Online Estimation of Diameter at Breast Height (DBH) of Forest Trees Using a Handheld LiDAR

Alexander Proudman[1], Milad Ramezani[1] and Maurice Fallon[1]

***Abstract*— While mobile LiDAR sensors are increasingly used to scan in ecology and forestry applications, reconstruction and characterisation are typically carried out offline (to the best of our knowledge). Motivated by this, we present an online LiDAR system which can run on a handheld device to segment and track individual trees and identify them in a fixed coordinate system. Segments relating to each tree are accumulated over time, and tree models are completed as more scans are captured from different perspectives. Using this reconstruction we then fit a cylinder model to each tree trunk by solving a least-squares optimisation over the points to estimate the Diameter at Breast Height (DBH) of the trees. Experimental results demonstrate that our system can estimate DBH to within ∼7 cm accuracy for 90% of individual trees in a forest (Wytham Woods, Oxford).**

I. Introduction

Researchers in ecology and forestry monitor the size and growth of individual trees within a forest to infer arboreal health. It is common to retrieve quantitative metrics such as Diameter at Breast Height (DBH) using manual methods, such as measuring tapes, to ensure consistency with previous surveys.

More recently approaches using 3D LiDAR scanners have been presented for tree segmentation and the construction of Quantitative Structure Models (QSMs) [1]. These models can be used to estimate the above-ground biomass and carbon stock by estimating the volume of individual trees [2], [3]. However, retrieval of these tree-scale metrics requires data processing of a large-scale point cloud which is often conducted offline in existing approaches.

In this paper, we present an online point-cloud processing tool chain which can segment and track individual trees relative to a fixed coordinate frame. The ground surface can also be inferred, allowing us to estimate the DBH of each tree (at 1.4m height) as the scanning process goes on.

The contributions of this work are as follows:

- An online system to track trees in dense forests and to identify individual trees.
- An experimental evaluation, of our system, in a forest area and the evaluation of the tree DBH using our system. Results are compared to measurements taken manually at the test site as well as comparison against a commercial tool (using the final accumulated point cloud).

Video: `https://youtu.be/Ut7d2eOypko`

This research is supported by the UKRI/ESPRC ORCA Robotics Hub (EP/R026173/1). M. Fallon is supported by a Royal Society University Research Fellowship.

[1] These authors are with the Oxford Robotics Institute, University of Oxford, UK. `alexander.proudman@wadham.ox.ac.uk`, {`milad, mfallon`}`@robots.ox.ac.uk`

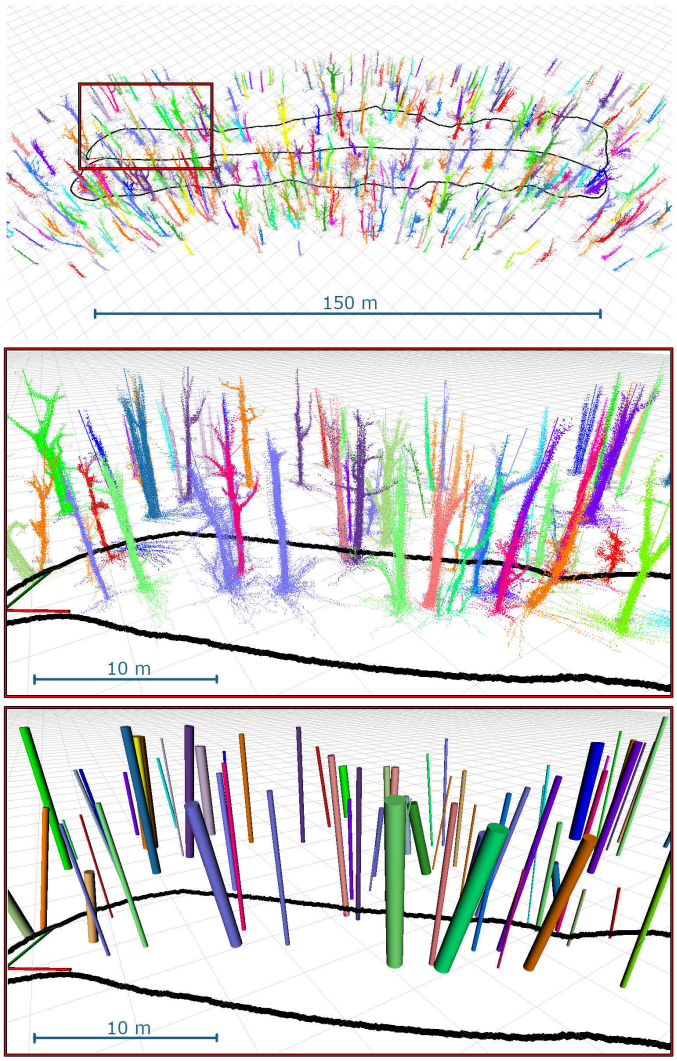

Fig. 1: Visualisation of the system output. **Top**: a high level view showing a 150 m transect. The path taken by the person carrying the sensor in Fig. 6 is shown in black. **Centre** and **Bottom**: Detail of a section is shown with two different representations: one showing each tree segmented and individually coloured and the other shows a cylinder (primary axis and diameter) fitted at breast height.

