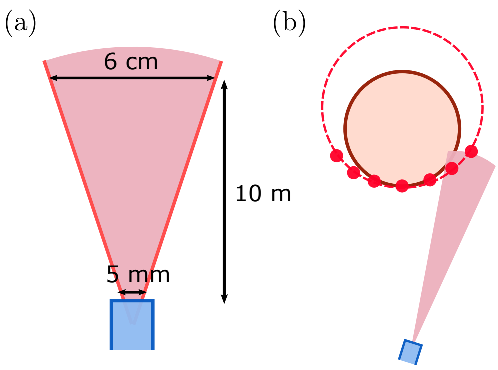

Figure 5. (**a**) Schematic illustration of the distance-dependent laser beam width of the Ouster scanner. At the output, the beam width is approximately 5 mm, and it increases to roughly 6 cm at a distance of 10 m due to a beam divergence of 6.1 mrad [39]. (**b**) Schematic illustration of the diameter overestimation resulting from a finite laser beam width. Here, the red dots represent the assumed locations of tree stem hits resulting from a laser beam with a non-zero width.

As the bias parameters mostly depend on the properties of the scanner, we generalize the obtained parameters for all of the test strips. For each of the detected stem arcs, the diameter estimate is corrected as

$$D_{\mathrm{bias-corr}} = D_{\mathrm{PCA-corr}} - D_{\mathrm{bias}}(d_{\mathrm{arc}}), \tag{8}$$

where $D_{\mathrm{bias-corr}}$ is the bias-corrected diameter, $D_{\mathrm{PCA-corr}}$ is the growth-direction-corrected diameter estimate (see Section 3.2.2), and d_{arc} denotes the scanner-to-arc distance for the given arc. The parameters of the bias model have been calibrated separately for the two parameter modes.

3.2.4. Estimation of Stem Curve and DBH

Following the bias correction, we compute estimates for the DBH and stem curve of the detected tree stems using an approach slightly adapted from Hyyppä et al. [8]. To estimate the stem diameter at various heights for each stem, we compute the median of the bias-corrected diameter estimates within each height interval containing at least one stem arc. We use the median instead of the arc matching algorithm proposed in [8] since the arc matching algorithm cannot account for the bias caused by the large beam width. Subsequently, clearly outlying diameter estimates are filtered out using an automatic outlier filtering step presented in [8]. For each of the tree stems, the final estimate for the stem curve is obtained by fitting a cubic smoothing spline to the remaining diameter estimates.

Using the estimated stem curves, we then evaluate the DBH for each of the detected stems either by interpolating the smoothing spline fit at the height of $z = 1.3$ m or by extrapolating the smoothing spline fit to the height of $z = 1.3$ m if the lowest diameter estimate at z_{min} is above $z = 1.3$ m. If the stem curve has been successfully estimated across a height range exceeding 3 m, we evaluate the smoothing spline at 100 points between z_{min} and $z_{\mathrm{min}} + 3$ m to fit a linear model that is used to extrapolate the DBH at $z = 1.3$ m. If the height range for stem curve estimation is less than 3 m, we prefer to fit a square root model $D(z) = D(0)\sqrt{1 - z/h}$ to the diameter estimates constituting the stem curve to extrapolate the DBH at the height of $z = 1.3$ m in a more robust manner. In the square root model, $D(0)$ is the only free parameter, whereas h denotes an estimate for the tree height.

In this study, we do not analyze in detail the estimation of height and stem volume of individual trees since the used Ouster scanner does not properly capture the canopy layer due to a limited field of view. In Section 5.3, we briefly discuss the results for the estimation of tree height and stem volume based on methods from [9].

3.2.5. Statistical Analysis of Stem Detection and Stem Attribute Estimation

In this section, we present the methods used for the statistical analysis of tree detection and stem attribute estimation. For the statistical analysis, we first determine matching pairs between the detected and reference stems. Since the locations of the reference trees are in a local coordinate frame, we apply the 2d coarse registration algorithm presented in [54] that fits a Euclidean transformation between the coordinate frames of the detected and reference stems using the stem locations as input. After applying the Euclidean transformation to the reference stem locations, we refine the registration by further fitting a 2D affine transformation $\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 + \mathbf{t}$ using the matching tree pairs obtained in the previous step, for which the root-mean-square error of the estimated stem curve is less than 4 cm. After applying the affine transformation, the matching tree pairs are determined by finding the closest detected stem for each of the reference stems if any within a distance of 0.75 m.

Using the matching tree pairs, we evaluate the completeness (i.e., recall) and correctness (i.e., precision) for the stem detection as

$$\text{Completeness} = \frac{N_{\text{matched}}}{N_{\text{ref}}}, \tag{9}$$

$$\text{Correctness} = \frac{N_{\text{matched}}}{N_{\text{detected}}}, \tag{10}$$

where N_{matched} denotes the number of detected stems with a matching reference pair within the region of interest, N_{ref} is the number of all reference stems within the corresponding region, and N_{detected} is the number of all detected stems within the corresponding region. We also study the completeness and correctness as a function of the distance from the harvester trajectory using disjoint regions between $d_{i,\text{start}} = (i-1) \times 3$ m and $d_{i,\text{end}} = i \times 3$ m from the trajectory, with $i \in \{1, 2, \ldots, 7\}$. To mitigate small positioning errors between the detected and reference stems, the location of each reference stem with a matching detected tree is set to the location of the matching detected tree. Furthermore, we analyze the completeness of stem detection as a function of the tree size by dividing the reference trees into four disjoint categories based on their DBH: DBH $\in [0, 20)$ cm, DBH $\in [20, 28)$ cm, DBH $\in [28, 36)$ cm, and DBH $\in [36, \infty)$ cm.

To evaluate the accuracy of the estimated stem attributes, such as DBH, we compute the bias, root-mean-square error (RMSE), and median absolute error (MAE)

$$\text{bias} = \sum_{i=1}^{N_{\text{matched}}} \frac{x_i - x_{\text{ref},i}}{N_{\text{matched}}}, \tag{11}$$

$$\text{RMSE} = \sqrt{\sum_{i=1}^{N_{\text{matched}}} \frac{(x_i - x_{\text{ref},i})^2}{N_{\text{matched}}}}, \tag{12}$$

$$\text{MAE} = \text{median}(\{|x_i - x_{\text{ref},i}|\}_{i=1}^{N_{\text{matched}}}), \tag{13}$$

where x_i denotes the estimated value of the stem attribute for the ith detected stem, and $x_{\text{ref},i}$ is the corresponding reference value. The relative bias, RMSE, and MAE are obtained by normalizing the absolute measure with the mean reference estimate. We report MAE since it provides an error metric robust to outliers. Thus, the comparison of MAE and RMSE may reveal whether the RMSE error is dominated by a few outliers. Normally distributed errors satisfy RMSE $= 1.4826 \times$ MAE, whereas outliers lead to an inequality RMSE $\gg 1.4826 \times$ MAE.

To assess the accuracy of the estimated stem curves, we modify Equations (11)–(13) to account for the fact that the comparison against the reference stem curve occurs at multiple heights

$$\text{bias} = \frac{1}{N_{\text{matched}}} \sum_{i=1}^{N_{\text{matched}}} \sum_{j=1}^{N_i} \frac{D_i(z_j) - D_{\text{ref},i}(z_j)}{N_i}, \tag{14}$$

$$\text{RMSE} = \sqrt{\frac{1}{N_{\text{matched}}} \sum_{i=1}^{N_{\text{matched}}} \sum_{j=1}^{N_i} \frac{(D_i(z_j) - D_{\text{ref},i}(z_j))^2}{N_i}}, \tag{15}$$

$$\text{MAE} = \text{median}(\{\text{median}(\{|D_i(z_j) - D_{\text{ref},i}(z_j)|\}_{j=1}^{N_i})\}_{i=1}^{N_{\text{matched}}}), \tag{16}$$

where $D_i(z_j)$ denotes the estimated stem curve of the detected tree i at the height of z_j, $D_{\text{ref},i}(z_j)$ is the corresponding reference value, and N_i denotes the number of heights at which the estimated stem curve can be compared against reference diameter values, typically spaced every 20 cm along the vertical direction. Hence, the value of N_i may vary from tree to tree depending on the height range of stem curve estimates.

For all of the above statistical metrics, we compute both overall results considering all of the three strips, as well as individual results obtained for each of the test strips separately. When computing the overall results, it is possible that a small number of trees between two neighboring test strips are detected on both test strips, and hence, a small number of trees may contribute twice to the overall results.

4. Results

4.1. Results for Harvester Localization Using SLAM

First, we present the results related to our SLAM solution, focusing on the positioning capabilities of our system. As discussed in Section 3.1.3, we study two positioning modes with two different goals. On one hand, we present the *online* trajectory using the *initial fix* alignment, which represents expected onboard performance with a fix that could be obtained before entering the forest. On the other hand, we show the *final* trajectory using the *optimized fix* alignment that showcases our best achievable result obtained after finalizing the harvester mission. Importantly, this final trajectory simply shows the state after the last update of the SLAM, and it is still computed in real time in the local map frame. The only required post-processing step is to compute the optimized fix that allows us to transform the obtained *final* trajectory to the global frame using the reference trajectory.

In Figure 6, we show a top view of the trajectories on the studied test strips for both positioning modes and for the reference measurements. We provide a more quantitative presentation in Figure 7, including the positioning error along each axis and the norm of the error as a function of time. In case of the *online* trajectories, the drift gradually increases until the harvester starts driving backwards. Meanwhile, for the *final* trajectories using the *optimized fix*, the errors are low and at most few decimeters throughout the entire experiment.

We summarize our results for the positioning accuracy in Table 3, including also results for the two other combinations, *online* trajectory with *optimized fix* and *final* trajectory with the *initial fix* alignment. With the *online* trajectory, the overall error on the test strips was on the scale of a few metres regardless of the alignment method. For the *online* trajectory with the *initial fix* alignment, the total positioning error was 2.44 m when averaged across the test strips. Importantly, this error represents the offset of the position throughout the operation, but the map is approximately locally consistent at each point in time. The internal map quality is better represented by the results for the *final* trajectory with the *optimized fix* that was as low as 0.21 m when averaged across the test strips. Furthermore, the errors obtained on each strip are proportional to the length of the trajectory, as is often the case with SLAM-based systems. The norm of the translational error ranges from 0.07 m on the shortest test (strip 2) with a total traveled distance of 270 m, up to 0.32 m on the longest test (strip 5) with a traveled distance of 447 m.

