Online Forest Mapping and Inventory Generation using Handheld LiDAR

Alexander Proudman[1] Milad Ramezani[2†] Sundara Tejaswi Digumarti[1] Nived Chebrolu[1] Maurice Fallon[1]

***Abstract*— Mobile LiDAR sensors are increasingly being used to scan environments in ecology and forestry applications. However reconstruction and characterization are typically performed offline. Motivated by this, we present a LiDAR based framework, running on a handheld device, that is capable of creating 3D point cloud reconstructions of large forest areas, segmenting and tracking individual trees and creating an inventory in an online manner. Segments of a tree from multiple views accumulated over time are combined and the corresponding tree models are also updated. Providing immediate feedback to the operator via a screen on the device is a key feature of this work as it enables satisfactory coverage of the area being mapped without gaps and missing sections. We employ a pose-graph based SLAM system with loop closure detection to correct for drift errors allowing us map large areas accurately. Multi-session mapping capability is also supported with the ability to automatically merge data captured during different runs in a post-processing step. As an example parameter for the forest inventory, we estimate the Diameter at Breast Height (DBH) of individual trees, in an online manner, by fitting cylinders to detected tree trunks through a least-squares optimization within a RANSAC loop. We demonstrate our mapping approach operating online in two different forests (both ecological and commercial) with a total travel distance spanning several kilometres. Further, our DBH estimates are within ∼7 cm accuracy for 90% of the detected trees in the ecological forest.**

I. Introduction

Monitoring the growth of individual trees in a forest is important to infer arboreal health. In ecology and forestry studies, metrics of growth such as Diameter at Breast Height (DBH) are commonly computed through manual measurements, for e.g. with measuring tapes. Such metrics form the forest inventory and describe the state of the forest.

Over the last decade, 3D LiDAR scanners have increasingly been used to map forests. Automated methods to segment individual trees, construct tree models [1] and extract metrics such as above ground biomass and carbon stock [2], [3] have also been developed. However, these approaches rely on high quality point clouds data captured from expensive and time consuming Terrestrial Laser Scanners (TLS). Furthermore, the analyses are performed in an offline manner or as a post-processing step.

In this work, we present an online forestry mapping and inventory system capable of providing real-time feedback to the operator. We use a hand-held LiDAR setup with onboard compute capability (see Fig. 2) as our scanning device.

[1] Oxford Robotics Institute (ORI), The George Building, 23 Banbury Road, Oxford, UK - OX2 6NN.

[2] CSIRO Data61, Queensland Centre for Advanced Technologies, 1 Technology Court, Pullenvale QLD 4069, Australia.

† The presented work was done when the author was with the ORI.