The remainder of this paper is organised as follows. In Section II we discuss related work. Section III explains how we designed our methods. Experimental results are presented in Section IV. Section V discusses conclusions and suggests some topics for future work.

II. Related Work

Advances in LiDAR technology have led to it being used in non-traditional fields such as ecology and forestry. Both

stationary terrestrial laser scanning and portable scanning LiDAR have been used. In this section we review the approaches that utilise LiDAR for individual tree segmentation.

A. Tree Segmentation Using Terrestrial LiDAR

Many existing tree segmentation techniques operate on a single unified point cloud collected by a Terrestrial Laser Scanner (TLS) from a small number of static scanning stations in post-processing. Raumonen *et al.* [4] and later Trochta *et al.* [5] clustered these scans into smaller point clouds to work around large size of these scans and to reduce memory consumption. They then segmented tree-level clouds within each cluster by growing segments using assumptions of fixed inter-cluster distance and orientation to infer connectivity. Methods such as [6] benefited from graph theory to find the connectivity between adjacent points and to finally segment each tree. These approaches, however, rely on multiple assumptions about a tree's architecture as well as assuming minimal interconnection between their crowns.

Recently, Burt *et al.* [7] presented a software package named *treeseg* in which a region-growing technique is used to segment individual trees. Key features in *treeseg*'s design are its independence of forest type and instrument, and assumptions about the tree structures. However, *treeseg* and the techniques aforementioned were mostly tested with high quality data collected by stationary laser scanners. The point clouds of these scanners were very accurate with minimal noise and therefore tree individuals were easy to identify.

There are also some techniques to automatically segment trees utilising learning pipelines such as a trained random forest classifier [8] or Convolutional Neural Networks (CNNs) [9], however these were applied offline using data collected from individual trees.

B. Tree Segmentation Using Mobile LiDAR

Although mobile LiDAR systems, which are normally carried by an agent at walking pace, make ranging measurements with lower precision than TLS, they have drawn attention in environmental science because of the falling cost of the sensor and the quick data collection time. Heo *et al.* [10] used mobile LiDAR to collect data in an urban area, including parks and streets, to estimate the height of trees and their DBH by calculating the height-above-ground and using a least-squares circle fit approach [11]. They emphasised the advantage of using mobile LiDAR to reduce shadow/occlusion effects, which are more prominent with terrestrial LiDAR systems, especially in an urban environment. They utilised a Stencil LiDAR system produced by Kaarta[1] which includes a Velodyne sensor.

Similarly, Zhou *et al.* [12] collected LiDAR data with a Velodyne VLP-16 LiDAR sensor. In offline processing the authors estimated the DBH of the trees. They removed the points residing on the ground and estimated the DBH using Random Sample Consensus (RANSAC) algorithm on the segments produced by an Euclidean-based clustering algorithm [13].

[1]https://www.kaarta.com/

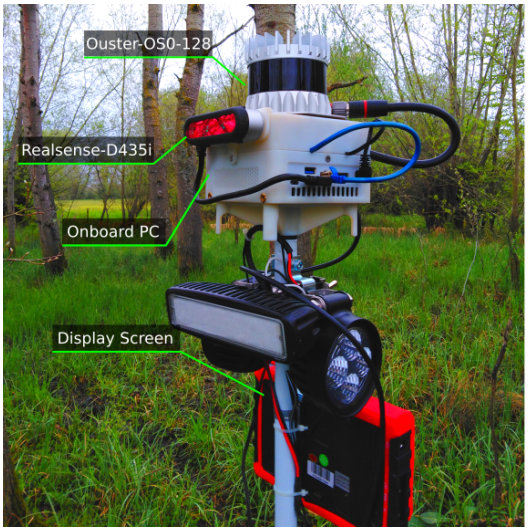

Fig. 2: The handheld device used in our system. The labelled components work together to provide online feedback on the forestry data collection

Westling *et al.* [14] scanned individual avocado trees with a 5 m spacing using a GeoSLAM Zebedee 1 handheld device. The authors first voxelised point clouds and conducted a graph-based search over the voxels to find all paths connecting to a root voxel, which was considered to be the tree node. The tree node is segmented from the ground by comparing the height of points locally within a search radius.

These approaches have been tested after data collection which doesnt allow the user to perceive the reconstruction during scanning. Our motivation is to develop a LiDAR-driven technique which can estimate the structural parameters of individual trees in dense forestry areas with rough terrain in real-time. A real-time mapping system can ensure full coverage by providing feedback to the operator, so that gaps are not left, ensuring the environment is fully scanned and processed at runtime.