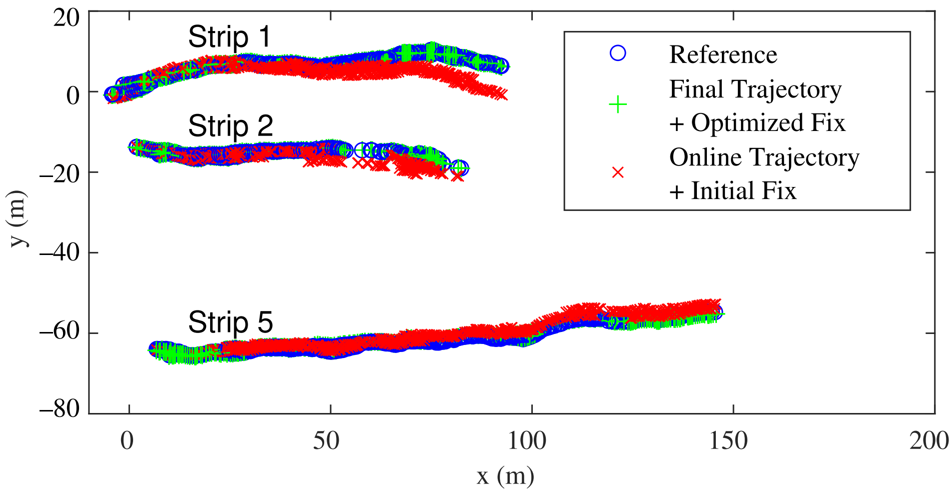

Figure 6. Top view of the trajectories on the three test strips, including the reference trajectories (blue), the final trajectories using the optimized fix (green), and the online trajectories using the initial fix (red).

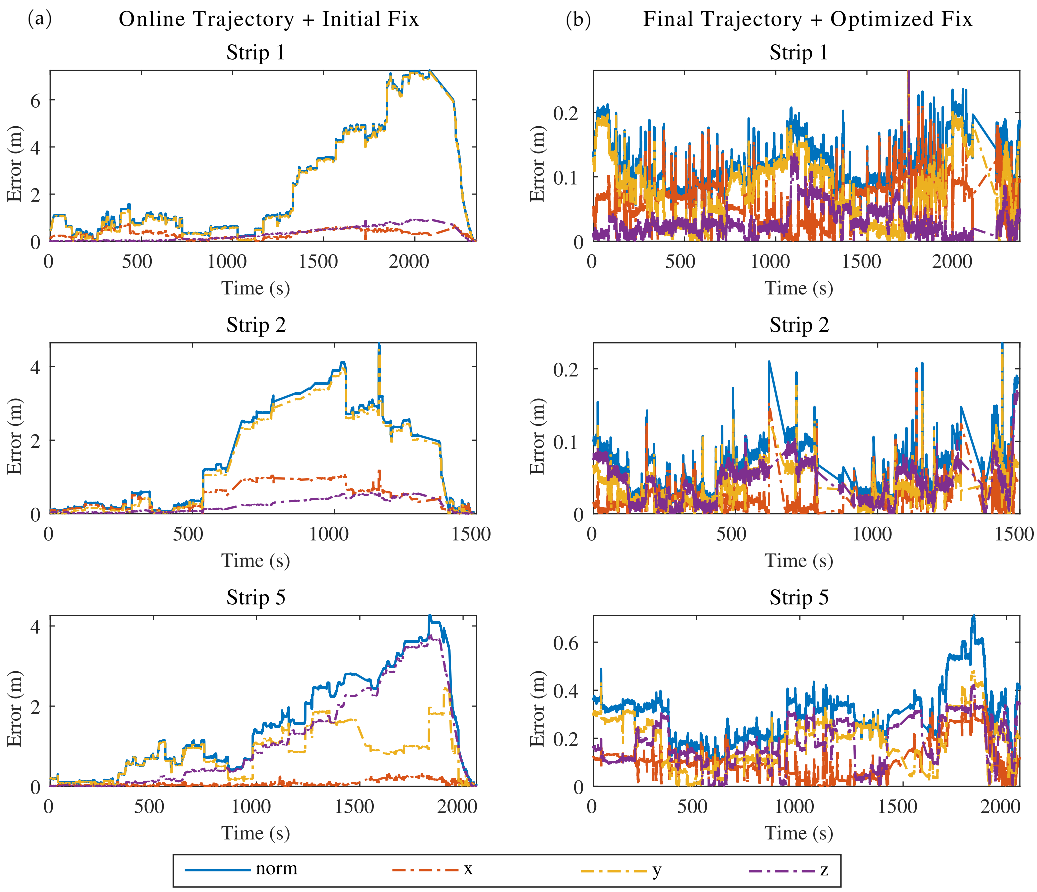

Figure 7. Translational positioning errors along each axis and the norm of the error as a function of time. (**a**) Errors related to the *online* trajectory using the *initial fix* and (**b**) the *final* trajectory using *optimized fix*. Subplots for both cases, going from top to down corresponds to the test strips 1, 2, and 5 accordingly. Note that the *x*-axis is approximately the direction of travel in the tests as shown in Figure 6.

Table 3. Summary of the positional errors for both the *final* and *online* trajectories, evaluated for each test strip and also for all the strips combined, where the trajectories are concatenated first and the errors are computed afterwards. Furthermore, the results are presented for both *initial fix* and *optimized fix*, where the first one imitates an optimized initial global fix, while the latter represents an ideal alignment between the estimated and reference trajectories. For the two main positioning modes *online* trajectory with *initial fix* and *final* trajectory with *optimized fix*, the norm of the positioning error across all strips is highlighted in bold.

	Positioning RMSE (m)							
	***Final* Trajectory**				***Online* Trajectory**			
	x	**y**	**z**	**norm**	**x**	**y**	**z**	**norm**
Initial Fix								
Test strip 1	0.135	0.405	0.358	0.557	0.360	3.023	0.400	3.070
Test strip 2	0.019	0.150	0.194	0.246	0.546	1.820	0.276	1.920
Test strip 5	0.203	0.871	1.547	1.787	0.100	0.974	1.592	1.869
All strips	0.150	0.588	0.961	1.136	0.353	2.206	0.998	**2.447**
Optimized Fix								
Test strip 1	0.074	0.094	0.038	0.126	0.354	2.661	0.078	2.686
Test strip 2	0.025	0.043	0.051	0.071	0.534	1.688	0.073	1.772
Test strip 5	0.131	0.214	0.200	0.321	0.116	1.751	0.304	1.781
All strips	0.093	0.143	0.125	**0.212**	0.348	2.162	0.193	2.199

In addition to the positioning accuracy, we provide the empirical distribution of the measured update times taken by the SLAM processing in Figure 8. We can see that the majority of the updates were performed between 30 ms and 50 ms, which is safely inside our target of 100 ms corresponding to the time to obtain a single scan (see Table 1). This quick update time suggests that the system is capable of real-time operation, and there is even remaining computational capacity within each of these 100 ms cycles to perform, e.g., onboard tree measurements in the future.

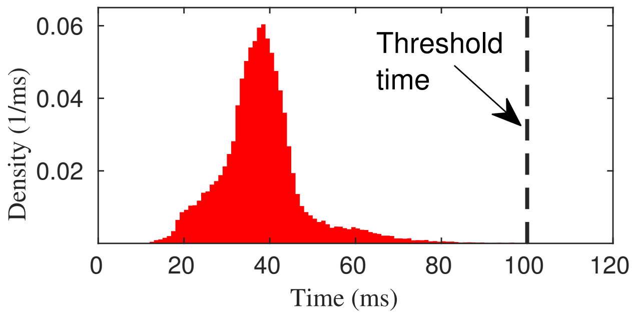

Figure 8. Empirical distribution of the time taken to perform all SLAM computations for an incoming scan. The vertical dashed line shows the threshold for real-time operation corresponding to the time to acquire a single scan with the Ouster scanner.

4.2. Results for Stem Detection and Stem Attribute Estimation

In this section, we present the results for automatic stem detection and stem attribute estimation from the SLAM-corrected point cloud data using the algorithms presented in Section 3.2. As discussed in Section 3.2, we provide the results for two different parameter modes, *accurate attributes* and *tree map*, the first of which aims to accurately estimate stem attributes at the cost of detecting a lower number of trees, whereas the latter is designed to detect the majority of the trees similar to Ponsse's thinning density assistant while allowing higher errors for the stem attributes.

The geometry of the tree stems is accurately modelled by the SLAM-corrected point cloud data as illustrated in Figure 9a,b, which is a prerequisite for obtaining accurate stem attribute estimates. However, the large beam width of the Ouster scanner leads to increased noise in the stem cross sections as visualized in Figure 9a. The upper forest canopy is not captured in the point clouds due to the limited field of view of the scanner used, as shown in Figure 9c, which prevents tree height measurements and limits the accuracy of stem volume estimation. As a result, we mostly focus on tree detection, DBH estimation, and stem curve estimation in the rest of this section.

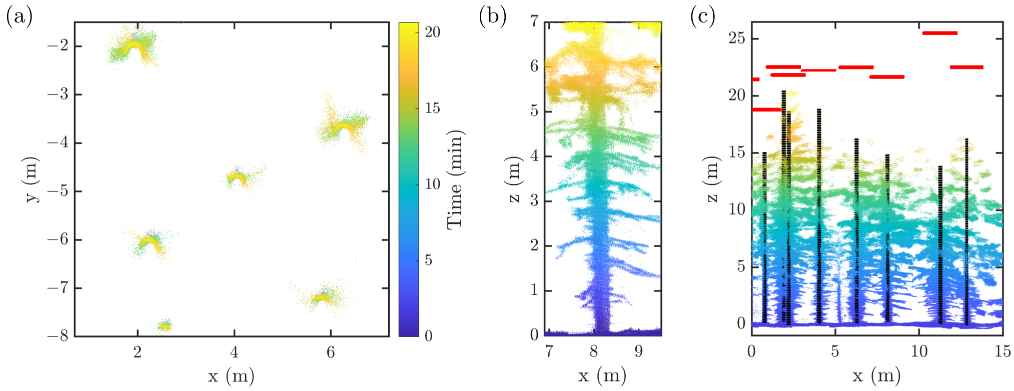

Figure 9. (**a**) Top view of the SLAM-corrected point cloud in the height interval $z \in [1.0, 1.5]$ m showing cross sections for a few tree stems. The points have been coloured based on the acquisition time of the data. (**b**) Side view of an example tree stem in the SLAM-corrected point cloud obtained on the test strip 1 with points coloured based on their z-coordinate. (**c**) Side view of the SLAM-corrected point cloud on the test strip 1 showing heights estimated for the detected trees (black cylinders extending up to the estimated tree height) together with the corresponding reference height values marked by red lines. Points are coloured based on their z-coordinate.