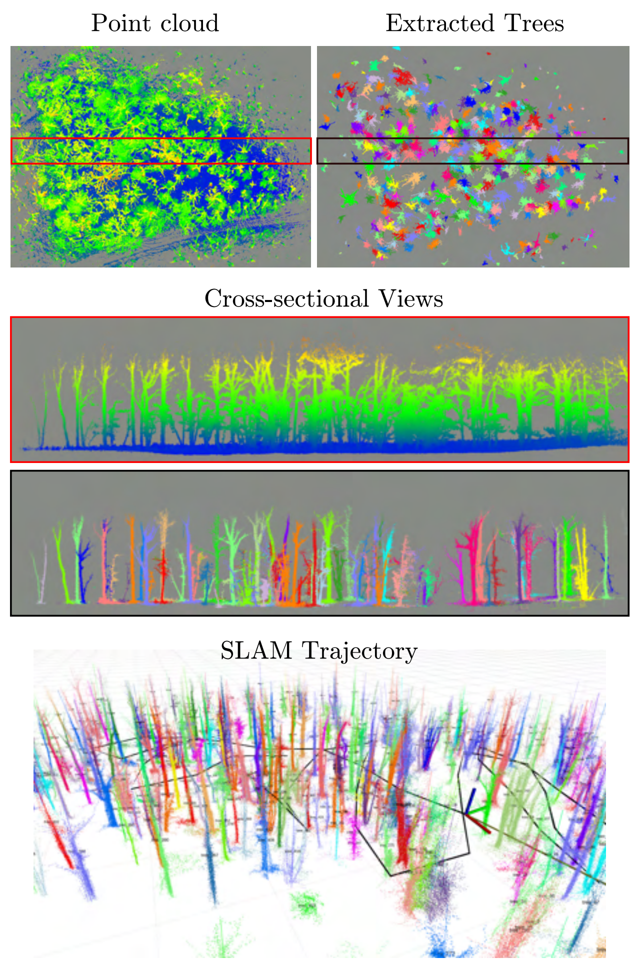

Fig. 1: Visualization of the system output. Mapping results on the Wytham Woods dataset surveyed in a lawn-mower pattern. **Top**: Accumulated point cloud reconstruction and the extracted trees. **Middle**: Cross-sectional views of the clouds shown above. **Bottom**: Zoomed in view of the extracted trees, showing the path of the camera in black.

This facilitates an easier, more flexible and faster surveying experience in comparison to traditional TLS systems.

present an online forestry mapping and inventory system capable of providing real-time feedback to the operator on the ground.

In our previous work [4] we presented a framework for estimating DBH metrics from LiDAR point clouds in an online manner. We showed that these estimates are within ∼ 7cm accuracy for 90% of the detected trees in an ecological forest. Here, we extend this framework with a mapping system that takes into account the challenges of typical forestry surveys. We use an elastic pose-graph structure as the underlying representation, which allows us to scale the map to large areas. We also support multi-session mapping allowing the operator to survey the forest in multiple runs, which may be necessary due to battery constraints or operator fatigue.

The main contributions of this work are as follows:

- An online mapping system for forest environments using a handheld LiDAR with real-time feedback.
- A pose-graph based SLAM system with loop closure integration to correct for odometry drift, and to maintain an accurate tree map over large areas.
- Segmentation, tracking, modelling and estimation of DBH for individual trees in the forest.
- Multi-session mapping capability to merge maps generated from separate runs as a post-processing step.
- Demonstration of the system in challenging large scale forest environments spanning several kilometres.

II. Related Work

Advances in LiDAR technology have led to it being used in non-traditional fields such as ecology, forestry and remote sensing. Here, we review some of the state-of-the-art approaches for tree segmentation from LiDAR point clouds. We first review techniques developed for high end Terrestrial LiDAR scanners followed by more recent approaches developed for mobile LiDAR scanners.

A. Tree Segmentation Using Terrestrial LiDAR

Many existing tree segmentation techniques operate on a point cloud accumulated from multiple static scanning locations using a Terrestrial Laser Scanner (TLS), in post-processing. Raumonen et al. [5] and Trochta et al. [6] 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 [7] utilize concepts from graph theory to find the connectivity between adjacent points and segment each tree. These approaches, however, rely on multiple assumptions about a tree's architecture as well as assuming minimal interconnection between their crowns.

In [8] Burt et al. present a software package named *treeseg* that uses region-growing technique is to segment individual trees. Key features in *treeseg*'s design are its independence of forest type, scanning instrument and no assumptions about the tree structure.

B. Tree Segmentation Using Mobile LiDAR

Most of the above approaches are intended primarily for very high quality TLS point clouds. Their performance drops when applied to noisier point clouds captured from Aerial Laser Scanners (ALS) and ground based mobile laser scanners (MLS). However, the lower cost and increased portability of mobile laser scanners in comparison to TLS outweighs makes them a desirable pick despite the measurement data being noisier. As a result several works have looked at tree segmentation and automatic inventory generation systems for these scanners.

Heo et al. [9] 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 [10]. They emphasised the advantage of using mobile LiDAR to reduce shadow and occlusion effects, which are more prominent with terrestrial LiDAR systems, especially in an urban forest environment. They utilised a Stencil LiDAR system produced by Kaarta[1] which includes a Velodyne sensor.

Similarly, Zhou et al. [11] 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 [12].

Westling et al. [13] 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.

More recently data driven and learning based approaches have also been used for tree segmentation. Digumarti et al. [14] train a random forest classifier to segment individual trees into their component structures. Convolutional Neural Networks (CNNs), trained on simulated tree data are used in [15] to segment trees from colour and depth images. Windrim and Bryson [16] detect trees in point clouds collected using ALS by encoding the point clouds as a 2D raster image and detecting points of high density. This is followed by applying a segmentation network, adapted from *PointNet* [17], to segment trees. Krisanski et al. [18] further extend the idea of learning based tree segmentation and present a sensor agnostic approach that works point clouds of different densities.