III. Methods

As shown in Fig. 2, our device consists of an Ouster OS0-128 LiDAR and an Intel Realsense D435i, each with a built-in IMU. However, we only use the data from the LiDAR sensor and its IMU measurements for our LiDAR odometry and tree tracking systems. The LiDAR has 128 beams and a 90° vertical field of view. While it cannot completely cover the environment vertically, the 90° vertical field is sufficient to identify the individual trees within about 20 m of the sensor. The sensors used in systems such as Zeb-Revo or Zebedee[2] have narrow field of view and require the sensor to be either rotated or oscillated to cover the environment entirely.

Our proposed system architecture is shown in Fig. 3. Due to the system's real-time nature, the pipeline is different from many existing methods, and alterations have been made to parallelise and speed up the overall pipeline. For example our system builds up an elevation map rather than

[2]https://geoslam.com/

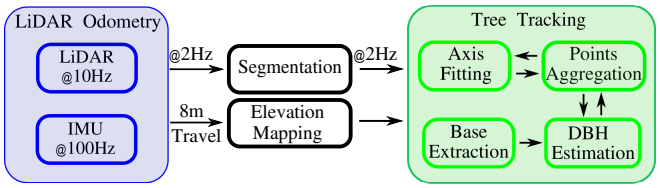

Fig. 3: An overview of the system pipeline, the frequencies of communication between blocks are given. Elevation mapping is triggered very 8 m travelled.

removing ground points prior to segmentation as done by Heo *et al.* [10] and Sanzhang *et al.* [12]. Removing the ground points improves the segmentation of trees however, this method has the potential to impede the real-time operation of our system, see Sec. IV for demonstration of this. Our system develops counter-measures to the potentially poorer segmentation detailed throughout this section specifically Sec. III-D.1.

In the following, we describe the algorithms and methods used by each functional block within the system architecture. Most methods used have been proven to work for terrestrial LiDAR in post-processing. We use a LiDAR-inertial odometry module to accumulate a point cloud reconstruction of the environment around the sensor. The *Tree Tracking* block, not used in the typical LiDAR forest inventory systems, segments and processes each tree detection. Advantages of using this online fused data approach are that:

- Prior inventory data can be used to automatically identify trees.
- Success of the data collection can be evaluated online, including the sensor's coverage.
- Data collection and analysis only takes the amount of time needed to walk along a selected transect (a narrow section of land along which measurements are taken).

A. LiDAR Odometry

Our LiDAR odometry system [15] is a factor-graph based windowed smoother which fuses Inertial Measurement Unit (IMU) readings with measurements from a multi-beam LiDAR scanner mounted on a handheld device (Fig. 2). The particular configuration used in this work, uses IMU pre-integration to remove motion distortion from scans and to initialise Iterative Closest Point (ICP) registration, specifically the implementation of Pomerleau [16], to determine the relative transform for every 2 metres of distance travelled.

The IMU measurements and the relative transforms then form constraints in a factor graph to estimate the poses in a sliding window optimisation utilising iSAM2 [17] (as part of the GTSAM library). Additionally, since IMU provides gravity information, we can align the clouds with the gravity direction. This alignment helps determine the base of trees using the z direction. Evaluation of our LiDAR odometry system is discussed in Sec. IV-A.

B. Segmentation

To roughly extract the tree trunks, branches, canopy and shrubbery in the original point cloud, Euclidean segmentation is carried out. Using the PCL library[3], the cloud is first downsampled using a voxel filter to lessen the radial variation of the point density and to speed up the segmentation process as also done by [7].

After voxel filtering the point cloud, it is reformulated into a k-d tree to quickly find the nearest neighbours of every point. These points are clustered together into groups of points which are within some distance threshold of each other.

Note that the parameters of the voxel filtering and Euclidean segmentation methods are tuned based on the LiDAR sensor's characteristics to maximise the chances that the point clusters contain individual trees.

C. Elevation Mapping

To perform *Base Segmentation* of the trees being tracked, we employ an open-source sensor-centric elevation mapping framework[4] which is built upon a universal Grid Map library [18]. The algorithm, presented in [19], generates consistent elevation maps of the environment surrounding the sensor, at the same frequency of the LiDAR output (10 Hz). In this work, we use a resolution of 16 cm for the grid of 32m $\times$ 32m.

Nevertheless, due to the presence of foliage and branches, the terrain created by the elevation mapping software can contain phantom spikes, making it difficult to detect the precise base of certain trees. To solve this problem, we compute the slope of each grid cell. Cells with slope greater than a threshold are removed. This removes the majority of spikes, resulting in smooth terrain with some remaining holes. Finally, we utilise a morphological closing filter on the grid map to fill the holes.