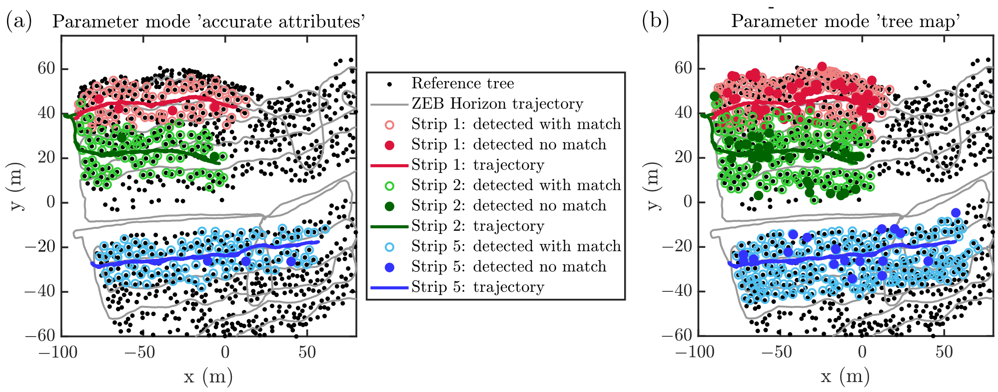

Figure 10. (**a**) A map of tree stems detected using the parameter mode *accurate attributes* from the SLAM-corrected point cloud data for the three harvester test strips together with reference tree locations (black dots) obtained using the ZEB Horizon system along the trajectory shown with the gray line. Detected trees with (without) a matching reference tree are denoted with light-coloured unfilled circles (dark-coloured filled circles). Different colours (red, green, blue) are used to distinguish between the three test strips. (**b**) Same as (**a**) but for the trees detected with the parameter mode *tree map*.

In Figure 10a,b, we illustrate maps of detected trees obtained with the two parameter modes. The figures show that the parameter modes *accurate attributes* and *tree map* enable tree detection up to a distance of roughly 10 m and 15 m from the scanner, respectively. To obtain more quantitative estimates of the tree detection rate, we show the mean completeness rate across all test strips as a function of distance from the trajectory in Figure 11a,b and the completeness rate averaged across a 15 m range from the trajectory in Figure 11c,d. In addition to the overall completeness rate, the figures also present the completeness rate for different DBH categories illustrating the effect of tree size on the detection rate. For the parameter mode *accurate attributes*, the completeness rate drastically increases with increasing DBH as shown in Figure 11a,c. When considering trees within a range of 15 m from the trajectory, the completeness rate equals only 9.6% for trees with DBH $\in [0, 20)$ cm, but it reaches 62.5% for trees with DBH > 36 cm. From Figure 11a, it is also clear that the tree detection rate sharply drops for distances above 10 m, and practically no trees are detected above a distance of 15 m.

For the parameter mode *tree map*, Figure 11b,d show that the average completeness rate within a 15-m-range from the trajectory is roughly constant at 90% for the DBH categories with DBH ≥ 20 cm, whereas the corresponding average completeness rate for trees with DBH < 20 cm is only 49%. Furthermore, Figure 11b illustrates that the completeness rate stays roughly constant up to a distance of 15 m, above which the completeness sharply reduces to zero.

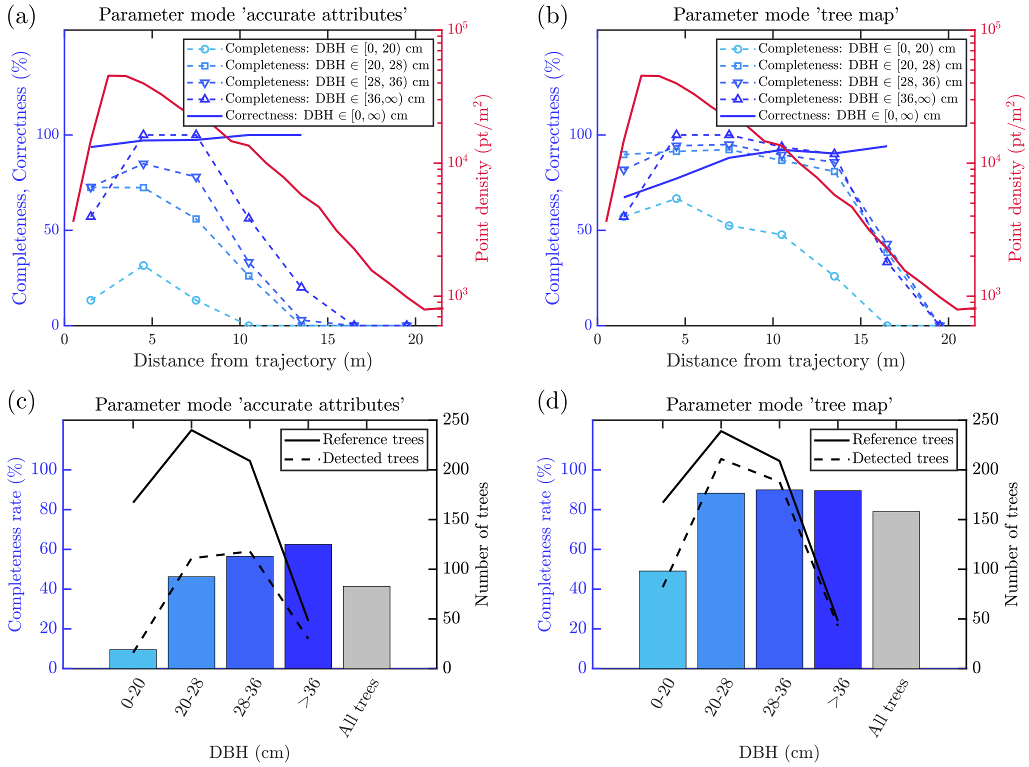

Figure 11. (**a**) Completeness (markers) and correctness rate (solid blue line) of stem detection averaged across the test strips as a function of the distance from the trajectory based on the parameter mode *accurate attributes*. The completeness rate is shown separately for four different DBH categories DBH $\in [0, 20)$ cm, DBH $\in [20, 28)$ cm, DBH $\in [28, 36)$ cm, and DBH $\in [36, \infty)$ cm. The point density averaged across the test strips as a function of the distance from the trajectory is shown with the red line and the right y-axis. (**b**) Same as (**a**) but for stems detected using the parameter mode *tree map*. (**c**) Average completeness rate of stem detection within 15 m range from the trajectory based on the parameter mode *accurate attributes* for the same four DBH categories as in panels (**a**,**b**). The overall completeness rate considering trees of all sizes is shown with the gray bar. The dashed and solid black lines show the number of detected trees and reference stems, respectively, within each size category. (**d**) Same as (**c**) but for the parameter mode *tree map*.

For both parameter modes, the completeness rate varies slightly across the different strips, as can be inferred from Table A1 in Appendix A. For both parameter modes, the lowest overall completeness rate was obtained on strip 1, while the highest completeness rate was attained on strip 5. The differences between the strips can be mostly attributed to differences in the tree size distributions since the average DBH was the lowest on strip 1, while strip 5 contained the lowest proportion of small trees (see Table A1). When comparing the completeness rates within each DBH category, there are no clear trends or differences between the different strips.

The average correctness rate of stem detection is 96.8% for the parameter mode *accurate attributes*, whereas it is only 78.0% for the parameter mode *tree map*. Interestingly, the correctness rate related to *tree map* increases as a function of distance from the scanner, reaching values above 90% for distances above 10 m as further discussed in Section 5.2.

Next, we experimentally demonstrate the bias correction procedure explained in Section 3.2.3 to highlight its importance for the current system based on a Ouster scanner with a large beam divergence of 6.1 mrad. Without the bias correction, we observe that the average diameter bias of individual stem arcs increases approximately linearly as a function of distance with a rate of 6.9 cm per 10 m as illustrated in Figure 12a. This estimate for the increase rate of bias approximately agrees with the beam divergence of 6.1 mrad corresponding to a 6.1 cm increase in the beam width at a distance of 10 m.

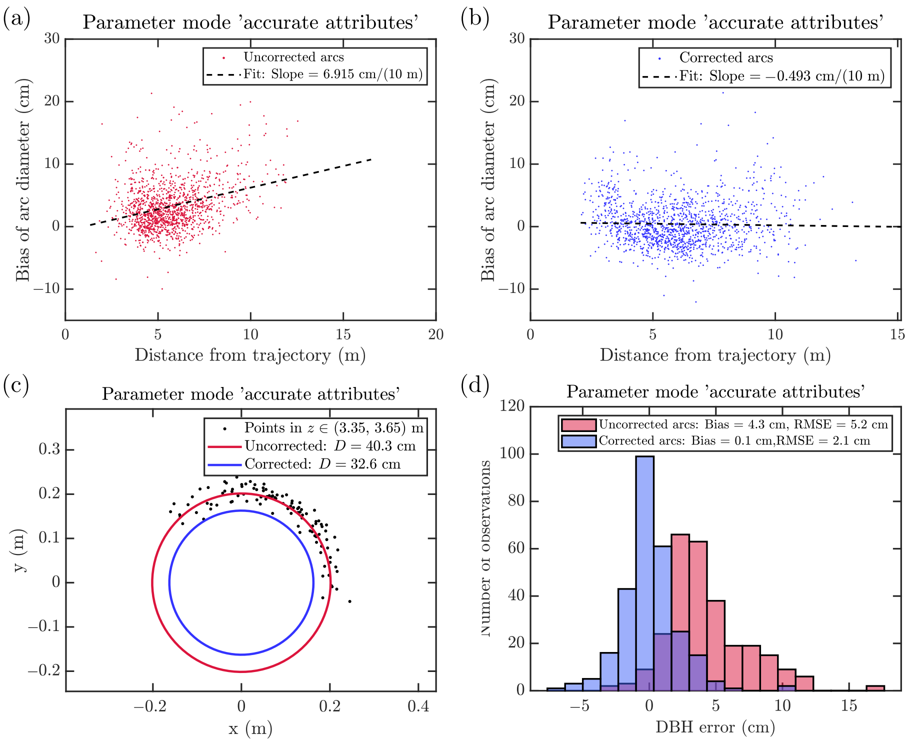

Figure 12. (**a**) Diameter bias of individual stem arcs on strip 1 as a function of the distance between the arc and the scanner before applying the bias correction. The dashed black line shows the linear model based on a least-squares fit that is used to extract the parameters for the bias correction. The number of points has been downsampled by a factor of 10 for illustration purposes. (**b**) Diameter bias of individual stem arcs on strip 2 as a function of the distance between the arc and the scanner after correcting the bias using parameters obtained from the linear fit shown in panel (**a**). Similar to (**a**), the number of points has been downsampled. (**c**) Stem arcs in the height interval $z \in (3.35, 3.65)$ m for an example tree on test strip 1, together with circles representing diameter estimates before (red) and after (blue) correcting for the bias. (**d**) Histogram of DBH errors before (red) and after (blue) applying the bias correction. The statistics include all detected trees with a matching reference tree on test strips 1, 2, and 5. In all of the panels, the parameter mode *accurate attributes* was used for estimating the stem diameters from the point cloud data.

We perform the bias correction explained in Section 3.2.3 by calibrating the parameters of the linear model in Equation (7) using the bias data from strip 1, after which we apply the obtained bias correction model for all of the three strips. As shown in Figure 12b, the distance dependence of the diameter bias is drastically reduced after the bias correction. On test strip 2, we find a residual increase rate of -0.5 cm per 10 m. The effect of the bias correction is further visualized in Figure 12c showing stem arcs of one of the trees together with a circular fit to the data and a circle based on the bias-corrected diameter.

