Digital Analytics and Robotics for Sustainable Forestry

CL4-2021-DIGITAL-EMERGING-01
Grant agreement no: 101070405

DELIVERABLE 4.2
Report describing map server — for robot and base station mapping

Due date: month 18 (March 2023)
Deliverable type: R
Lead beneficiary: UOXF

Dissemination Level: PUBLIC

Main author: Nived Chebrolu, Matias Mattamala, Maurice Fallon

Contents

1 Introduction

The goal of this deliverable is to report on the map server, which is responsible for merging maps from multiple robots and over multiple missions. This deliverable builds on the work done in D3.1 and D4.1. In D3.1, the pose graph map representation is described as a common SLAM framework for all the platforms in the project. D4.1, on the other hand, describes the map payload format and software API guidelines for accessing and retrieving the data for downstream processing. The report showcases the results from trials conducted in Stein am Rhein, Switzerland and Evo, Finland. Additionally, it presents the results from a demo application that involves a robot localizing against a prior map and visualizing data from the digital twin in an online fashion.

2 Multi-Session Map Merging

In this section, we discuss the map server, which plays a crucial role in merging maps from multiple missions. Each mission is represented as a pose graph map, as defined in D3.1. The primary challenge in this process is to establish loop closures between maps from different missions. To address this challenge, we evaluated various place recognition models, including both hand-crafted and learned models.

2.1 Evaluation of place recognition models

The evaluation of place recognition models involved analyzing precision-recall curves and heatmap visualizations of descriptor distances. The precision-recall curves provide insights into the system's ability to correctly identify loop closures, while the heatmaps illustrate the similarity between scans and the discriminative power of different models.

We evaluated the descriptors of four different place recognition models (Logg3dNet [4], EgoNN [3], ScanContext[2], STD [5]) focusing on their ability to accurately capture loop candidates in forest environments. Logg3dNet and EgoNN models are learning-based methods and were pre-trained on the Wild-Places dataset, whereas ScanContext and STD are handcrafted methods. To evaluate the performance of these models, we used data collected from four different forests: Evo, Stein am Rhein, Wytham Woods, and Forest of Dean, UK. These forests were chosen to represent a variety of forest environments, covering conifer, mixed, and deciduous forest types. The ground truth loop closures were obtained by exhaustively matching against scans within a radius of 30 m in an offline step and verifying them using an ICP-based inliers check.

Our experiments measured the precision-recall curves to assess how well each model detected correct loop candidates within a reasonable distance threshold (here set to 10 m). From our obtained precision-recall curves Fig. 1, it is evident that Logg3dNet consistently outperforms the other models across the four different forests. Particularly, on the Evo and Stein am Rhein datasets, Logg3dNet showed the best performance both in terms of precision and recall, without experiencing any drastic drops in precision. In contrast, ScanContext demonstrated a significant decrease in precision, attributed to its dependency on the vertical field of view of the input scan. In more challenging scenarios, such as Wytham Woods, handcrafted models showed a notable decline in performance. However, Logg3dNet remained at the top, successfully retrieving a substantial portion of correct loop candidates, achieving a 70% precision at a 50% recall rate.

To further analyze the distinctiveness of each descriptor, we measured the descriptor