The above presented approaches typically work as post-processing techniques applied after data collection. This does not allow the operator to perceive the reconstruction during scanning. Our motivation is to develop a LiDAR-driven technique which can reconstruct point clouds, extract individual trees and estimate their structural parameters in dense forests with rough terrain in real-time. A real-time mapping system can ensure full coverage by providing feedback to the operator, to help identify gaps in the scanned region, enable rescanning and ensure that the environment is fully scanned and processed at runtime.

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

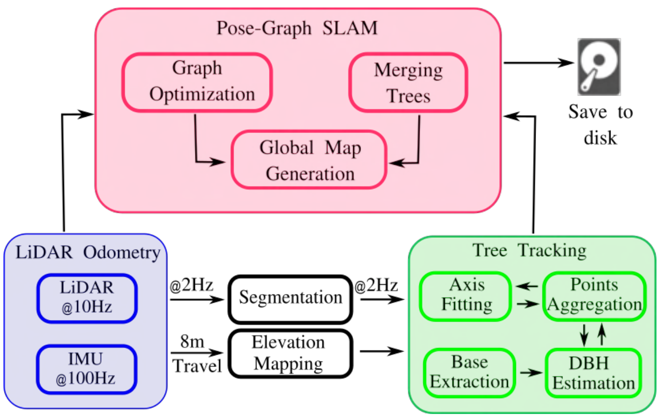

Fig. 3: An overview of the system architecture. The frequencies of communication between blocks are shown.

III. Approach

A. System Description

Our device consists of an Ouster OS0-128 LiDAR and an Intel Realsense D435i(Fig. 2). In this work, we only use data from the LiDAR scanner and its inbuilt IMU as inputs to our odometry and tree tracking systems. The LiDAR has 128 beams and a 90° vertical field of view (FOV). While it cannot completely cover the environment vertically, the 90° vertical FOV is sufficient to identify individual trees within about 20 m of the sensor.

The proposed system architecture is shown in Fig. 3. Due to the system's online nature, the architecture is different from many existing methods, and alterations have been made to parallelise and speed up the overall pipeline.

Advantages of using this online fused data approach are that the success of data collection can be evaluated online through real time feedback, data collection and analysis is quick, and prior inventory data can be used to automatically identify trees.

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

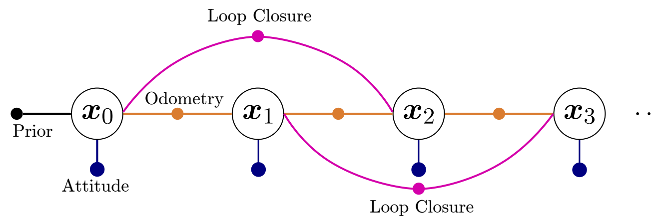

Fig. 4: Pose-graph representation of the SLAM system. Each node represents the pose of the sensor and the edges represent constraints from multiple sources.

B. Localization and Mapping

1) LiDAR Odometry: A LiDAR-inertial odometry module estimates the pose of the sensor. Our system [19] is a factor-graph based windowed smoother which fuses IMU readings with range measurements from the LiDAR. We use IMU pre-integration to remove motion distortion from scans and to initialise Iterative Closest Point (ICP) registration, specifically the implementation of Pomerleau [20]. ICP is used to determine the relative transforms at a frequency of 2Hz. The IMU measurements and the relative transforms form constraints in a factor graph used to estimate poses in a sliding window optimisation utilising iSAM2 [21] (as part of the GTSAM library). Using the gravity information from the IMU, we can align the clouds with the gravity direction. This helps in determining the bases of trees.

2) Pose-graph SLAM with Loop Closures: The estimated odometry is accurate over short distances ($\sim$100 m), but accumulates drift as we travel further. The pose-graph SLAM module corrects for these drifts using loop closure constraints. Fig. 4 shows the pose-graph structure where nodes represent the pose of the sensor, and the edges represent constraints from multiple sources.