Applying the chain of filters is relitievly time consuming so we update the elevation map every 8 m of travel distance based on the state estimate from our LiDAR odometry. This travel distance allows online operation. We analyse the computation time of each component including elevation mapping in the next section. Fig. 4 shows this process.

D. Tree Tracking

To facilitate a formal definition of a tree the following minimal, 'tree descriptor' t was defined as follows:

- A unique id,
- Major axis of the tree incline ($\mathbf{I}$ a vector),
- Diameter at Breast Height (DBH) (D),
- 3D position of the tree base ($\mathbf{b}$),
- The minimum and maximum height of the point cloud defining the tree (min & max).

Every tree descriptor is derived from a corresponding point cloud of the tree points P ($t = f(P)$). These points are accumulated over time, gathering information from different angles as well as reducing the effect of occlusion and improving the estimate of the tree descriptor as Heo *et al.* [10] indicated.

[3]https://pointclouds.org/
[4]https://github.com/ANYbotics/elevation_mapping

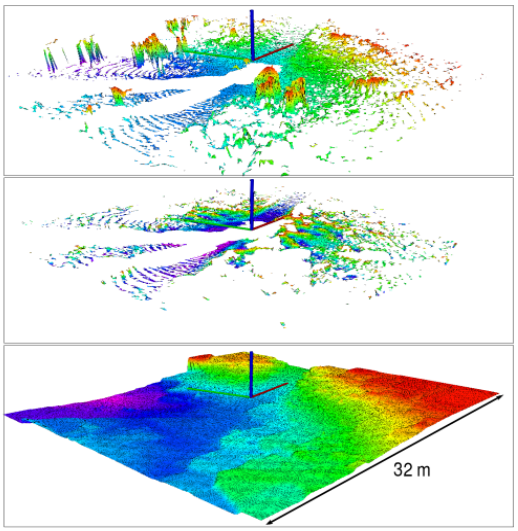

Fig. 4: Filtering process on the elevation map: Original elevation map (top) is filtered to remove spikes (middle). Finally, holes are filled using a morphological closing filter on the grid map (bottom).

Most of the descriptors characteristics are easily evaluated with the exception of the major axis of the tree. We used fitting of a line, to the set of tree points (P), to find this axis. The algorithm is designed to ignore branches and any potentially segmented crown or ground points and is described in Sec. III-D.1.

Internally, the *Tree Tracker* holds an inventory of n distinct trees T defined by a descriptor and set of tree points $T_i \sim t_i, P_i$ for $0 \leq i \leq n$.

Referring to the system diagram, Fig. 3, incoming clusters C from the *Segmentation* block are either discarded or assigned to a new/existing tree descriptor. If assigned to an existing descriptor, the cluster of points is merged with the corresponding set of tree points and the descriptor is re-evaluated $P_{new} = P_{old} \bigcup C$, $t_{new} = f(P_{new})$.

An assign/discard decision is made by evaluating a tree descriptor for each input cluster. A cluster is confirmed and subsequently assigned only if it has characteristics within the tolerances of given parameters. The parameter tolerances that resulted in the most consistent tree assignment are:

- Requiring the major axis of the tree to be close to vertical, i.e. $|\mathbf{I} \cdot [0, 0, 1]| < \theta_{\text{threshold}}$
- Requiring a minimum height, i.e. $max - min < h_{\text{threshold}}$

Should a cluster satisfy the above properties we deem that it contains points belonging to a tree. It is then compared to the existing set of confirmed trees T and merged if a match is found. In order for a match between two clusters to be found, their major axis must converge to within some threshold distance at the base of the highest cluster or at a plane segmenting them. Fig. 5 (left) demonstrates an example of this.

Alg. 1 summarises the *Tree Tracking* pipeline. The evaluation of a match and the subsequent merge process is assessed until no more matches are found for an input cluster. This is important to ensure all potential clusters belonging to a tree are merged.

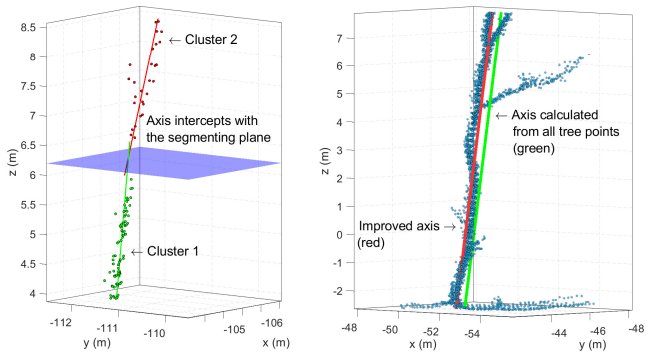

Fig. 5: An example of *Tree Tracking*. **Left**: The two clusters C with their major axis shown and the plane of between them. **Right**: Line of best fit on all tree points (green) and on the points excluding branch and ground points (red).