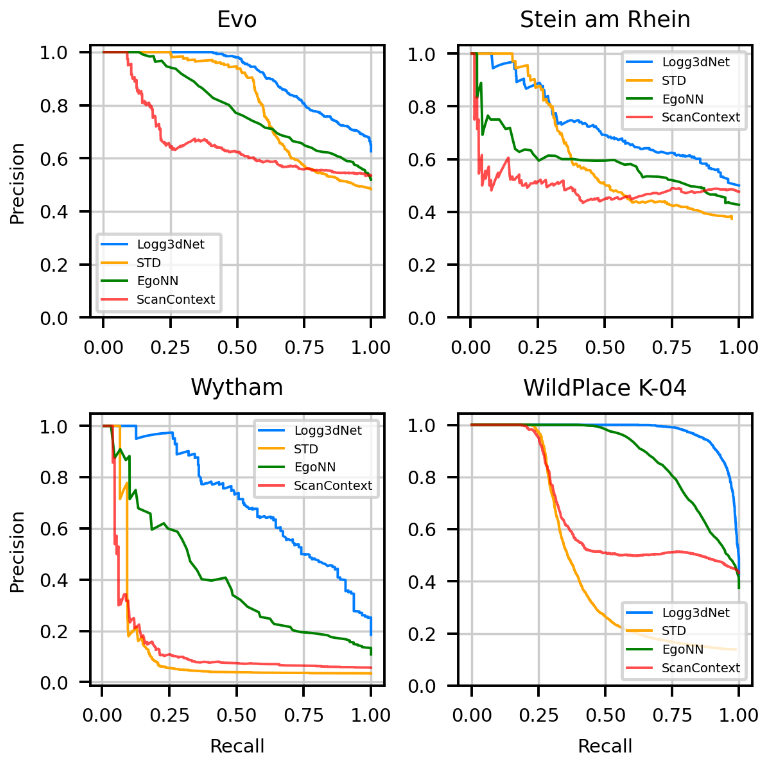

Figure 1: Multi-session map merging using factor graph optimization. Loop closures are detected between maps from different missions and added to the factor graph as constraints. The factor graph is then optimized to obtain the merged map.

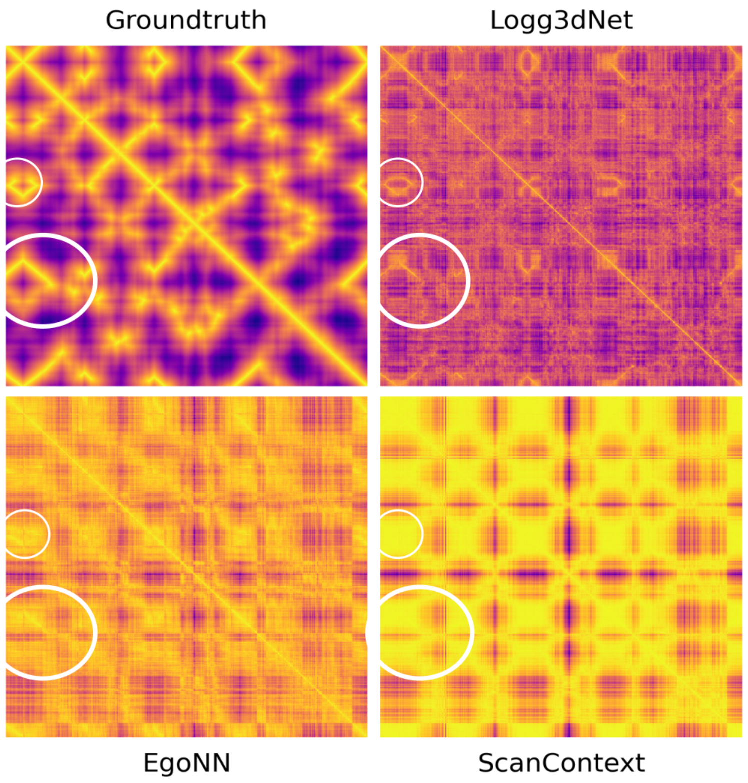

Figure 2: Heatmaps depicting descriptor distances for the Evo dataset. Yellow hues denote a high descriptor similarity between scans, whereas purple indicates low similarity. Patterns more closely resembling the ground truth (top-left) indicate better descriptor performance. Logg3dNet descriptors show the most similar patterns, whereas ScanContext descriptors are the least discriminative among these models.

distances between all query and database descriptors. This is shown in Fig. 2 as a heatmap, which provides a visual representation of the discriminative potential of each descriptor. Consistent with the precision-recall curves, Logg3dNet descriptors exhibited higher similarity with the ground truth heatmap as observed in the highlighted areas, indicating a high true-positive rate and low false-positive rate, respectively. This implies that Logg3dNet descriptors can effectively detect corresponding loop candidates during revisits, whereas EgoNN and ScanContext tend to be less discriminative, often returning numerous false-positive candidates. Based on this evidence, we chose Logg3dNet as main the place recognition method for the rest of the experiments.

Based on the evaluation results, we developed a system that efficiently performs place recognition and loop-closure detection using the logg3dnet model. This system has shown promising performance in merging maps from different missions.