New nodes are added to the pose-graph as the device moves through the environment (for every 5 m of travel). Local maps created by accumulating laser scans ($\mathcal{M}_{\mathbf{x_i}}$) are attached to each node ($\mathbf{x_i}$) of the pose graph. The point cloud of each local map is transformed into a co-ordinate system whose origin corresponds to a pose from the SLAM node.

Consecutive nodes in the graph are constrained with LiDAR odometry (shown in orange) When a loop closure is detected on re-visiting a place, a new constraint (shown in pink) is added. By optimizing the entire graph with the loop closure constraints, at regular intervals, pose estimates are corrected for drift.

We detect the loop closures between non-consecutive nodes in the graph based on their distances in the global coordinate frame. Nodes within a distance of 3 m are identified as candidates for a possible loop closure. Local sub-maps associated with these nodes are matched using ICP. If a successful registration is obtained between the two sub-map point clouds, we add a loop closure constraint in the pose-graph given by the transformation estimated by ICP.

Finally, we obtain the combined map of the environment $\mathcal{M}$ by transforming the local sub-maps $\mathcal{M}_{\mathbf{x_i}}$ in to the global frame using the pose of the corresponding node in the pose-

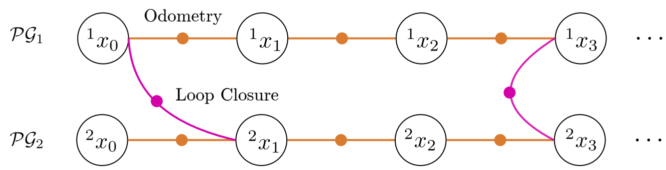

Fig. 5: Mapping over multiple sessions as a post-processing step. Loop closures between the two sessions are computed (in purple) and jointly optimized. A global map is then computed using the jointly optimized poses.

graph.

3) Multi-Session Mapping: While the pose-graph SLAM module allows us to map large areas, forestry survey missions can require multiple scanning sessions. This may be due to the limited battery capacity, operator fatigue or multiple operators scanning different parts of a forest. It is therefore vital for an effective mapping system to be able to merge maps across multiple sessions.

We exploit the pose-graph structures built during individual missions and use them in a post-processing step to enable merging. At the of end of each session, we save the pose-graph as well as the corresponding local maps to the disk. Once all the sessions are completed, we use a place recognition system by Giseop et al. named *ScanContext* [22] to propose loop closure constraints between the different sessions. *ScanContext* computes a compact descriptor for each local map in our pose-graph. The descriptor captures the distribution of the points in the 3D space and creates a discriminative signature for the point cloud. We compute these descriptors for each local map in all of the sessions, and compare them to retrieve a set of loop-closure candidates. The loop-closure candidates correspond to the places where the local maps created during different sessions overlap with each other. Note that our approach assumes that the individual sessions have an overlap with at least one other session in the entire mission.

The final pose-graph then consists of inter-session loop closure constraints as illustrated in Fig. 5. We then optimize the combined pose-graph to obtain the final poses of the missions in a common co-ordinate frame.

C. Tree Segmentation and Tracking

In parallel to the mapping module, we process the incoming LiDAR scans to segment and track trees. Unlike typical forest inventory systems that segment and model trees as a post-processing step, we segment, track and fuse each detected tree in an online manner. Simultaneously we also model the detected trees and compute the DBH metric.

Firstly, Euclidean segmentation is applied on the LiDAR point cloud to roughly segment tree trunks, branches, canopy and shrubbery. To speed up the segmentation and lessen the radial variation of the point density, the cloud is first downsampled using a voxel filter as also done in [8]. The voxel grid thus formed is reformulated into a k-d tree to efficiently find nearest neighbours of every point. These points are clustered together into groups of points which are within some distance threshold of each other.