Furthermore, Figure 12d shows the distribution of DBH errors obtained with the parameter mode *accurate attributes* before and after the bias correction. Importantly, the bias correction not only shifts the center of the distribution to approximately 0 cm, but it also reduces the spread of the distribution since the magnitude of diameter overestimation varies from arc to arc before the correction. For the parameter mode *accurate attributes*, we observe that the variance part of the RMSE of DBH (obtained as $\sqrt{\text{RMSE}^2 - \text{bias}^2}$) equals 3.0 cm before the bias correction, while it is reduced to 2.1 cm after the correction. For the parameter mode *tree map*, the corresponding reduction is even larger.

Figure 13a–d show scatter plots for the DBH and stem curve estimates obtained using the parameter modes *accurate attributes* and *tree map*. For the parameter mode *accurate attributes*, the bias, RMSE, and MAE of DBH were 0.1 cm (0.2%), 2.1 cm (7.3%), and 1.0 cm (3.4%), respectively, whereas the corresponding metrics for the stem curve were 0.1 cm (0.5%), 2.3 cm (8.3%), and 1.2 cm (4.3%), respectively. For the parameter mode *tree map*, the RMSE and MAE were higher: the DBH error had a bias of −0.4 cm (1.4%), an RMSE of 3.3 cm (12.1%), and an MAE of 1.5 cm (5.6%), whereas the corresponding metrics for the stem curve were −0.2 cm (−0.6%), 3.7 cm (14.0%), and 1.8 cm (7.0%), respectively. For both parameter modes, as well as for both DBH and stem curve, the ratio between RMSE and MAE exceeds 2.0, which is clearly above the ideal value of 1.48 expected for normally distributed errors due to a few outlying values visible in the scatter plots of Figure 13a–d. As can be seen from Table A2 in Appendix A, the bias, RMSE, and MAE of DBH and stem curve are mostly of similar magnitude across the three strips for a given parameter mode. For the parameter mode *tree map*, slightly higher RMSE and MAE are obtained for test strip 1, which may be caused by the large number of small trees, the diameter of which is difficult to estimate due to the large beam width of the scanner used.

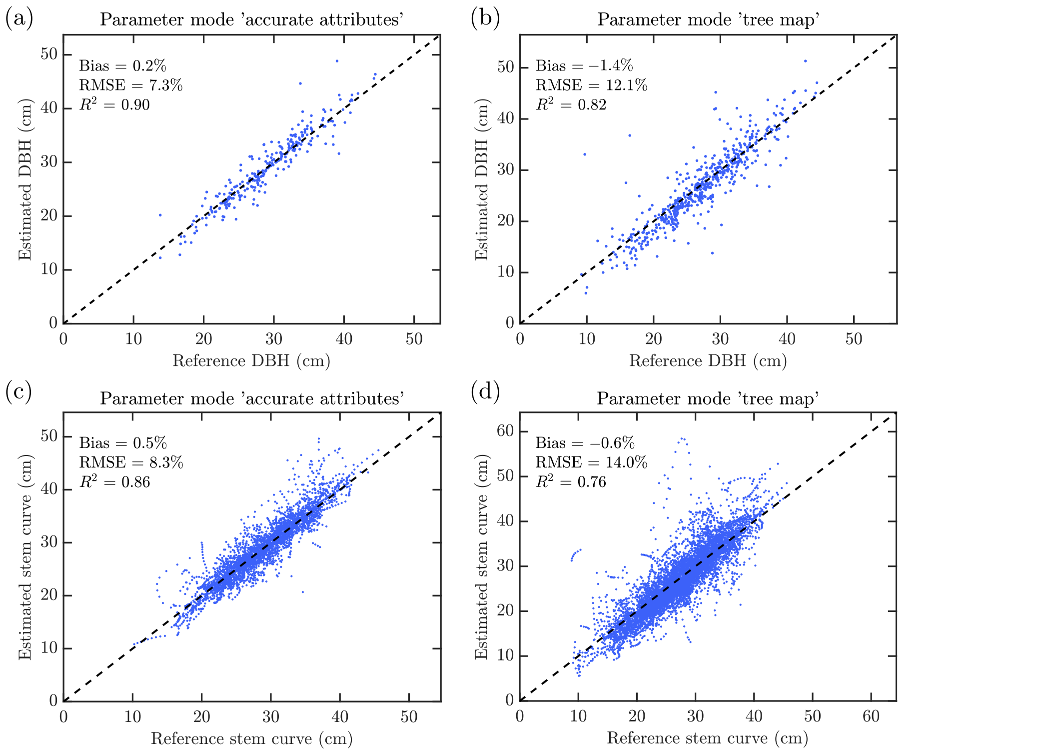

Figure 13. (**a**) Scatter plot of estimated DBH values vs. reference values based on the parameter mode *accurate attributes*. (**b**) Same as (**a**) but for the parameter mode *tree map*. (**c**) Scatter plot of the stem curve estimates vs. reference values based on the parameter mode *accurate attributes*. (**d**) Same as (**c**) but for the parameter mode *tree map*.

In Figure 14a, we illustrate a stem curve obtained for an example tree on test strip 1 using the parameter mode *accurate attributes*. From the figure, we observe that the smoothing spline fit to the diameter estimates agrees well with the corresponding reference measurements, but the tree height is underestimated due to the limited field of view of the scanner. To illustrate the typical height range enabling stem curve estimation, Figure 14b shows the distributions of the lowest and highest heights, for which the stem curve could be estimated for the individual trees. On average, the stem curve could be measured between $z = 0.9$ m and $z = 5.4$ m for the parameter mode *accurate attributes*, whereas the corresponding height range was $z \in (0.9, 6.0)$ m for *tree map*. However, the highest height enabling stem curve estimation varied significantly, ranging between $z = 2$ m and $z = 8$ m, as illustrated in Figure 14b.

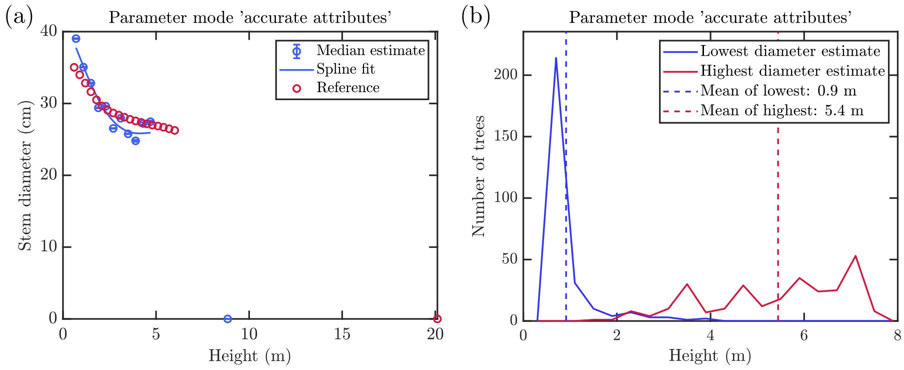

Figure 14. (**a**) Example stem curve from the SLAM-corrected point cloud data as a function of the height from the ground together with the reference stem curve (red circles) based on the ZEB Horizon system. The blue circles show the median diameter of bias-corrected stem arcs at each height, whereas the solid blue line shows the corresponding smoothing spline fit. The stem curve has been estimated for one of the trees on the test strip 1. (**b**) Distribution of lowest (blue) and highest (red) heights, for which the stem curve could be estimated for individual trees on the three test strips. The vertical dashed lines denote the mean values for the lowest and highest height.

Finally, we study how the DBH and stem curve errors depend on the tree size and the tree-to-harvester distance. In Figure 15, we illustrate the bias, RMSE, and MAE of DBH and stem curve estimates for the four studied DBH size categories when using either of the parameter modes *accurate attributes* or *tree map*. From the figure, we first see that the bias of both DBH and stem curve estimates is close to zero for all of the DBH categories regardless of the parameter mode used. When comparing the parameter modes with each other, we observe that the RMSE and MAE within each DBH category are systematically lower for *accurate attributes* as expected. For both of the parameter modes, the absolute RMSE and MAE are typically clearly higher for trees with DBH < 20 cm than for trees with DBH ≥ 20 cm. This may be explained by a lower number of stem hits for the smaller trees and a lower effective signal-to-noise ratio. For the DBH categories with DBH ≥ 20 cm, the differences are relatively small, especially when using the outlier-robust MAE as the error metric.

In Figure 16a,b, we show the bias, MAE, and RMSE of DBH and stem curve estimates as a function of the distance from the trajectory for the two parameter modes. Based on the figures, the bias of DBH and stem curve estimates oscillates around zero and has no clear trend after the bias correction. However, the outlier-robust MAE appears to increase with increasing distance from the trajectory for both DBH and stem curve estimates and for both studied parameter modes. This suggests that the increased effective noise caused by the distance-dependent laser beam width hampers stem diameter estimation even after performing the bias correction. Furthermore, we observe that the MAE and RMSE for the parameter mode *tree map* are systematically higher for a given tree-to-trajectory distance compared to *accurate attributes* indicating that the higher MAE and RMSE values for *tree map* in Table A2 of Appendix A are not only caused by the the higher completeness rate of *tree map* at further distances.

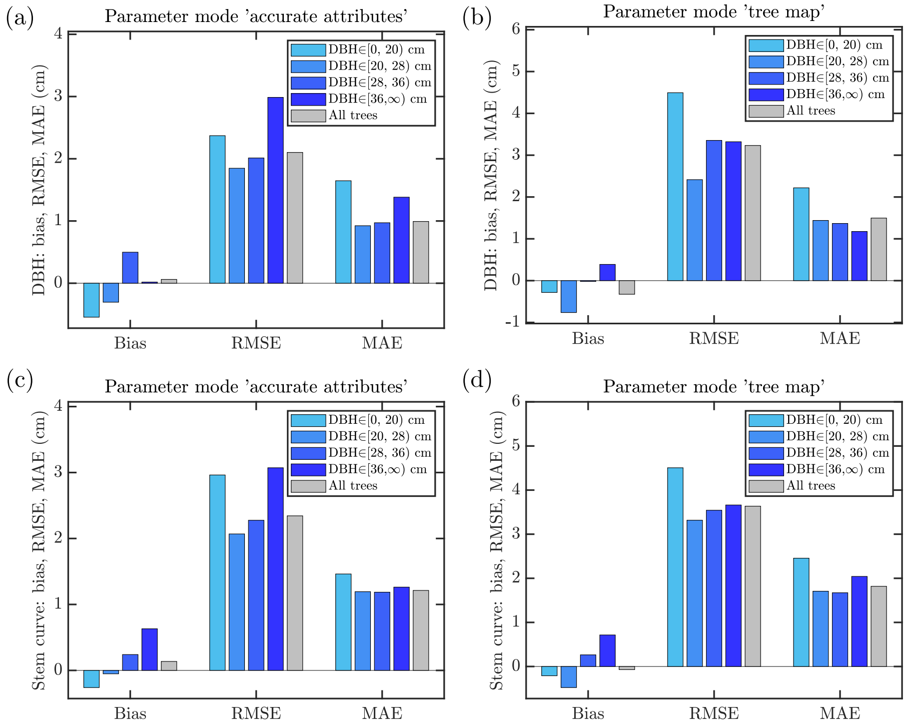

Figure 15. (**a**) Bias, RMSE, and MAE of DBH estimates obtained using the parameter mode *accurate attributes* for trees in four different DBH categories DBH $\in [0, 20)$ cm, DBH $\in [20, 28)$ cm, DBH $\in [28, 36)$ cm, and DBH $\in [36, \infty)$ cm. The overall bias, RMSE, and MAE are shown with a gray bar. The results include all detected tree stems on the three test strips. (**b**) Same as (**a**) but for the DBH estimates obtained using the parameter mode *tree map*. Note the different y axis limits compared to panel (**a**). (**c**) Bias, RMSE, and MAE of stem curve estimates obtained using the parameter mode *accurate attributes* for trees in four different DBH categories DBH $\in [0, 20)$ cm, DBH $\in [20, 28)$ cm, DBH $\in [28, 36)$ cm, and DBH $\in [36, \infty)$ cm. The overall bias, RMSE, and MAE are shown with a gray bar. The results include all detected tree stems on the three test strips. (**d**) Same as (**c**) but for the stem curve estimates obtained using the parameter mode *tree map*. Note the different y axis limits compared to panel (**c**).