2.2 Methodology and Pipeline Description

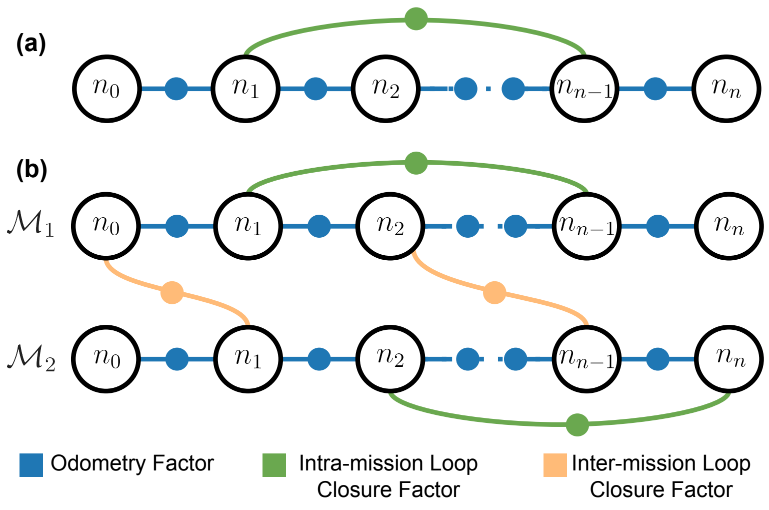

Figure 3: Multi-session map merging using factor graph optimization. Loop closures are detected between maps from different missions and added to the factor graph as constraints. The factor graph is then optimized to obtain the merged map.

The pose graph framework for multi-session map merging is illustrated in Fig. 3. Each SLAM mission output is represented as a pose graph, where nodes correspond to robot poses and edges represent odometry constraints. Our map merging framework is agnostic to the source of the SLAM system, as long as it outputs a pose graph according to the conventions specified in D3.1.

The main task of the map server is to obtain loop closures between maps from different missions. To achieve this, we use the place recognition system to identify potential loop candidates. These candidates are then verified in a multi-stage approach consisting of checks based on the descriptor distances, geometric consistency, and temporal consistency. All these steps are performed within the *place recognition and the verification server* as shown in Fig. 4.

After the multi-stage verification, we add the constraints between pose graphs representing multiple missions. Finally, the graph is optimized to obtain the merged map.

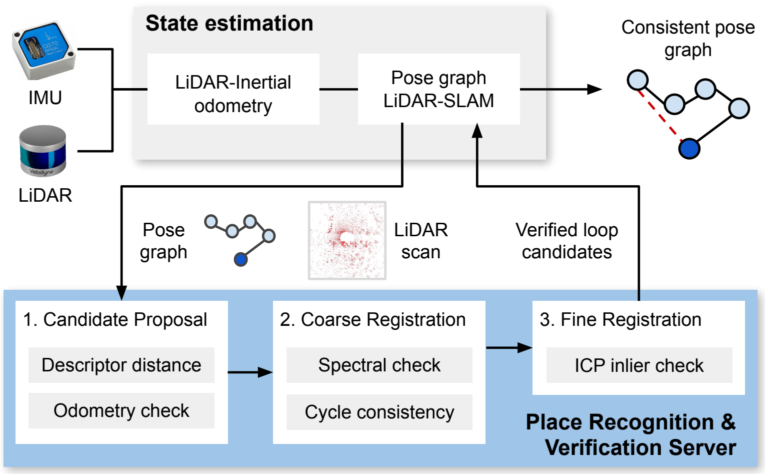

Figure 4: Pipeline for multi-session map merging. The place recognition module consists of three steps: loop candidate proposal, coarse registration, and fine registration. We verify the loop candidates at the global descriptor level, local feature consistency -level, and finally at the fine registration level. A loop candidate is integrated into the pose graph only if it passes these three stages.

2.3 Testing and Evaluation on Real-World Datasets

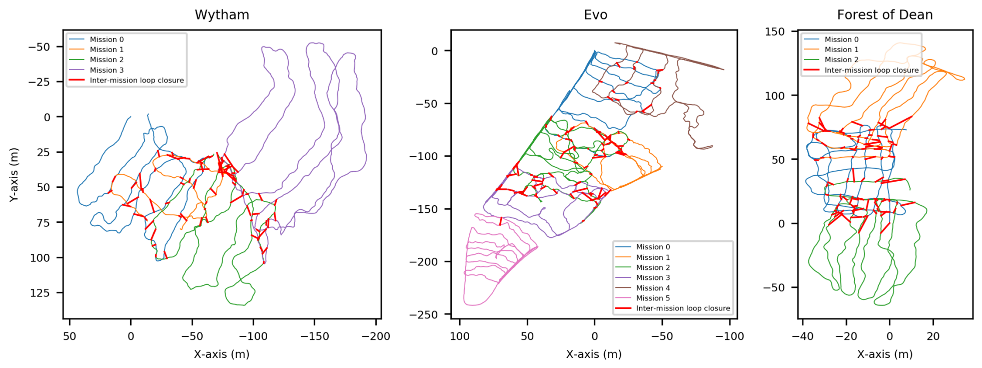

Figure 5: Offline multi-mission SLAM. Left: Wytham - a densely wooded area with uneven terrain, including hills. Center: Evo - featuring a LiDAR setup on an incline, with loop closures occurring primarily when viewpoints are closely aligned. Right: Forest of Dean - flatter terrain compared to Wytham, with a sparser plantation, allowing more frequent loop closures within intersecting areas.