1) Robust estimation of the tree's major axis: As mentioned earlier, the calculation of the tree's major axis, or a clusters major axis, is done using regression on a subset of the tree's points, to find a line of best fit. Beginning with regression utilising every point in the cluster gives an initial estimate of the incline (**I**) and its intercept.

However, since the objective is to find the axis of the tree's trunk, least squares regression can be affected when the branches, ground or crown points are included in the segmentation. Therefore, points that are a multiple of the standard deviation away from the calculated line or outside the central 90% of the tree's height are ignored. The regression is re-calculated with the subset of points. This is iterated multiple times with successively smaller tolerances on the inclusion of points. This result is shown in Fig. 5 (right).

Algorithm 1: Tree Tracking Algorithm.

input: Set of point clouds C
output: Set of updated trees T
begin
  for Cluster $C_i \subset C$ **do**
    Calculate the tree descriptor $t_i = f(C_i)$
    if t_i *is classified as being a tree* **then**
      if *a first match for* t_i *is found in* T (T_j) **then**
        merge cluster into the set of tree points $P_j = P_j \bigcup C_i$ and re-calculate the tree descriptor $t_j = f(P_j)$
        while *a subsequent match for the updated* t_j *is found in* T (T_{tmp}) **do**
          merge the two sets of tree points $P_j = P_j \bigcup P_{tmp}$ and re-calculate the tree descriptor $t_j = f(P_j)$
          delete T_{tmp} from T
        end
      end
      else
        Push back the t_i, C_i into the set T
      end
    end
  end
end

E. Base Segmentation

As the ground has not been removed prior to segmentation, points that are not necessarily portions of the tree have the potential to be segmented. As a result of this the lowest point in P (minimum z value) may not be an accurate estimate of the tree's base. We segment the base as a result.

The two most recent overlapping elevation maps, from Sec. III-C, are converted into a point cloud so that a search over these 'ground points' can be achieved. Points within a 2 metre radius of the lowest point in P are identified, using a nearest neighbour method from the PCL library. From this set, the closest point to the tree's axis is found and set as the tree's base $\mathbf{b}$.

F. Estimation of Diameter at Breast Height (DBH)

To estimate the DBH of each individual tree, we employ a procedure similar to that used in the literature for general cylinder fitting [11]. This procedure is broken down into three parts: firstly, segmenting the points at breast height, secondly, projection of these points onto a plane, and lastly, fitting a circle to the projected points.

Utilising the method described by Zhou *et al.* [12] and Heo *et al.* [10], our system segments the LiDAR points, for recently updated trees in T. Points (S), from the accumulated point cloud (P), that are located within a 10 cm height range centered 1.4 m above the tree's base are segmented. These points are projected onto a plane with normal ($\hat{\mathbf{n}}$) equal to the tree's incline ($\mathbf{I}$) to ensure the set of points can be modelled closely to a circle. The alternative is to use elliptical fitting for trees angled from the vertical. In practice the sensor only captures data from a single side of the tree, which may cause overfitting when using an elliptical distribution, especially for trees which have irregularly curved surfaces.

The potential problem with this method is that these points are taken from the downsampled point cloud so are likely to be inaccurate. However, an advantage of this method is that points should cover a greater proportion of the tree's surface if the LiDAR sensor measured it from a set of different angles.

RANSAC circle fitting is then used on the set of resultant 2D points, which is robust to potential outliers from errors in segmentation [20].

IV. Experiments and Evaluation

The described system was implemented in C++ and the Robot Operating System (ROS) was used for communication between the hand-held device and the software modules.

Evaluation of the system was carried out using data collected in Wytham Woods, Oxford, UK. The woods is 400 ha in area and has been used for ecology research for over 80 years. It is a Smithsonian ForestGEO site; which is a collection of forestry plots located world-wide used for collective ecology research. Within this scheme, it has taken part in 3 mass censuses and about 16200 trees have been manually measured.

We selected 55 trees along a $\sim$150 m transect. The selected trees had various diameters, ranging from $\sim$10 cm to $\sim$80 cm, which could be reasonably measured manually at breast height and had clearly visible bases.

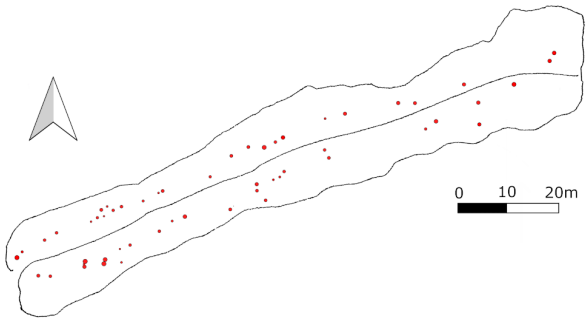

Fig. 6: Visualisation of the trees selected for measurement within Wytham Woods, Oxford, as well as the path taken using the handheld device. The location of the chosen transect is 51°46'60.59"N, 001°20'21.47"W.