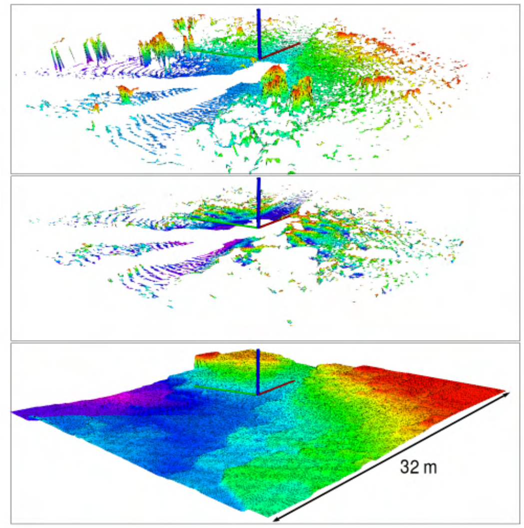

Fig. 6: Filtering process on the elevation map: Original elevation map (top) is filtered to remove spikes (middle). Finally, holes are filled using a morphological closing filter on the grid map (bottom).

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.

1) Elevation Mapping: In order to extract the bases of the trees being tracked, we employ an open-source sensor-centric elevation mapping framework[2] which is built upon a universal Grid Map library [23].

Consistent elevation maps of the environment surrounding the sensor are generated at the same frequency of the LiDAR output (10 Hz). A resolution of 16cm for the grid of 32m $\times$ 32m is used in this work.

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. Slopes of each grid cell are computed and cell with slopes greater than a threshold are removed, resulting in a smooth terrain with a few holes that are filled using a morphological closing filter. Fig. 6 shows this process.

Applying the chain of filters is relatively time consuming so we update the elevation map every 8 m of travel distance based on the state estimate from our LiDAR odometry, to allow online operation.

2) Tree Tracking: We formally define a tree using a 'tree descriptor' (t) comprising of the following elements.

- A unique id,
- Major axis of the tree incline ($\mathbf{I}$, a vector): the line that best fits a subset of the tree's points in a least squares sense,
- Diameter at Breast Height (DBH) (D),
- 3D position of the tree base ($\mathbf{b}$): the point from the most recent elevation map closest to the major axis,
- The minimum and maximum height of the point cloud defining the tree $(min$ & $max)$.

Every tree descriptor is derived from a corresponding point

[2]https://github.com/ANYbotics/elevation_mapping

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. [9] indicated. 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 $0 \leq i \leq n$.

Every segmented cluster of points, is either assigned to an existing tree or to a new tree. If assigned to an existing tree, the descriptor is re-evaluated after merging the points of the cluster to those of the tree. We also discard clusters that do not satisfy empirically determined characteristics ($\theta_{\text{threshold}}$, $h_{\text{threshold}}$) of a tree. These are:

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

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.

D. Estimation of Diameter at Breast Height (DBH)

To estimate the DBH of each individual tree, we employ a cylinder fitting procedure [10] commonly used in literature. Utilising the method described by Zhou et al. [11] and Heo et al. [9], 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 centred 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. RANSAC circle fitting is then used on the set of resultant 2D points, which is robust to potential outliers from errors in segmentation [24].

IV. Experimental Evaluation

We evaluated our system at two locations. The first is an ecological forest called Wytham Woods in Oxford, UK. The woods are spread over 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. The second location for our evaluation is a commercial pine-tree forest one hour north of Helsinki, Finland spread over several kilometers. These two locations differ in the type and densities of trees as well as different profiles of the underlying terrain.

Evaluation of the accuracy of the LiDAR odometry module, DBH estimation module, and timing analysis are presented in our previous work [4]. Here we focus present results of large-scale mapping and multi-session mapping capabilities of our framework.

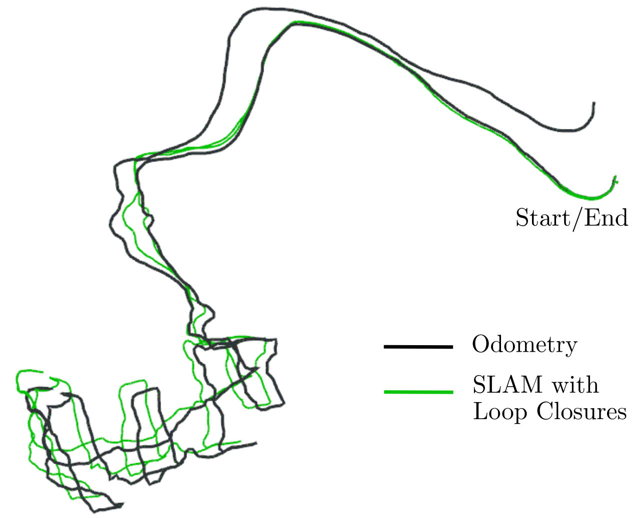

Fig. 7: Effect of loop closures on the estimated trajectory of the device. Odometry estimate (in black) and SLAM pose estimate with loop closures (in black) for a 1.3 Km long trajectory. The odometry estimate drift can be clearly seen at the start/end position. The drift accumulated for this trajectory is around 25 m.