As a part of our tests, we showcase the ability of the map server to obtain loop closures between different mapping missions and to merge those missions into a common map. Figure 5 presents the results of merging different sequences within three different datasets: Wytham Woods, Evo, and the Forest of Dean. Each individual mission covers approximately one hectare, with merged map areas ranging from three to five hectares. We observed that there were more frequent *inter-mission* loop closures in the Forest of Dean compared to Wytham in overlapping areas. This can be attributed to the higher tree density, foliage, and vegetation present in Wytham, making the descriptors less distinctive.

Further, we tested the robustness of our system on the Evo multi-mission dataset, where the XT32 LiDAR was placed at a 45° inclination aimed at capturing the forest canopy. Despite the asymmetry in point clouds introduced by this inclination change, which primarily captured points in the forward direction, our approach successfully identified loop closures and achieved multi-mission map merging.

Overall, our experiments showed potential for efficient large-scale mapping, as we were not required to start at the open access roads nor follow exactly the same paths to achieve loop closures between inter-missions. We also built the map incrementally, one section at a time, hence overlapping areas could only be guaranteed for subsequent missions.

Overall, the map server and the developed system for efficient place recognition and loop-closure detection have shown promising results in merging maps from multiple missions. These findings contribute to the overall goal of creating a robust and comprehensive multi-robot map-sharing framework.

3 Aerial-Terrestrial Map Merging

Merging maps from multiple missions is a challenging task, but merging maps from multiple robots, specifically aerial and ground robots, presents even greater challenges. This is due to the large differences in viewpoints between aerial and ground robots. To address this, we propose a system that leverages tree positions to establish constraints between aerial and ground robots.

In this section, we present our aerial-terrestrial co-registration pipeline, shown in Fig. 7. The inputs are an aerial point cloud in UTM frame (*ALS cloud*), and terrestrial scans from MLS (*MLSclouds*). The MLS clouds can be in either a *pose graph format* or a*tile format*.

In *pose graph format* the MLS clouds are represented by a SLAM pose graph in the UTM frame with local terrestrial point clouds attached to each pose. The clouds—which we name *data payloads*—are obtained by temporally aggregating LiDAR scans at sensor rate (20 Hz) within a fixed distance; an example is shown in Fig. 6 (a). The data payload is converted from a local coordinate frame to the UTM frame using the GNSS measurements acquired by the onboard receiver. This pose graph format enables fine-grained aerial-terrestrial alignment by exploiting the temporal information about how the data was acquired.

In *tile format* we consider the full, single cloud obtained by joining all the data payloads acquired over a mission and then partitioning the data into a fixed-size, gravity-aligned 2D grid to produce a collection of *tiles*. We use a tile is of 20 m × 20 m in this work, as shown in Fig. 6 (b). This representation is independent of the MLS device, and does not require access to the SLAM information—only the relative transformation between the tiles specified by the grid.

3.1 Methodology and Pipeline Description

In the aerial-terrestrial matching process, we find the relative transformation between each MLS cloud and the aerial cloud, providing global constraints for the pose graph optimization procedure. We first pre-process the data by cropping the ALS cloud around the neighborhood of the MLS cloud and correcting misalignments in height between ALS and MLS clouds. Next, we extract features, i.e. location of tree trunks, both from aerial and terrestrial data, to find correspondences between the

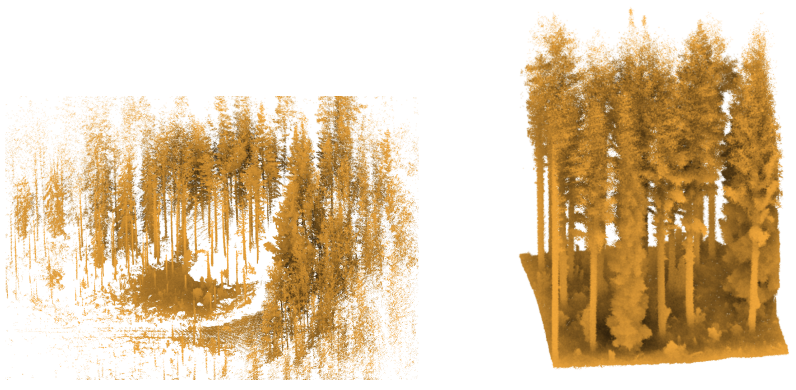

Figure 6: We consider two formats for the MLS data. The *pose graph format* is defined by a pose graph and a collection of data payloads (a), obtained by temporal aggregation of consecutive LiDAR scans; payloads cover a larger area but are sparse and lack canopy points. The *tile format* represents the MLS mission as a collection of tiles (b), obtained by aggregating all the mission scans into a single, dense cloud and partitioning them into a fixed-size grid.