We chose a small section of the forest shown in Fig. 6. The DBH of every tree was measured using a tape measure at a height of 1.4 m from the ground prior to the LiDAR data collection. Two types of LiDAR sensors were used: The mobile LiDAR device specified earlier and a Leica BLK terrestrial LiDAR[5] (for comparison). The path taken by the mobile LiDAR can be seen in Fig. 6 and Fig. 1, while a terrestrial LiDAR map was made along the central path to generate the ground truth for evaluation of our LiDAR odometry system and to provide a performance baseline.

A. LiDAR Odometry

To evaluate the odometry component of the system, we used the reconstruction created by the Leica BLK as a prior map. We then registered the individual point clouds from the hand-held LiDAR against the prior map to generate an accurate 6 degree-of-freedom ground truth trajectory at 10 Hz. The estimated trajectory from the lidar odometry system (without access to the Leica map) was then compared against the ground truth trajectory using Relative Translation Error (RTE) at travel distances of between 10 metres and 50 metres to produce Fig. 7.

We aim to accumulate and process LiDAR scans for individual trees as we walk up and pass them at ranges up to 20 m, thus having a rate of just a few centimetres on this scale is satisfactory.

B. Tree Tracking

The *Tree Tracking* system performed satisfactorily within the constraints of online performance and the density of foliage and trees on the test transect. Occasionally, trees were clustered together when they should not have been, this is most often due to low hanging interconnecting branches or foliage close to a tree's base. Out of the 55 measured trees 3 of them contained points which belonged to other objects. It was observed that this was also partially due to a transient

[5]https://leica-geosystems.com/en-gb/products/laser-scanners/scanners/blk360

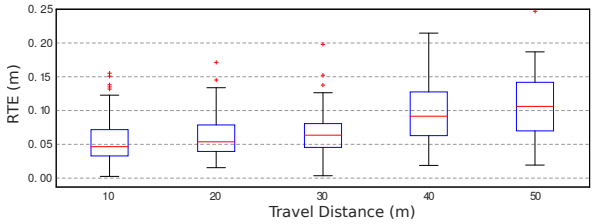

Fig. 7: Relative translation error of the LiDAR odometry at different subtrajectory presented as a series of boxplots. Each box indicates the 25 and 75 percentiles of the estimation errors, the horizontal line through the box the median, and the whiskers are the maximum and minimum excluding outliers.

effect in *Tree Tracking* in which assignment between nearby clusters is uncertain when the tree is first detected. This problem is shown in Fig. 8.

On the other hand, from the 55 measured trees, one failed to be automatically segmented due to its small size, having a DBH ~ 0.11 m. This indicates that our current hardware setup and software configuration limits the performance of the system detecting small trees with DBH below about 10 cm.

C. DBH Estimation with LiDAR360

To provide a baseline, we used the commercial package LiDAR360, in particular its TLS forestry module[6], which is designed for tree segmentation and DBH estimation, on both our handheld LiDAR and Leica BLK datasets.

To evaluate the difference between the estimated values of DBH and the manual ground truth measurements, we use the Root Mean Square Error (RMSE) metric.

LiDAR360 successfully identified 53 of the 55 trees in the Leica BLK data. The remaining two, unidentified trees, were manually selected, then DBH estimation was done, accross all 55 trees, resulting in an overall RMSE of 0.046 m. The mobile LiDAR point clouds were accumulated, after odometry, and also tested using the the same software. Every tree was successfully segmented in the accumulated point cloud and the resulting RMSE was 0.056 m.

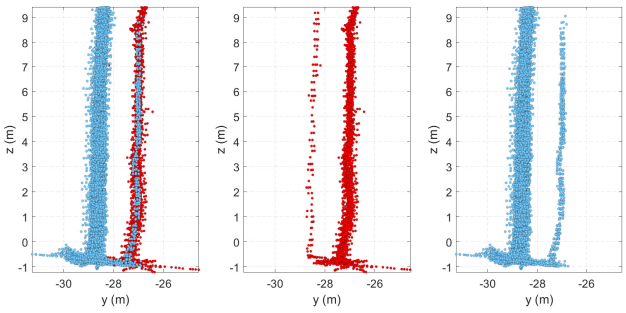

Fig. 8: An example of tree tracking failing for trees in close proximity, the ground points clearly interconnect the two trees. **Left**: The two trees shown entirely. **Center and Right**: Segmented tree with incorrectly assigned points.