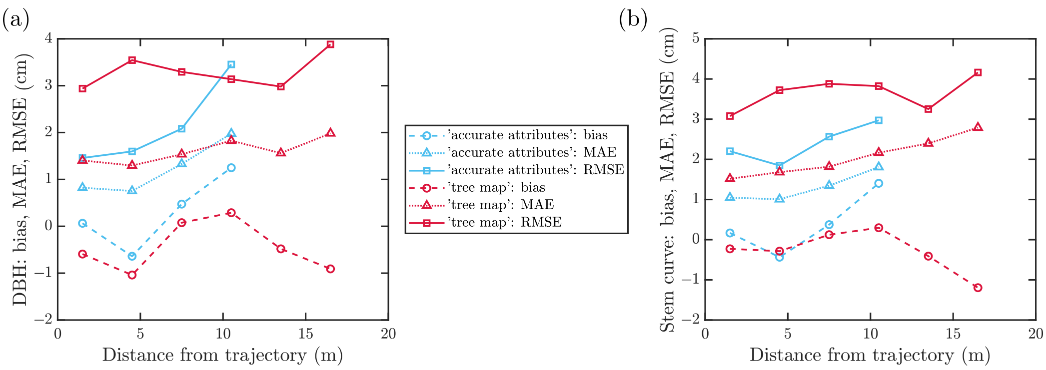

Figure 16. (**a**) Absolute bias (circles), MAE (triangles), and RMSE (squares) for DBH estimates as a function of the tree-to-trajectory distance. The results for the parameter modes *accurate attributes* and *tree map* are shown in blue and red colours, respectively. The results have been aggregated using distance intervals of 3 m and considering distance intervals with at least 5 detected trees. (**b**) Same as (**a**) but for the bias, MAE and RMSE of stem curve estimates.

5. Discussion

5.1. Discussion on Harvester Localization Using SLAM

Related to our SLAM solution, we can categorize the different sources of localization errors into random and systematic errors. Random errors are primarily propagated from the raw measurements and from the additional cabin pose estimation that was necessary to compare our SLAM trajectory to the reference one. A good approximation of the random error can be measured at the beginning of the test strips, where drift has not yet accumulated significantly. At these initial segments (also used to compute the initial fix), we measured an average RMSE of 2.4 cm across the three test strips. Meanwhile, systematic errors can be introduced by a variety of factors. The alignment between the reference and estimated trajectory can highly alter the measured error between them. Further systematic error can be accumulated due to the inaccuracy of the noise models constructed within the observation factors for the graph optimizations. However, we attribute most of the errors to the straight path that the harvester was following. Since we are operating an iterative SLAM method without global measurements, our only mechanism to reduce the drift error is by loop closures. Although conventional loops are not present in the harvester trajectories compared to other MLS systems, our algorithm was still able to recognize previously visited areas and apply corrections. The occasional jumps visible in the *online* trajectories were also an expected consequence of such corrections. In our tests, most loop detections occurred when the harvester returns to the initial position at the end of the measurements, reducing the online error significantly, thus having much smaller errors also in the *final* state of the trajectory. Although the errors are significantly reduced in the *final* trajectory, we still observed that the error is proportional to the length of the trajectory, suggesting that our system cannot fully compensate for the drift. Additionally, we observed that the largest errors accumulated along the y- and z-coordinates as our trajectories were spanned primarily along the x-direction, providing more constraints and information along the x-axis.

It is difficult to directly compare our results to previous works, such as the one by Nevalainen et al. [7], where they measured an error of 0.2 m per 50 m and 0.5 m per 100 m for the xy-components excluding the error along the z-axis (i.e., axis of gravity). The closest comparison we obtained was by observing the final trajectory using the initial fix alignment as shown in Figure 17. Since the alignment is performed before entering the forest, a large error on longer distances to the start is expected. Here, using the initial fix, we observe errors from 1 to 2 m at the first 100 m of distance depending on the strip. However, when studying the optimized fix (x markers in Figure 17), the error of the trajectory is greatly reduced and is always less than 0.5 m across the first 100 m. However, not all of the observed drifts using the *initial fix* are caused by the errors in the initial alignment. A small part of the drift is also present using the *optimized fix*, which can be noted from the increased mean error (in Table 3) in the longer trajectories.

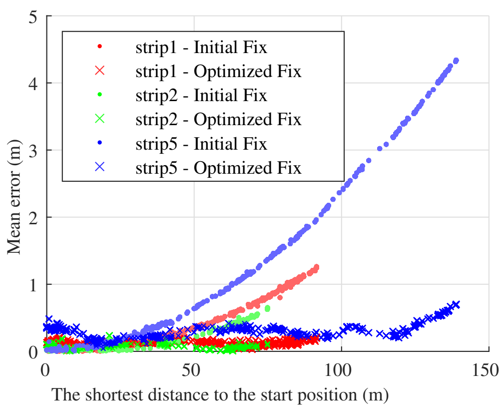

Figure 17. Mean positioning error of the *final* trajectory as a function of the shortest distance to the start position. We included both the *initial* (point marker) and *optimized* (x marker) fix. Furthermore, the test strips are differentiated by colours. Plotted for every tenth trajectory element to improve visibility.

5.2. Discussion on Stem Detection and Stem Attribute Estimation

We observed in Section 4.2 that the completeness rate of tree detection is rapidly reduced as the distance from the scanner increases. We attribute this observation to two main factors: First, the point density rapidly decreases as a function of distance as shown in Figure 11a,b. Beyond a distance of 20 m, the point density is less than 1000 pt/m^2, i.e., approximately 20 times lower than the average point density within a 15 m distance from the trajectory. Second, the large beam divergence of the Ouster scanner causes the effective noise of the stem arcs to increase with distance, which can prevent successful stem arc detection at large distances if the heuristic quality criteria are no longer satisfied.

The parameter mode *tree map* was furthermore observed to result in a non-negligible number of commission errors, which can arise from multiple reasons: First, the lenient parameter thresholds lead to incorrectly identified arcs representing, e.g., points reflected from tree branches, and a sufficient number of such incorrectly identified arcs may further lead to an incorrect detection of a tree stem. The number of incorrectly detected tree stems is higher closer to the scanner due to, e.g., a higher point density, which partly explains the distance dependence of the correctness rate. Second, the reference tree map obtained from the ZEB Horizon point cloud may not contain all the small trees, especially close to the boundaries of the ZEB Horizon point cloud. As a third potential reason, sufficiently large distortions in the point cloud may potentially lead to a single tree being detected multiple times.

5.3. Current Limitations and Potential Future Applications

We foresee that harvester-based MLS systems may have several different applications as illustrated in Figure 18, the most obvious being real-time assistance of the harvester operator based on forest information estimated from the laser scanner data. As mentioned in the introduction, Ponsse has lately introduced a prototype concept for an MLS-based thinning density assistant utilizing the number of detected trees to help the operator to achieve a targeted thinning density. However, the decision-making could be optimized in much greater detail if the harvester MLS system would enable accurate estimation of a variety of stem attributes in line with the studied parameter mode *accurate attributes*. In an ideal scenario, the harvester MLS system and the algorithms processing the data would enable deriving accurate real-time information of, e.g., the geometry and species for most of the surrounding trees while also extracting other information of the ecosystem, such as the detection of valuable habitat, suitable retention trees, and decaying wood. This information could be further converted to instructions on which trees to fell and which to leave growing during a thinning or logging process while ensuring that valuable habitat is saved and sufficient amount of wood is left in the forest to meet green house gas emissions targets.

This application of harvester MLS would not only require real-time positioning of the harvester but also necessitate real-time detection and measurement of surrounding trees. Even though the SLAM-based positioning algorithm used in this work could in principle be executed in real time as explained in Section 4.1, the current Matlab implementation of the stem attribute estimation algorithm does not run in real time but works in a post-processing setting taking the SLAM-corrected point cloud as its input. To understand the bottlenecks of the current implementation, we profiled the performance of the current method using a subset of the point cloud corresponding to a 40 m by 20 m rectangle with 20 million points and a point density of 2.5×10^4 pts/m^2. Executing the Matlab code on a laptop with Intel Core i7-7820HQ 2.9 GHz processor and 32 GB of RAM, the analysis of the studied point cloud took approximately 385 s, including the DTM generation, tree detection, and stem attribute estimation. For a test strip with an assumed length of 150 m and a width of 40 m, this execution time translates to an estimated total analysis time of 1500 s = 48 min, which exceeds the harvester operation time of 25–40 min on a single test strip in our experiments. In the current implementation, the detection of stem arcs dominates the runtime and especially the DBSCAN clustering algorithm takes up to one third of the total execution time. We expect that the runtime may potentially be reduced significantly in the future by using a different programming language, such as C++, and/or by making more extensive use of parallel computing. The point cloud could also be subsampled before the stem detection step to reduce the execution time of the DBSCAN clustering algorithm. Points in the original point cloud nearby the detected clusters may be reintroduced before the circle fitting and the comparison against heuristic quality criteria to increase the number of points used for the stem attribute estimation. However, the completeness rate of tree detection may suffer from a reduced point density in the clustering step. Further improvements to the runtime may potentially be obtained by developing less computationally expensive and more efficient methods for the stem detection, e.g., using deep learning methods enabling rapid GPU execution [55,56].