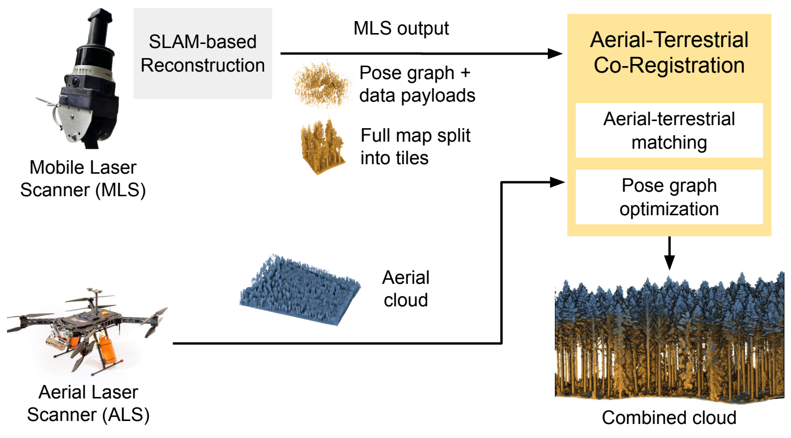
SLAM-based Reconstruction
MLS output
Pose graph + data payloads
Full map split into tiles
Mobile Laser Scanner (MLS)
Aerial-Terrestrial Co-Registration
Aerial-terrestrial matching
Pose graph optimization
Aerial cloud
Aerial Laser Scanner (ALS)
Combined cloud

Figure 7: The aerial-terrestrial co-registration system combines the aerial clouds from ALS, with a terrestrial reconstruction obtained by means of LiDAR-Inertial odometry and pose graph SLAM system.