Fig. 8: Pose-graphs from individual sessions visualized in different colors. Loop closures both with-in the same session, and in-between different sessions in shown in red. The total travel distance of all the 4 km.

A. Large-scale Mapping with Loop Closures

Fig. 1 shows the mapping results from the Wytham woods dataset. This dataset covers an area of about 0.5 ha traversed in a lawn-mower pattern with a spacing of about 10 m. Several loop-closures have been detected towards the end of each row allowing the SLAM system to correct for drift error. The figure also shows the point cloud reconstruction and the extracted trees. The terrain profile of the forest is visible in the cross sectional view.

We highlight the effect of the loop closures on the pose estimation in Fig. 7. The estimated trajectory using only LiDAR odometry (in black) accumulates a drift of 25 m in comparison to the SLAM trajectory which uses loop closures (in green) over a total trajectory length of 1.3 Km.

B. Mapping over Multiple Sessions

Data in the commercial pine forest in Finland was recorded over three sessions, mapping areas with increasing density of trees in each session. The operator starts and ends at the

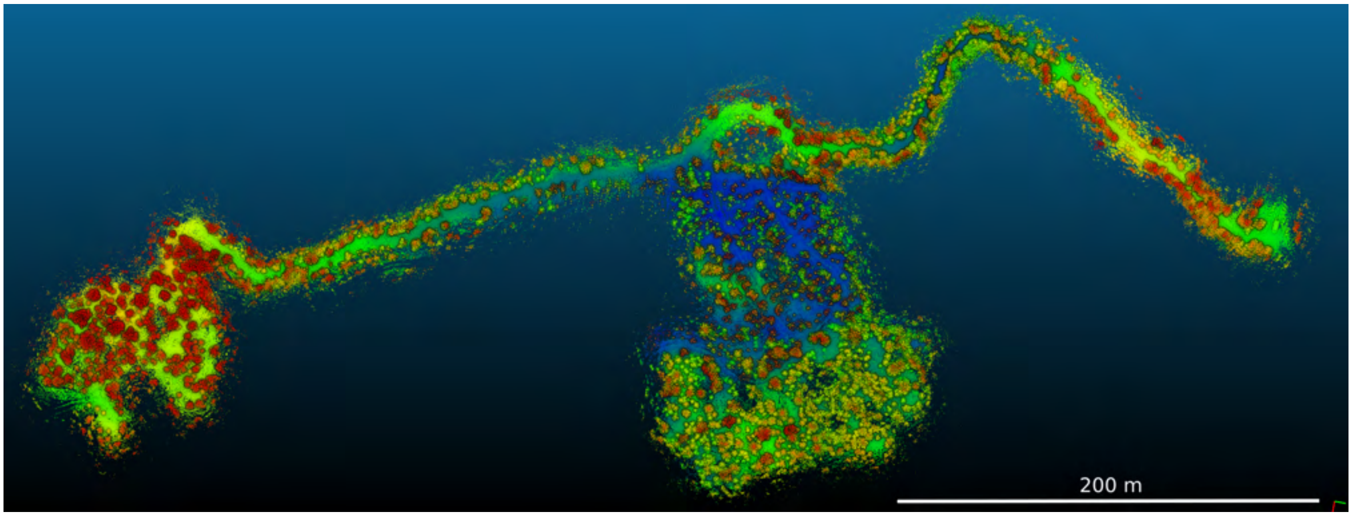

Fig. 9: Mapping results on a large-scale dataset from a commercial pine tree forest in Finland. Figure shows the point cloud reconstruction of the forest obtained by merging all the local maps after joint pose-graph optimization.