Potential future applications of harvester MLS

	Reference data collection for ALS surveys	Thinning/felling assistant	Thinning/felling optimizer
Brief explanation of application	Derive attributes of individual trees from harvester MLS data to obtain large amounts of training/reference data for large-scale ALS surveys	Detect trees around the harvester in real-time and use the locations of trees for assisting the operator of the harvester	Detect trees and their attributes accurately around the harvester in real-time to help in optimizing the harvester operation. Prior planning may be combined with the real-time derived information
Requirements for positioning	• Offline • Georeferenced frame • Low positioning errors throughout • *Final trajectory + Optimized fix*	• Real-time • Local frame • Low positioning errors locally, but drifts on long time scales allowed • *Online trajectory + initial fix*	• Real-time • Local or georeferenced frame • Low positioning errors throughout • *Online trajectory + initial fix*
Requirements for tree attribute estimation	• Offline • Locations of individual trees • Sufficiently accurate DBH, height, stem volume, species etc for representative sample of trees • Parameter mode *accurate attributes* (or *tree map*)	• Real-time • Locations of individual trees • Rough estimate of tree size, e.g., from DBH, • Parameter mode *tree map*	• Real-time • Locations of individual trees • Accurate DBH, height, stem volume, species etc • Parameter mode *accurate attributes*

Figure 18. Summary of potential future applications of harvester MLS with requirements for positioning and tree attribute estimation. The graph also presents the connection between the applications and the studied modes of positioning (*online trajectory + initial fix* vs. *final trajectory + optimized fix*) and tree attribute estimation (parameter mode *tree map* vs. *accurate attributes*).

In addition to runtime improvements, the real-time analysis of the point cloud data would require updating the working principle of the stem attribute estimation algorithm to enable continuous update and aggregation of stem attributes as new data are accumulated. Since the stem arc detection algorithm presented in Section 3.2 analyzes the point cloud data one brief time interval at a time, the same arc detection method may be applied in the case of continuously accumulated data. However, some additional logic may be required to continuously improve the assignment of detected stem arcs into trees and to update estimated stem attributes and their uncertainties, e.g., using a Bayesian approach.

The real-time operation may also require improved accuracy for the harvester positioning. According to the results of Section 4.1, the real-time positioning error during the thinning operation may be significantly higher than the positioning error obtained after the optimization of the full trajectory. For some applications, the temporal drift of the trajectory may be tolerated if it is sufficient to only consider a local map of the surroundings accumulated during a limited period of time. More complex applications (e.g., thinning/felling optimizer in Figure 18) may, however, require a higher accuracy for the real-time positioning, e.g., due to a need to match the detected trees with the ones in a prior plan. To overcome this issue, future work is needed to study how to integrate a prior map of the forest to the processing flow and if such a map could also improve the accuracy of real-time positioning. The prior map could be obtained, e.g., using national ALS-based forest inventories.

Another potential application of harvester-based MLS is the collection of vast amounts of up-to-date training data for large-scale individual-tree-based forest inventories conducted using ALS. Here, the idea is to use the high-quality tree attributes, such as stem volume, estimated from the harvester MLS data as teaching data for machine learning or deep learning models predicting the corresponding tree attributes from sparse ALS data (e.g., 5 pt/m^2) covering large regions or even entire countries [57,58]. Compared to conventional field measurements or terrestrial laser scanner (TLS), harvester MLS systems would enable the collection of significantly larger, up-to-date training datasets, potentially enabling a higher accuracy for the trained machine learning models and the use of local models instead of national models, further improving accuracy. However, there are a few key challenges related to this potential application of MLS data: First, the tree attributes detected from the harvester MLS data need to provide a representative sample of the trees to be modelled. Second, the trees detected from harvester MLS data must be correctly matched to trees detected from the sparse ALS data. Thus, the positioning errors within the harvester MLS data need to be sufficiently small, in addition to which the harvester MLS data need to be georeferenced to the global coordinate system, e.g., using the detected tree locations from the harvester MLS data and the ALS data for the final alignment [54,59–61].

When using the harvester MLS data to produce high-quality training data for ALS inventories, it is important to note that the trajectory computation and the stem attribute estimation can be performed offline. However, this would require that the onboard computer of the harvester should have sufficient memory to store the acquired data or the data need to be continuously transferred to the cloud. The offline computation may take more computational resources and utilize the full dataset, potentially leading to higher accuracy for the positioning and stem attribute estimation compared to the real-time analysis. Note that the tree attributes derived for the harvested trees can be used as training data for ALS data collected before the harvesting, whereas the trees remaining in the forest can be used as reference for ALS data collected before or after the harvesting. In an actual implementation, many further details need to be considered, such as the effect of snow on tree attributes, as well as tree growth between the acquisition of datasets.

A further possibility is to utilize HPR data, i.e., stem curve measurements physically conducted by the harvester for the logged trees [16], to improve the stem attribute estimation. For example, the HPR data could be used to optimize the parameters of the stem attribute estimation algorithm or to calibrate parameters needed for the bias correction by comparing MLS-derived stem diameters of logged trees to the measurements in the HPR data. Furthermore, the MLS-derived stem curves of logged trees to be used as training data could be even updated or replaced using the HPR data in order to improve the quality of the training data.

For the above discussed applications to provide added value, the estimated tree attributes should have low enough errors and they need to be obtained for a sufficiently high proportion of the trees. In this work, we studied stem attribute estimation using two parameter modes demonstrating the trade-off between the accuracy of obtained stem attributes and the completeness rate of stem detection. The RMSE of DBH and stem curve estimates ranged between 2.1 cm and 3.6 cm, which is somewhat higher than state-of-the-art error of 1–2 cm obtained in the literature using MLS systems with a low beam divergence, such as ZEB Horizon [6,33,41]. Importantly, the RMSE is also higher than in our previous studies applying a relatively similar stem attribute estimation algorithm for MLS data collected with different systems in comparable boreal forest conditions [9,38], which suggests that the relatively high error obtained in this work is likely caused by the measurement system and not the algorithms used. We attribute the error of stem diameter estimates mainly to the high beam divergence of the used Ouster OS0 scanner and the limited field of view of the measurements. The high beam divergence resulted in increased effective noise in the point cloud and overestimation of the stem diameters when not correcting the distance-dependent bias. The large beam divergence reduced the detection rate of distant trees as their arcs were of low quality due to the beam-width-induced increased noise. As another source of error, the stems were only mapped from one side due to the linear trajectory of the harvester in contrast to typical MLS measurements viewing each stem from multiple directions. This may have hampered tree detection and diameter estimation due to occlusion. The problems related to the limited field of view are exacerbated if the harvester MLS system is used for real-time assistance of the operator since the trees to be cut down are mainly observed in front of the harvester.

Furthermore, the limited vertical field of view of Ouster OS0 scanner prevented us from accurately estimating the tree height and consequently the stem volume. Regardless of the used parameter mode, the bias of tree height estimation was approximately −6 m with the RMSE equaling 7 m. The stem volume estimates obtained from the tree height and stem curve consequently had a large negative bias of approximately −20% with the RMSE equaling 30% for *accurate attributes* and 33% for *tree map*. We expect that the integration of a scanner with a smaller beam divergence and increased field of view would enable stem attribute estimation with significantly reduced errors while also improving the completeness and correctness rates of tree detection even for more distant trees. Such improvements may make it feasible to significantly improve the optimization of the harvesting process. Naturally, there are other properties of the scanner that also need to be considered in a practical implementation of a harvester MLS system, such as the overall cost of procurement and maintenance, durability, reliability, and tolerance to winter conditions.

We also point out that the bias correction procedure described in Section 3.2.3 may be extended further to account for different sources of bias [53]. For example, previous studies have observed that the bias of stem curve estimation tends to increase as a function of height from the ground for different MLS systems [38]. Such height-dependent bias could also be mitigated at least partly by fitting a parametric or non-parametric model to the bias as a function of the height variable. The calibration of the bias correction naturally requires some reference data, but after the calibration the same model can be applied for the same scanner system provided that the model generalizes well and is not specific to, e.g., certain type of test sites.

Finally, we emphasize that our experiments on the harvester MLS system were performed in a coniferous forest that represents a typical managed forest in the boreal forest zone at the time of the first thinning. Our tests were carried out during the winter time, but we expect the system to perform equally well in different seasons. However, further research would be needed to demonstrate the performance of our methods and system in different forest conditions, such as in a temperate forest. In principle, the proposed SLAM solution is not specific to the forest type, but the resulting positioning error may naturally depend on a multitude of factors, such as the density of the forest, as well as the length and shape of the trajectory. On the other hand, our algorithm for stem attribute estimation is primarily designed for trees with a single, relatively straight stem, such as coniferous trees in the boreal forest zone. Adaptations to other forest types, such as temperate forests with deciduous trees, may require modifications to the stem attribute estimation algorithm if the assumption of a single, straight stem is not valid.

6. Conclusions

We demonstrated a first full processing flow for a harvester equipped with an MLS system, starting from a SLAM-based positioning solution followed by algorithmic detection of trees and measurement of their stem attributes.

The SLAM-based positioning solution was shown to be capable of real-time operation with a positioning error of 2.45 m (for *online* trajectory with the *initial fix*) across the three studied test strips. The test strips were linear, representing a real-world harvesting operation devoid of any loop closures that are crucial for positioning a typical SLAM-based

MLS. The error was reduced to 0.21 m (for *final* trajectory with the *optimized fix*) after performing all the measurement updates and determining an improved transformation to the global coordinate system by comparing the final trajectory to the reference one. Only approximately 2 cm of the error can be attributed to random error, and the rest to the accumulating drift perpendicular to the travel direction, and the quality of the fix, i.e., miss-alignment between the map and the georeferenced world frame. We hope to improve the drift and the fix in future work by incorporating existing (potentially incomplete) tree maps, e.g., from previous harvester missions or from national-level ALS scans.

The full SLAM-corrected point clouds were subsequently utilized for stem attribute estimation. By correcting the distance-dependent bias caused by the relatively large beam width of the laser scanner, we estimated the stem diameter at an RMSE of approximately 3 cm with a completeness of 90% for trees larger than 20 cm in diameter and within 15 m of the harvester. Even though harvester MLS data acquisition does not collect a complete, 360-degree view of the tree stems, the obtained results are similar to the state-of-the-art MLS methods using the Ouster OS0 scanner in similar forest conditions. However, significant improvements are required for real-time detection and attribute estimation of the trees. We foresee that such an improved system may ultimately enable real-time optimization of the harvester operation and also help to automatically collect vast amounts of accurate training data for large-scale ALS-based forest inventories at the individual tree level.

Author Contributions: T.F. developed the SLAM algorithm and performed analysis related to the positioning with H.H. supporting. E.H. developed the algorithms for automatic tree measurements and conducted the associated analysis with J.M. supporting. T.H. and T.F. built the MLS system and integrated it to the harvester with H.H., H.K., and A.K. supporting. T.F., T.H., H.H., H.K., and A.K. conducted measurements using the harvester-based MLS system. H.K. conducted reference measurements using the total station and the Zeb Horizon scanner. E.H. and T.F. wrote the manuscript with support and comments from all authors. E.H., T.F., and H.H. performed the visualizations, and H.H. created the graphical abstract. A.K. and J.H. acquired funding for the project and supervised the work. All authors have read and agreed to the published version of the manuscript.