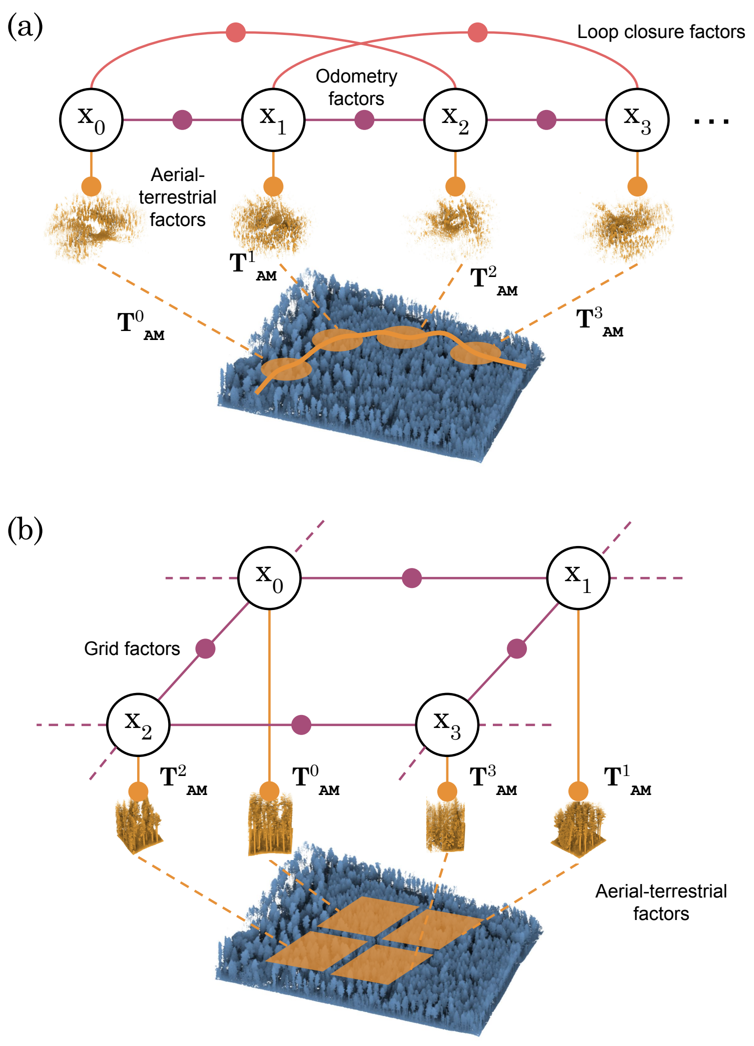

Figure 8: The factor graph formulation for aerial-terrestrial registration for our two data formats. Top: For pose graph/payload optimization we have standard odometry (purple) and loop closure factors (red) to which we add individual aerial-terrestrial prior factors (in orange). Bottom: For the tile format, the grid representation forms local constraints (purple) and the individual tiles are registered to add aerial-terrestrial prior factors (in orange)

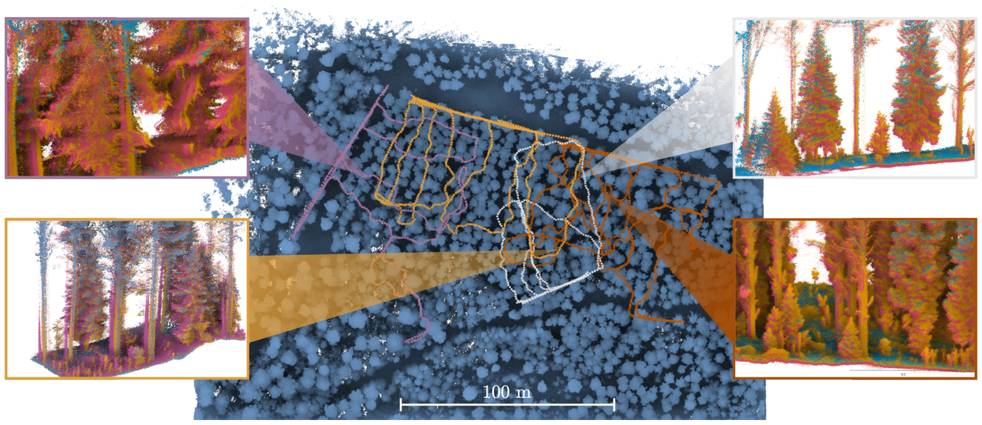

Figure 9: Illustrative examples ALS and MLS point cloud co-registrations using our proposed pipeline for four SLAM missions in Evo, Finland. MLS clouds are shown before optimization (pink), after pose graph refinement (orange), and the aerial cloud (blue). Discrepancies observed before and after the alignment highlight the corrections introduced through the addition of aerial-terrestrial constraints to the pose graph structure.

two clouds. For the aerial point cloud, we generate a canopy height map (CHM) and select the maximum height of tree peaks to identify stem locations, which serve as features for aerial map matching. Whereas, for the terrestrial point cloud, we extract points up to 5 m above the ground plane and apply a density-based spatial clustering algorithm (DBSCAN) to identify tree stem clusters. We then fit cylinders within a RANSAC loop for each cluster to obtain the principal axis and center position of each tree, which serve as features for the MLS cloud. To establish correspondences, we employ a maximum clique algorithm inspired by Bailey et al. [1]. We then use the correspondences to compute the relative transformation between the MLS and ALS clouds. This transformation is added as a constraint to the factor graph optimization procedure, which is then optimized to obtain the merged map. The complete pipeline is shown in Fig. 7.

While the relative transformation between the ALS and MLS is sufficient to merge the two maps, we aim for a finer registration between the two point clouds by incorporating the spatial structure of the SLAM via a pose graph optimization formulation. Fig. 8 illustrates the factor graph formulation for aerial-terrestrial registration for both the pose graph and tile formats. The optimization jointly minimizes the registration cost by incorporating structure factors representing odometry and loop closures along with aerial-terrestrial constraints imposing relative transformations between MLS and ALS clouds. This process results in a globally consistent map that aligns the aerial and terrestrial point clouds over large areas.

3.2 Testing and Evaluation on Real-World Datasets

We tested and evaluated our aerial-terrestrial registration approach using data collected from two field campaigns conducted in Finland and Switzerland. The first campaign took place in Evo, Finland, in a Southern boreal forest characterized by conifers consisting mostly of pines and spruces. ALS point cloud data was collected using an Avartek Bower drone covering up to 730 m $\times$ 540 m. MLS data was captured by a backpack-carried Hesai XT32 LiDAR, covering areas above a hectare.