[6]https://greenvalleyintl.com/wp-content/GVITutorials/LiDAR360TLSForest/LiDAR360TLSForestDataPreprocessing.html

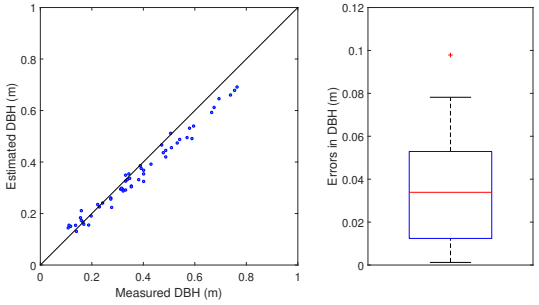

Fig. 9: Estimation of DBH using the LiDAR360 Forestry module (using the mobile LiDAR data). **Left**: Plot of the 55 measured trees vs estimated values as extrapolated from the package. **Right**: Boxplot of the errors between the estimated and measured values, RMSE = 0.056 m.

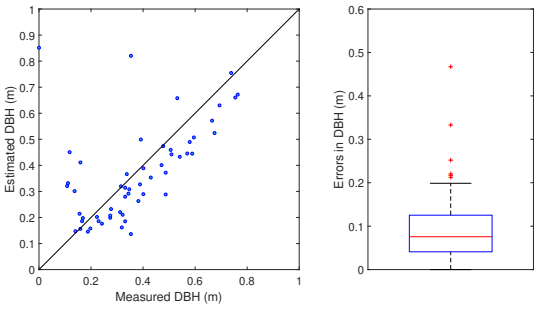

Fig. 10: Estimation of DBH using the designed system. **Left**: Plot of 54 measured trees vs estimated values, one tree was not segmented. **Right**: Boxplot of the errors between the estimated and measured values, RMSE = 0.07 m excluding the outliers and 0.14 m including them.

The results for LiDAR360's forestry module are shown in Fig. 9. The graphical results for the terrestrial LiDAR was similar to that of the mobile LiDAR so the graph is omitted.

D. DBH Comparison with Proposed System

The results for our system are shown in Fig. 10, these results contain data points for every segmented tree (54/55). Three of the trees were poorly segmented which resulted in outlier measurements in the figure. The RMSE of these estimates, not including the outliers is 0.07 m. The RMSE is slightly higher than that of the commercial software, indicating reasonable performance of the system but some room for improvement. Table I summarises the DBH estimates and segmentation results for the proposed system and LiDAR360 using Leica BLK and accumilated mobile LiDAR data.

E. Timing Analysis

To demonstrate the key feature of our presented system, that it can run online, we carried out an evaluation of the

Algorithm	Detections	RMSE (m)	Mode	Data used
LiDAR360	53/55	0.046	Offline	Leica BLK
LiDAR360	55/55	0.056	Offline	Ouster (accumilated)
Proposed	54/55	0.07/0.14	Online	Ouster (live)

TABLE I: Summary of the BDH results for the systems evaluated.

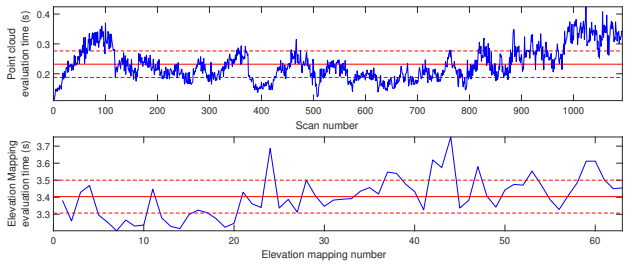

Fig. 11: Evolution of computation time, taken over the 622 sec to walk along the Wytham Woods transect, for the two major components. **Top**: The total time for tree segmentation and fitting. **Bottom**: Time for the elevation map filtering process (computed every 8 m of travel distance). The mean and standard deviation are shown by the red solid and dashed lines respectively.

Process	Tree Tracking	Circle Fitting	Euclidean Clustering	Base Segmentation	Total Time
Time (ms)	88	33	66	20	232
std (ms)	±30	±8	±33	±6	±44

TABLE II: Computation times for the *Tree Tracking* modules.

computation time for each component. Fig. 11 demonstrates the time required for the two major components of the system — (individual) scan processing in *Tree Tracking* and (accumulated) elevation map filtering. A breakdown of individual modules of the *Tree Tracking* pipeline is given in Table II.

The *Tree Tracking* system takes about 0.23 sec (about 4 Hz) to process an individual scan, which is almost two times faster than the output of our 2 Hz LiDAR odometry.

The elevation map filtering process takes 3.4 sec on average. The elevation map is processed every ∼8 m and walking speeds of ∼1 m/s are expected. Therefore, on average, this process takes less than half the time available. Future work aims to optimise this block to run closer to sensor frame rate.