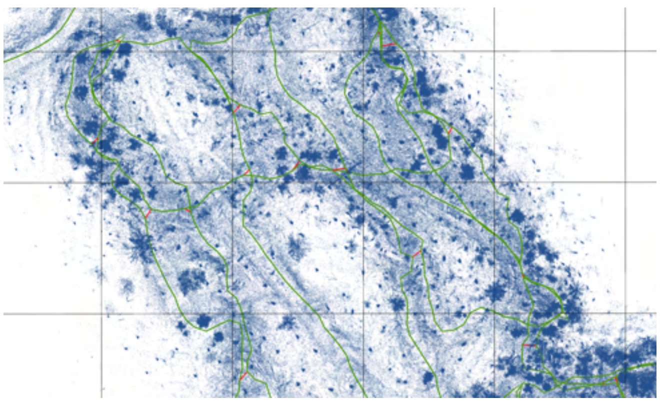

Fig. 10: Zoomed-in view of a portion of the forest shown in Figure 9 showing dense clusters corresponding to individual trees. The trajectory traversed (green) and the detected loop closures (red) are also shown.

same position and walks along an access road to a plot of trees, in all the sessions, as shown in Fig. 8. The total travel distance of the combined missions is over 4 kms with each session being roughly equal in length. Note that the three sessions have overlapping areas which is recognized by the loop closure module. An example of such a loop closure is shown in Fig. 11. The point cloud reconstruction of the forest obtained by merging the local maps of the three sessions after joint pose-graph optimization is shown in Fig. 9.

This dataset consists of several challenging elements, including long corridor like passages between rows of trees, dense forest patches with limited LiDAR range due to occlusions, transitions in between areas with different densities. Further, as the dataset is captured with a human operator walking on rough surface, the sensor module is often subject to shaking and fast motions. Despite these challenges, our mapping system was successfully able to map each of sessions as well as to merge them together in a post-processing step.

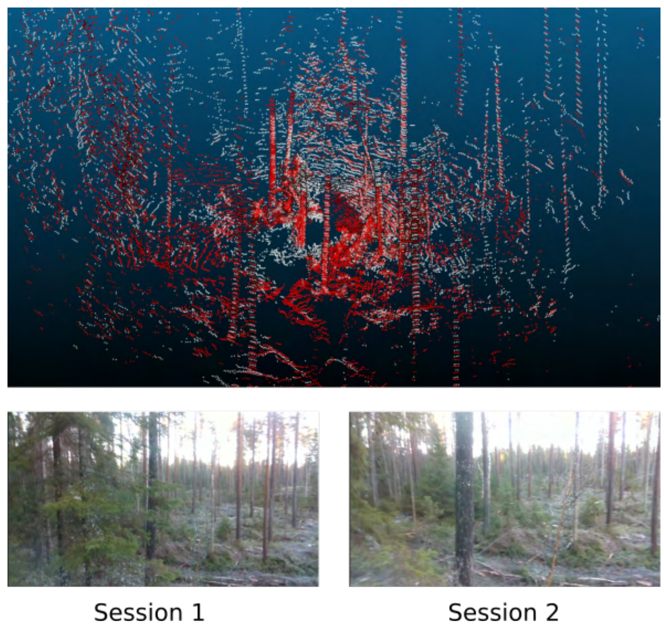

Fig. 11: Example of a loop closure detected between Session 1 and Session 2. **Top**: The corresponding point clouds overlaid on each other after ICP registration. Point cloud from Session 1 is shown in red whereas white points are from Session 2. **Bottom**: Corresponding camera images of the loop closure candidates are shown. Note that the loop closures are detected only from the LiDAR data, and the images are shown for illustrative purposes only.

The performance of the *ScanContext* loop proposal mechanism was impressive. Despite being originally developed for automotive localisation, it was still successful in our forestry application. We noticed that loops could be found in the forests where the separation between trees was managed. Performance degraded in unmanaged forests where trees were typically smaller samplings and with little clear separation. A further study would be necessary to properly evaluate place recognition algorithms in these types of environments.