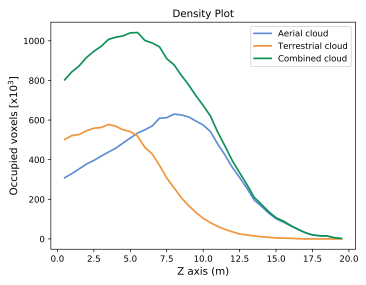

Figure 10: Completeness analysis. Voxel occupancy (with 5 cm resolution) at different heights for the aerial, terrestrial, and co-registered point clouds. The co-registered point cloud shows a higher occupancy through the height of the tree.

The second campaign was in Stein am Rhein (SaR), Switzerland, in a mixed forest. The aerial data was collected by a DJI M600 drone carrying a Velodyne HDL-32E LiDAR and an RTK GNSS receiver in a single mission covering 1850 m $\times$ 500 m. For the MLS we used the same setup as in Evo, and we collected data in four different missions, covering about 1 ha each.

Fig. 9 shows a large-scale example of the co-registrations produced by our method across multiple missions in Evo, Finland. We register four MLS missions to an aerial map covering an overall area of around 4 ha. Each MLS mission trajectory is represented by distinct colored lines overlaid on the ALS point cloud (in blue). The cross-sectional views at different points on the map highlight the precise alignment accomplished through our approach. The consistency in trunk reconstructions along the length of trees suggests a high level of accuracy in co-registration.

3.3 Accessing Completeness of Co-Registered Point Clouds

Lastly, we evaluated the benefits of co-registering terrestrial and aerial clouds in terms of *completeness*. We quantify cloud completeness through density measurements across various height intervals, utilizing a voxelized representation to ensure occupancy evaluation rather than relying solely on raw point density, which may vary depending on the LiDAR type and density. We achieved this by voxelizing the point cloud with a resolution of 5 cm for the ALS, MLS, and combined clouds. Fig. 10 shows the results obtained for the Evo dataset (Plot 1). The results shown in Fig. 10 correspond to the Evo dataset (Plot 1). The ground truth height measurements for the trees in the plot were obtained through manual measurements made using a laser rangefinder. On average, the trees in this plot extend to a height of 20 m, as per ground truth measurements, aligning with the upper limit of point detection in the combined point cloud as observed in Fig. 10.

We also demonstrate the benefit of combining point clouds for tree trait estimation by plotting the probability density functions for tree height and canopy volume derived from the MLS, ALS, and co-registered point clouds. In Fig. 11, we observe a shift to the right in the probability density plot for the combined point cloud (green) compared to the MLS (orange) and ALS (blue) point clouds. This shift indicates a greater height and canopy volume captured by the combined dataset, emphasizing the importance of merging ALS and MLS point clouds for accurate forest inventory and monitoring.

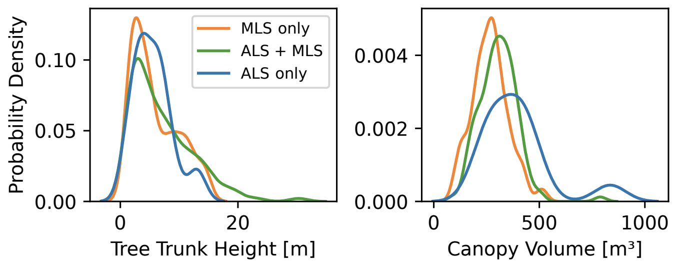

Figure 11: Probability density functions for tree height and canopy volume. The combined point cloud (green) shows a shift to the right, indicating greater height and canopy volume captured compared to MLS (orange) or ALS (blue) only.

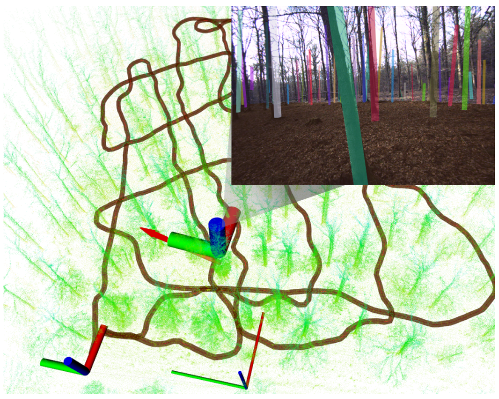

Figure 12: Demonstration of relocalization capability. The LiDAR sensor, illustrated by the thick frames, is relocalized in a prior map. We rendered a virtual view of the forest digital map synchronized with images from our camera (right).