V. Conclusion and Future Work

This paper presented a mobile LiDAR scanning system for the automatic segmentation of trees and the online estimation of their DBH. The architecture of a tree tracking functional block was introduced to facilitate the online operation of this system. We tested the system's performance using data from a natural forest, Wytham Woods, to prove that acceptable results can be achieved online which compares favourably to the performance achieved by a commercial software package conducted in post-processing.

Future work will implement methods within the *Tree Tracking* functional block to overcome long term drift such as loop closure and SLAM pose graph optimisation. We would like to consider adaptive segmentation algorithm such as in [8] to improve performance on the smaller trees.

Finally by more fully using the screen we intend to provide feedback to the operator to better direct their actions during operation.

References

[1] P. Raumonen, M. Kaasalainen, M. Åkerblom, S. Kaasalainen, H. Kaartinen, M. Vastaranta, M. Holopainen, M. Disney, and P. Lewis, "Fast Automatic Precision Tree Models from Terrestrial Laser Scanner Data," *Remote Sensing*, vol. 5, no. 2, pp. 491–520, 2013.

[2] K. Calders, G. Newnham, A. Burt, S. Murphy, P. Raumonen, M. Herold, D. Culvenor, V. Avitabile, M. Disney, J. Armston, *et al.*, "Nondestructive estimates of above-ground biomass using terrestrial laser scanning," *Methods in Ecology and Evolution*, vol. 6, no. 2, pp. 198–208, 2015.

[3] J. Gonzalez de Tanago, A. Lau, H. Bartholomeus, M. Herold, V. Avitabile, P. Raumonen, C. Martius, R. C. Goodman, M. Disney, S. Manuri, *et al.*, "Estimation of above-ground biomass of large tropical trees with terrestrial LiDAR," *Methods in Ecology and Evolution*, vol. 9, no. 2, pp. 223–234, 2018.

[4] P. Raumonen, M. Åkerblom, M. Kaasalainen, E. Casella, K. Calders, and S. Murphy, "MASSIVE-SCALE TREE MODELLING FROM TLS DATA," *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, vol. 2, 2015.

[5] J. Trochta, M. Krůček, T. Vrška, and K. Král, "3D Forest: An application for descriptions of three-dimensional forest structures using terrestrial LiDAR," *PLOS ONE*, vol. 12, no. 5, p. e0176871, 2017.

[6] L. Zhong, L. Cheng, H. Xu, Y. Wu, Y. Chen, and M. Li, "Segmentation of individual trees from TLS and MLS data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 2, pp. 774–787, 2016.

[7] A. Burt, M. Disney, and K. Calders, "Extracting individual trees from lidar point clouds using treeseg," *Methods in Ecology and Evolution*, vol. 10, no. 3, pp. 438–445, 2019.

[8] S. T. Digumarti, J. Nieto, C. Cadena, R. Siegwart, and P. Beardsley, "Automatic Segmentation of Tree Structure From Point Cloud Data," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3043–3050, 2018.

[9] S. Tejaswi Digumarti, L. M. Schmid, G. M. Rizzi, J. Nieto, R. Siegwart, P. Beardsley, and C. Cadena, "An Approach for Semantic Segmentation of Tree-like Vegetation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1801–1807.

[10] H. K. Heo, D. K. Lee, J. H. Park, and J. H. Thorne, "Estimating the heights and diameters at breast height of trees in an urban park and along a street using mobile LiDAR," *Landscape and Ecological Engineering*, vol. 15, no. 3, pp. 253–263, 2019.

[11] V. Pratt, "Direct least-squares fitting of algebraic surfaces," *ACM SIGGRAPH computer graphics*, vol. 21, no. 4, pp. 145–152, 1987.

[12] S. Zhou, F. Kang, W. Li, J. Kan, Y. Zheng, and G. He, "Extracting Diameter at Breast Height with a Handheld Mobile LiDAR System in an Outdoor Environment," *Sensors*, vol. 19, no. 14, p. 3212, 2019.

[13] A. J. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient Organized Point Cloud Segmentation with Connected Components," *Semantic Perception Mapping and Exploration (SPME)*, 2013.

[14] F. Westling, D. J. Underwood, and D. M. Bryson, "Graph-based methods for analyzing orchard tree structure using noisy point cloud data," *arXiv preprint arXiv:2009.13727*, 2020.

[15] D. Wisth, M. Camurri, and M. Fallon, "Robust Legged Robot State Estimation Using Factor Graph Optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4507–4514, 2019.

[16] F. Pomerleau, "Applied registration for robotics: Methodology and tools for ICP-like algorithms," Ph.D. dissertation, ETH Zurich, 2013.

[17] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[18] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS)*. Springer, 2016, pp. 99–120.

[19] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic Terrain Mapping for Mobile Robots With Uncertain Localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.

[20] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for Point-Cloud Shape Detection," *Computer Graphics Forum*, vol. 26, no. 2, pp. 214–226, 2007. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2007.01016.x