Funding: We gratefully acknowledge Academy of Finland that funded this research through grants "Forest-Human-Machine Interplay—Building Resilience, Redefining Value Networks and Enabling Meaningful Experiences" (357908) and "Collecting Accurate Individual Tree Information for Harvester Operation Decision Making" (359554) and the Ministry of Agriculture and Forestry of Finland and the European Union NextGenerationEU through the project "IlmoStar" (VN/27353/2022). Academy-funded research infrastructure grant "Measuring Spatiotemporal Changes in Forest Ecosystem" (346382) was also applied in this study, especially for the hardware.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to the sensitivity of the private forest data.

Acknowledgments: The authors thank Ville Kankare for taking photographs during the test. Namely, inset B in Figures 1a and 2c are taken by him. The authors thank also Kimmo Savolainen and Petri Kuokkanen from Motoajo Oy for organizing the test site and operating their harvester on the tests.

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

Abbreviations

The following abbreviations are used in this manuscript:

GNSS	Global Navigation Satellite Systema
DBH	Diameter at Breast Height
SLAM	Simultaneous Localization and Mapping
ALS	Airborne Laser Scanning
IMU	Inertial Measurement Unit
NDT	Normal Distributions Transform
MLS	Mobile Laser Scanning

DOF Degrees of Freedom
EKF Extended Kalman Filter
SE(3) Special Euclidean Group

Appendix A. Strip-Wise Results for Stem Detection and Stem Attribute Estimation

In this section, we summarize strip-wise results for stem detection in Table A1 and stem attribute estimation in Table A2 for both studied parameter modes *accurate attributes* and *tree map*.

Table A1. Completeness rate of stem detection for the studied DBH categories on each of the test strips considering trees within a range of ±15 m from the harvester trajectory. For both studied parameter modes accurate attributes and *tree map*, the completeness rate in each DBH category is reported, and the overall correctness rate of stem detection is also provided. For the completeness and correctness rates, the average across the strips is reported in the last row. The values within parenthesis refer to the number of trees used to compute the completeness and correctness rates.

	Completeness (%)					**Correctness (%)**
	DBH $\in$[0, 20) cm	**DBH $\in$[20, 28) cm**	**DBH $\in$[28, 36) cm**	**DBH $\in[36, \infty)$ cm**	**Overall**	
accurate attributes						
Test strip 1	8.1 (9/111)	48.1 (38/79)	65.4 (34/52)	60.0 (6/10)	34.5 (87/252)	95.6 (87/91)
Test strip 2	12.8 (5/39)	50.0 (31/62)	51.5 (34/66)	68.2 (15/22)	45.0 (85/189)	97.7 (85/87)
Test strip 5	11.8 (2/17)	42.4 (42/99)	54.5 (50/91)	56.3 (9/16)	46.2 (103/223)	97.2 (103/106)
All strips	9.6 (16/167)	46.3 (111/240)	56.5 (118/209)	62.5 (30/48)	41.4 (275/664)	96.8 (275/284)
tree map						
Test strip 1	53.2 (59/111)	92.4 (73/79)	94.2 (49/52)	80.0 (8/10)	75.0 (189/252)	74.1 (189/255)
Test strip 2	33.3 (13/39)	87.1 (54/62)	83.3 (55/66)	100.0 (22/22)	76.2 (144/189)	76.6 (144/188)
Test strip 5	58.8 (10/17)	85.7 (84/98)	92.3 (84/91)	81.3 (13/16)	86.0 (191/222)	83.4 (191/229)
All strips	49.1 (82/167)	88.3 (211/239)	90.0 (188/209)	89.6 (43/48)	79.0 (524/663)	78.0 (524/672)

Table A2. Absolute and relative bias, RMSE, and MAE of DBH and stem curve estimates for each of the test strips together with average results based on all detected stems across the strips. The table includes results for both of the considered parameter modes and also provides the average results considering all trees on the test strips. The statistics are based on the trees detected within a range of ±15 m from the harvester trajectory.

	DBH					
	Bias (cm)	**Bias (%)**	**RMSE (cm)**	**RMSE (%)**	**MAE (cm)**	**MAE (%)**
Parameter mode ***accurate attributes***						
Test strip 1	−0.3	−1.2	1.9	6.9	1.3	4.6
Test strip 2	0.6	1.9	1.9	6.3	0.9	3.1
Test strip 5	0.0	0.0	2.4	8.2	1.0	3.5
All strips	0.1	0.2	2.1	7.3	1.0	3.4
Parameter mode ***tree map***						
Test strip 1	−0.4	−1.5	3.9	16.2	1.5	6.3
Test strip 2	−0.2	−0.7	2.5	8.6	1.3	4.6
Test strip 5	−0.4	−1.3	3.0	10.5	1.6	5.7
All strips	−0.3	−1.2	3.2	12.0	1.5	5.5
	Stem Curve					
	Bias (cm)	**Bias (%)**	**RMSE (cm)**	**RMSE (%)**	MAE **(cm)**	**MAE (%)**
Parameter mode ***accurate attributes***						
Test strip 1	−0.2	−0.6	2.3	8.5	1.4	5.3
Test strip 2	0.4	1.4	2.3	7.9	1.2	4.0
Test strip 5	0.2	0.6	2.4	8.5	1.0	3.5
All strips	0.1	0.5	2.3	8.3	1.2	4.3
Parameter mode ***tree map***						
Test strip 1	0.2	0.8	4.4	18.9	2.0	8.4
Test strip 2	−0.2	−0.8	3.0	10.5	1.7	5.9
Test strip 5	−0.2	−0.8	3.2	11.6	1.8	6.6
All strips	−0.1	−0.3	3.6	13.9	1.8	6.9

References

1. Visser, R.; Obi, O.F. Automation and robotics in forest harvesting operations: Identifying near-term opportunities. *Croat. J. For. Eng.* **2021**, *42*, 13–24. [CrossRef]
2. Ponsse. *Ponsse to Demonstrate New Solutions at FinnMETKO 2022*; Ponsse: Vieremä, Finland, 2022. Available online: https://www.ponsse.com/company/news/-/asset_publisher/P4s3zYhpxHUQ/content/ponsse-to-demonstrate-new-solutions-at-finnmetko-2022#/ (accessed on 2 August 2024).
3. Ryding, J.; Williams, E.; Smith, M.J.; Eichhorn, M.P. Assessing handheld mobile laser scanners for forest surveys. *Remote Sens.* **2015**, *7*, 1095–1111. [CrossRef]
4. Kukko, A.; Kaijaluoto, R.; Kaartinen, H.; Lehtola, V.V.; Jaakkola, A.; Hyyppä, J. Graph SLAM correction for single scanner MLS forest data under boreal forest canopy. *ISPRS J. Photogramm. Remote Sens.* **2017**, *132*, 199–209. [CrossRef]
5. Pierzchała, M.; Giguère, P.; Astrup, R. Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. *Comput. Electron. Agric.* **2018**, *145*, 217–225. [CrossRef]
6. Cabo, C.; Del Pozo, S.; Rodríguez-Gonzálvez, P.; Ordóñez, C.; González-Aguilera, D. Comparing terrestrial laser scanning (TLS) and wearable laser scanning (WLS) for individual tree modeling at plot level. *Remote Sens.* **2018**, *10*, 540. [CrossRef]
7. Nevalainen, P.; Li, Q.; Melkas, T.; Riekki, K.; Westerlund, T.; Heikkonen, J. Navigation and mapping in forest environment using sparse point clouds. *Remote Sens.* **2020**, *12*, 4088. [CrossRef]
8. Hyyppä, E.; Kukko, A.; Kaijaluoto, R.; White, J.C.; Wulder, M.A.; Pyörälä, J.; Liang, X.; Yu, X.; Wang, Y.; Kaartinen, H.; et al. Accurate derivation of stem curve and volume using backpack mobile laser scanning. *ISPRS J. Photogramm. Remote Sens.* **2020**, *161*, 246–262. [CrossRef]
9. Hyyppä, E.; Hyyppä, J.; Hakala, T.; Kukko, A.; Wulder, M.A.; White, J.C.; Pyörälä, J.; Yu, X.; Wang, Y.; Virtanen, J.P.; et al. Under-canopy UAV laser scanning for accurate forest field measurements. *ISPRS J. Photogramm. Remote Sens.* **2020**, *164*, 41–60. [CrossRef]
10. Bienert, A.; Georgi, L.; Kunz, M.; von Oheimb, G.; Maas, H.G. Automatic extraction and measurement of individual trees from mobile laser scanning point clouds of forests. *Ann. Bot.* **2021**, *128*, 787–804. [CrossRef]
11. Liang, X.; Kukko, A.; Balenović, I.; Saarinen, N.; Junttila, S.; Kankare, V.; Holopainen, M.; Mokroš, M.; Surovỳ, P.; Kaartinen, H.; et al. Close-range remote sensing of forests: The state of the art, challenges, and opportunities for systems and data acquisitions. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 32–71. [CrossRef]
12. Kaartinen, H.; Hyyppä, J.; Vastaranta, M.; Kukko, A.; Jaakkola, A.; Yu, X.; Pyörälä, J.; Liang, X.; Liu, J.; Wang, Y.; et al. Accuracy of Kinematic Positioning Using Global Satellite Navigation Systems under Forest Canopies. *Forests* **2015**, *6*, 3218–3236. [CrossRef]
13. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]
14. Pohjala, J.; Vahtila, M.; Ovaskainen, H.; Kankare, V.; Hyyppä, J.; Kärhä, K. Effect of Prior Tree Marking on Cutting Productivity and Harvesting Quality. *Croat. J. For. Eng. J. Theory Appl. For. Eng.* **2024**, *45*, 25–42. [CrossRef]
15. Hauglin, M.; Hansen, E.; Sørngård, E.; Næsset, E.; Gobakken, T. Utilizing accurately positioned harvester data: Modelling forest volume with airborne laser scanning. *Can. J. For. Res.* **2018**, *48*, 913–922. [CrossRef]
16. Saukkola, A.; Melkas, T.; Riekki, K.; Sirparanta, S.; Peuhkurinen, J.; Holopainen, M.; Hyyppä, J.; Vastaranta, M. Predicting forest inventory attributes using airborne laser scanning, aerial imagery, and harvester data. *Remote Sens.* **2019**, *11*, 797. [CrossRef]
17. Rossmann, J.; Schluse, M.; Schlette, C.; Buecken, A.; Krahwinkler, P.; Emde, M. Realization of a highly accurate mobile robot system for multi purpose precision forestry applications. In Proceedings of the 2009 International Conference on Advanced Robotics, Munich, Germany, 22–26 June 2009; pp. 1–6.
18. Sihvo, S.; Virjonen, P.; Nevalainen, P.; Heikkonen, J. Tree Detection around Forest Harvester Based on Onboard LiDAR Measurements. In Proceedings of the 2018 Baltic Geodetic Congress, BGC-Geomatics 2018, Olsztyn, Poland, 21–23 June 2018; pp. 364–367.
19. Salmivaara, A.; Miettinen, M.; Finér, L.; Launiainen, S.; Korpunen, H.; Tuominen, S.; Heikkonen, J.; Nevalainen, P.; Sirén, M.; Ala-Ilomäki, J.; et al. Wheel rut measurements by forest machine-mounted LiDAR sensors–accuracy and potential for operational applications? *Int. J. For. Eng.* **2018**, *29*, 41–52. [CrossRef]
20. Jutila, J.; Kannas, K.; Visala, A. Tree measurement in forest by 2D laser scanning. In Proceedings of the 2007 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA), Jacksonville, FL, USA, 20–23 June 2007; pp. 491–496.
21. Miettinen, M.; Öhman, M.; Visala, A.; Forsman, P. Simultaneous localization and mapping for forest harvesters. In Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 517–522.
22. Hyyti, H.; Visala, A. Feature Based Modeling and Mapping of Tree Trunks and Natural Terrain Using 3D Laser Scanner Measurement System. *IFAC Proc. Vol.* **2013**, *46*, 248–255. [CrossRef]
23. Tremblay, J.F.; Béland, M.; Pomerleau, F.; Gagnon, R.; Giguère, P. Automatic 3D Mapping for Tree Diameter Measurements in Inventory Operations. *arXiv* **2019**, arXiv:1904.05281.
24. Xu, X.; Zhang, L.; Yang, J.; Cao, C.; Wang, W.; Ran, Y.; Tan, Z.; Luo, M. A Review of Multi-Sensor Fusion SLAM Systems Based on 3D LIDAR. *Remote Sens.* **2022**, *14*, 2835. [CrossRef]
25. Zhang, J.; Singh, S. Low-drift and real-time lidar odometry and mapping. *Auton. Robot.* **2017**, *41*, 401–416. [CrossRef]

26. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. *arXiv* **2020**, arXiv:2007.00258.
27. Qu, C.; Shivakumar, S.S.; Liu, W.; Taylor, C.J. LLOL: Low-Latency Odometry for Spinning Lidars. *arXiv* **2021**, arXiv:2110.01725.
28. Xu, W.; Cai, Y.; He, D.; Lin, J.; Zhang, F. FAST-LIO2: Fast Direct LiDAR-inertial Odometry. *arXiv* **2021**, arXiv:2107.06829.
29. Rossmann, J.; Krahwinkler, P.; Schlette, C. Navigation of Mobile Robots in Natural Environments: Using Sensor Fusion in Forestry. *J. Syst. Cybern. Inform.* **2010**, *8*, 67–71.
30. Chen, Y.; Tang, J.; Khoramshahi, E.; Hakala, T.; Kaartinen, H.; Jaakkola, A.; Hyyppa, J.; Zhu, Z.; Chen, R. Scan matching technology for forest navigation with map information. In Proceedings of the IEEE/ION Position, Location and Navigation Symposium, PLANS 2016, Savannah, GA, USA, 11–14 April 2016; pp. 198–203.
31. Li, Q.; Nevalainen, P.; Queralta, J.P.; Heikkonen, J.; Westerlund, T. Localization in Unstructured Environments: Towards Autonomous Robots in Forests with Delaunay Triangulation. *Remote Sens.* **2020**, *12*, 1870. [CrossRef]
32. de Lima, L.C.; Ramezani, M.; Borges, P.; Brunig, M. Air-Ground Collaborative Localisation in Forests Using Lidar Canopy Maps. *IEEE Robot. Autom. Lett.* **2023**, *8*, 1818–1825. [CrossRef]
33. Bauwens, S.; Bartholomeus, H.; Calders, K.; Lejeune, P. Forest inventory with terrestrial LiDAR: A comparison of static and hand-held mobile laser scanning. *Forests* **2016**, *7*, 127. [CrossRef]
34. Balenović, I.; Liang, X.; Jurjević, L.; Hyyppä, J.; Seletković, A.; Kukko, A. Hand-Held Personal Laser Scanning–Current Status and Perspectives for Forest Inventory Application. *Croat. J. For. Eng. J. Theory Appl. For. Eng.* **2021**, *42*, 165–183.
35. Vandendaele, B.; Martin-Ducup, O.; Fournier, R.A.; Pelletier, G.; Lejeune, P. Mobile laser scanning for estimating tree structural attributes in a temperate hardwood Forest. *Remote Sens.* **2022**, *14*, 4522. [CrossRef]
36. Liang, X.; Hyyppä, J.; Kukko, A.; Kaartinen, H.; Jaakkola, A.; Yu, X. The use of a mobile laser scanning system for mapping large forest plots. *IEEE Geosci. Remote Sens. Lett.* **2014**, *11*, 1504–1508. [CrossRef]
37. Hyyppä, E.; Kukko, A.; Kaartinen, H.; Yu, X.; Muhojoki, J.; Hakala, T.; Hyyppä, J. Direct and automatic measurements of stem curve and volume using a high-resolution airborne laser scanning system. *Sci. Remote Sens.* **2022**, *5*, 100050. [CrossRef]
38. Hyyppä, E.; Yu, X.; Kaartinen, H.; Hakala, T.; Kukko, A.; Vastaranta, M.; Hyyppä, J. Comparison of backpack, handheld, under-canopy UAV, and above-canopy UAV laser scanning for field reference data collection in boreal forests. *Remote Sens.* **2020**, *12*, 3327. [CrossRef]
39. Ouster. *OS0: Ultra-Wide View High-Resolution Imaging Lidar*; Revision: 12/07/2020, Hardware Version: Rev C; Ouster, Inc.: San Francisco, CA, USA, 2020. Available online: https://data.ouster.io/downloads/datasheets/datasheet-rev05-v2p1-os0.pdf (accessed on 1 August 2024).
40. Leica. *Leica Nova TS60*;Retrieved 12.2.2024; Leica: Heerbrugg, Switzerland, 2020. Available online: https://leica-geosystems.com/products/total-stations/robotic-total-stations/leica-nova-ts60 (accessed on 1 August 2024).
41. Chen, S.; Liu, H.; Feng, Z.; Shen, C.; Chen, P. Applicability of personal laser scanning in forestry inventory. *PLoS ONE* **2019**, *14*, e0211392. [CrossRef]
42. Hunčaga, M.; Chudá, J.; Tomaštík, J.; Slámová, M.; Koreň, M.; Chudý, F. The Comparison of Stem Curve Accuracy Determined from Point Clouds Acquired by Different Terrestrial Remote Sensing Methods. *Remote Sens.* **2020**, *12*, 2739. [CrossRef]
43. Gollob, C.; Ritter, T.; Nothdurft, A. Forest inventory with long range and high-speed personal laser scanning (PLS) and simultaneous localization and mapping (SLAM) technology. *Remote Sens.* **2020**, *12*, 1509. [CrossRef]
44. Faitli, T.; Hakala, T.; Kaartinen, H.; Hyyppä, J.; Kukko, A. Real-Time Lidar-Inertial Positioning and Mapping for Forestry Automation. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2023**, *XLVIII-1/W1-2023*, 145–150.
45. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Trans. Robot.* **2017**, *33*, 1–21. [CrossRef]
46. Solà, J.; Deray, J.; Atchuthan, D. A micro Lie theory for state estimation in robotics. *arXiv* **2018**, arXiv:1812.01537.
47. Dellaert, F.; Kaess, M. Factor Graphs for Robot Perception. *Found. Trends Robot.* **2017**, *6*, 1–139. [CrossRef]
48. 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 KDD, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
49. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
50. Muhojoki, J.; Tavi, D.; Hyyppä, E.; Lehtomäki, M.; Faitli, T.; Kaartinen, H.; Kukko, A.; Hakala, T.; Hyyppä, J. Benchmarking Under-and Above-Canopy Laser Scanning Solutions for Deriving Stem Curve and Volume in Easy and Difficult Boreal Forest Conditions. *Remote Sens.* **2024**, *16*, 1721. [CrossRef]
51. Forsman, M.; Börlin, N.; Olofsson, K.; Reese, H.; Holmgren, J. Bias of cylinder diameter estimation from ground-based laser scanners with different beam widths: A simulation study. *ISPRS J. Photogramm. Remote Sens.* **2018**, *135*, 84–92. [CrossRef]
52. Ringdahl, O.; Hohnloser, P.; Hellström, T.; Holmgren, J.; Lindroos, O. Enhanced Algorithms for Estimating Tree Trunk Diameter Using 2D Laser Scanner. *Remote Sens.* **2013**, *5*, 4839–4856. [CrossRef]
53. Kuželka, K.; Surový, P. Noise Analysis for Unbiased Tree Diameter Estimation from Personal Laser Scanning Data. *Remote Sens.* **2024**, *16*, 1261. [CrossRef]
54. Hyyppä, E.; Muhojoki, J.; Yu, X.; Kukko, A.; Kaartinen, H.; Hyyppä, J. Efficient coarse registration method using translation-and rotation-invariant local descriptors towards fully automated forest inventory. *ISPRS Open J. Photogramm. Remote Sens.* **2021**, *2*, 100007. [CrossRef]

55. Wielgosz, M.; Puliti, S.; Wilkes, P.; Astrup, R. Point2Tree (P2T)–framework for parameter tuning of semantic and instance segmentation used with mobile laser scanning data in coniferous forest. *arXiv* **2023**, arXiv:2305.02651.
56. Xiang, B.; Wielgosz, M.; Kontogianni, T.; Peters, T.; Puliti, S.; Astrup, R.; Schindler, K. Automated forest inventory: Analysis of high-density airborne LiDAR point clouds with 3D deep learning. *arXiv* **2023**, arXiv:2312.15084.
57. National Land Survey of Finland. *Laser Scanning Data 5 p*; National Land Survey of Finland: Helsinki, Finland, 2023. Available online: https://www.maanmittauslaitos.fi/en/maps-and-spatial-data/datasets-and-interfaces/product-descriptions/laser-scanning-data-5-p (accessed on 1 August 2024).
58. Maltamo, M.; Kinnunen, H.; Kangas, A.; Korhonen, L. Predicting stand age in managed forests using National Forest Inventory field data and airborne laser scanning. *For. Ecosyst.* **2020**, *7*, 44. [CrossRef]
59. Yu, X.; Hyyppä, J.; Kukko, A.; Maltamo, M.; Kaartinen, H. Change detection techniques for canopy height growth measurements using airborne laser scanner data. *Photogramm. Eng. Remote Sens.* **2006**, *72*, 1339–1348. [CrossRef]
60. Polewski, P.; Yao, W.; Cao, L.; Gao, S. Marker-free coregistration of UAV and backpack LiDAR point clouds in forested areas. *ISPRS J. Photogramm. Remote Sens.* **2019**, *147*, 307–318. [CrossRef]
61. Ghorbani, F.; Chen, Y.C.; Hollaus, M.; Pfeifer, N. A Robust and Automatic Algorithm for TLS - ALS Point Cloud Registration in Forest Environments based on Tree Locations. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *17*, 4015–4035.