4 Application: Inspection and Monitoring with ANYmal

In addition to merging maps in an offline manner, we have also developed a system for online localization against a previously mapped area. This system allows for real-time visualization of the data from the digital twin. We have demonstrated the system in action in a teleoperated experiment with the ANYmal platform, where it successfully localizes against a prior map in the Forest of Dean and renders the data in an online fashion. Fig. 12 shows a visualization of the system where the sensor is relocalized in a prior map and a virtual view of the forest digital map synchronized with the live image from the camera. This capability provides an intuitive and interactive way to explore and interact with the digital twin, enhancing the overall user experience. This application demonstrates the potential of our system to support forestry tasks such as forest inventory or inspection.

In Figure 13, we illustrate how such a capability can be used both online for real-time reconstruction of trees running on a mobile robot walking through a forest as well as retrieve tree traits from an existing database created earlier. In this example, the robot walks on a test plot of around 0.7 ha and retrieves tree traits from the database in real-time, as well as extends the digital twin with new trees as they are detected in the new mission.

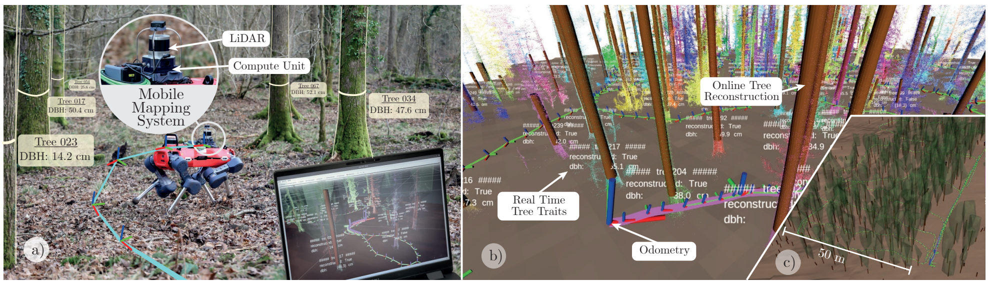

Figure 13: The online pipeline for real-time reconstruction of trees running on a mobile robot walking through a forest. While acquiring data, a forester can evaluate the mapping process in real-time to evaluate coverage and reconstruction quality. The pipeline is able to reconstruct important tree traits online.

5 Summary

This deliverable presents details of the map server capabilities, focusing on merging maps from multiple robots and missions. Leveraging the groundwork laid in previous deliverables D3.1 and D4.1, we provide a system for merging maps in a robust manner tackling challenges of obtaining loop closures in forest environments and dealing with large viewpoint differences when merging data from multiple platforms. We conduct rigorous evaluation across multiple forests in Switzerland, Finland, and the UK which showcases the robustness of the developed systems.

Furthermore, the deliverable introduces an innovative approach to aerial-terrestrial map merging, addressing the challenges posed by different viewpoints of aerial and ground robots. We evaluated the approach on real-world datasets from Finland and Switzerland, the proposed system demonstrates remarkable accuracy in aligning aerial and terrestrial point clouds, providing the way for integration of multi-platform data sources.

The report concludes with an exploration of practical applications, showcasing the inspection and monitoring capabilities demonstrated with the ANYmal platform. By enabling users to conduct online localization against prior maps and visualize digital twins in real-time, the developed systems provide intuitive tools that empower users to enhance situational awareness and make informed decisions in complex environments.

References

[1] Tim Bailey et al. "Data Association for Mobile Robot Navigation: A Graph Theoretic Approach". In: *IEEE Int. Conf. Robot. Autom.* IEEE, 2000, pp. 2512–2517. DOI: 10.1109/ROBOT.2000.846406. URL: https://doi.org/10.1109/ROBOT.2000.846406.

[2] Giseop Kim and Ayoung Kim. "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map". In: *IEEE/RSJ Int. Conf. Intell. Robots Syst.* 2018.

[3] Jacek Komorowski, Monika Wysoczanska, and Tomasz Trzcinski. "EgoNN: Egocentric Neural Network for Point Cloud Based 6DoF Relocalization at the City Scale". In: *IEEE Robot. Autom. Lett.* 7.2 (2022), pp. 722–729. DOI: 10.1109/LRA.2021.3133593.

[4] Kavisha Vidanapathirana et al. “LoGG3D-Net: Locally guided global descriptor learning for 3D place recognition”. In: *IEEE Int. Conf. Robot. Autom.* 2022.

[5] Chongjian Yuan et al. “STD: Stable Triangle Descriptor for 3D place recognition”. In: *IEEE Int. Conf. Robot. Autom.* 2023.