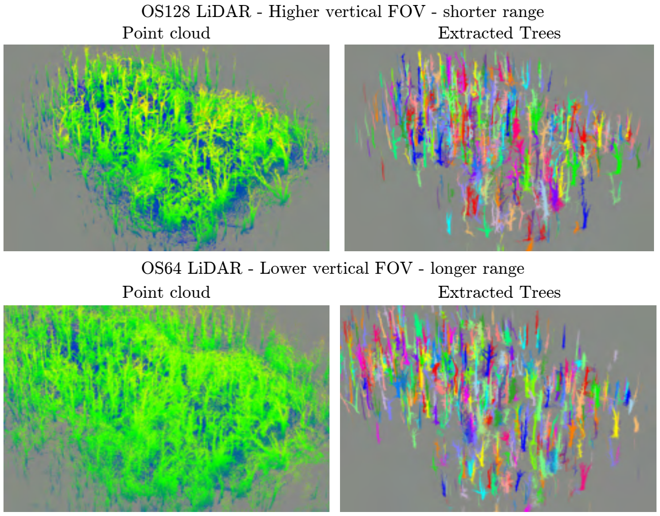

Fig. 12: Reconstruction and tree extraction results using LiDAR scanners with different sparsity levels. Top: 128 beam LiDAR resulting in a denser point cloud reconstruction. Bottom: 64 beam LiDAR resulting in a sparser point cloud reconstruction. The proposed tree extraction approach is robust to different sparsity levels.

C. Mapping with Different Sparsity Levels

In this section, we show that our mapping and segmentation pipeline can deal with different sparsity of LiDAR without additional tuning. To test this capability, we recorded datasets using Ouster LiDAR OS0-128 and Ouster LiDAR OS1-64, which are 128 beam and 64 beam LiDARs respectively. As seen in Fig. 12, OS0-128 provides a much denser point cloud in comparison to OS1-64. The two sensors also have different vertical field of views (FOVs) and maximum ranges. The experimental results suggest that a narrow FOV and range are sufficient for our application of estimating DBH and the motion from odometry. Our algorithms have a voxel filtering step at the beginning which ensures that the subsequent steps are robust to sensor changes.

V. Conclusion

This paper presented a mobile LiDAR scanning system for mapping large forest areas, 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 well studied ecological forest, Wytham Woods, UK as well as a commercial forest in Finland 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.

We currently only extract tree trunks and compute the DBH metric. In the future, we would like to extend the breadth of metrics used to characterise trees. In particular, we would like to quantify information on the branching of the tree, as well as properties related to the tree crown. We plan to integrate information from the cameras as well as extract semantics to estimate a richer set of quantitative and qualitative metrics for the forest inventory.

One of the limitations identified in the experimental analysis was the downgraded performance on small trees. We like to address this issue by considering a adaptive segmentation algorithm such as in [14] 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.

Acknowledgments

This research is supported by the UKRI/EPSRC ORCA Robotics Hub (EP/R026173/1). M. Fallon is supported by a Royal Society University Research Fellowship. The authors would like to Taneli Vuornos (Silvere) for the collection of the data used in Sec. IV-B.

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] A. Proudman, M. Ramezani, and M. Fallon, "Online estimation of diameter at breast height (dbh) of forest trees using a handheld lidar," in *2021 European Conference on Mobile Robots (ECMR)*. IEEE, 2021, pp. 1–7.

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

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] V. Pratt, "Direct least-squares fitting of algebraic surfaces," *Proc. of the Intl. Conf. on Computer Graphics and Interactive Techniques (SIGGRAPH)*, vol. 21, no. 4, pp. 145–152, 1987.

[11] 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.

[12] 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.

[13] 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.

[14] 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 (RA-L)*, vol. 3, no. 4, pp. 3043–3050, 2018.

[15] 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 *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019, pp. 1801–1807.

[16] L. Windrim and M. Bryson, "Detection, segmentation, and model fitting of individual tree stems from airborne laser scanning of forests using deep learning," *Remote Sensing*, vol. 12, no. 9, p. 1469, 2020.

[17] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[18] S. Krisanski, M. S. Taskhiri, S. Gonzalez Aracil, D. Herries, and P. Turner, "Sensor agnostic semantic segmentation of structurally diverse and complex forest point clouds using deep learning," *Remote Sensing*, vol. 13, no. 8, p. 1413, 2021.

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

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

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

[22] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.

[23] 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.

[24] 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.