REAL-TIME LIDAR-INERTIAL POSITIONING AND MAPPING FOR FORESTRY AUTOMATION

T. Faitli[1], T. Hakala[1], H. Kaartinen[1], J. Hyyppä[1], A. Kukko[1]

[1] Dept. of Remote Sensing and Photogrammetry, Finnish Geospatial Research Institute,
FI-02150 Espoo, Finland - (tamas.faitli, teemu.hakala, harri.kaartinen, juha.hyyppa, antero.kukko)@nls.fi

KEY WORDS: SLAM, Lidar, Laser scanning, Real-time positioning, Mapping, Forestry automation

ABSTRACT:

Accurate real-time positioning in forests is challenging due to GNSS signal degradation and an unstructured spatial environment that is difficult to conceptualize through visual sensing. Positioning is vital in any forestry automation application, such as collecting inventory, harvesting, or search and rescue missions. Lidar and inertial based solutions are popular, however they often obtain real-time computation by effectively compressing the number of measurement points to track utilizing regular geometric shapes that do not adapt well to forest. Other solutions sacrifice the high-frequency of positioning estimates or they rely on post-processing. We propose a real-time lidar-inertial SLAM-based approach that utilizes NDT scan registration, factor graphs and loop closure corrections to produce accurate and high-frequency pose estimates. To test our method, data was captured with a lidar and imu sensor mounted on a stick surveying forest sites. Ground truth trajectory for accuracy evaluation was computed by fine registering individual laser scans onto a high-quality reference point cloud recorded from the same forest area using terrestial and airborn laser scanning methods. Experiments shows, that our method can produce real-time position estimates up to 200 Hz within 2-15 cm error.

1. INTRODUCTION

Robotic applications are already present in many industries, but forestry automation is still in an early phase due to its high complexity (Visser and Obi, 2021). As in most fields, first human assisting systems appear till more autonomous machines can be developed and adapted. For example, recent proposals in forest inventory collection involve real-time feedback for the operator during surveying (Proudman et al., 2022), while other techniques try to remove the human effort as much as possible (Jaakkola et al., 2017).

Accurate positioning for automation is critical, however, it is still challenging in forest conditions (Aguiar et al., 2020). Global Navigation Satellite System (GNSS) sensors were reported to have an accuracy of approximately 0.7 meters under the canopy (Kaartinen et al., 2015), which is insufficient for most field tasks. Positioning can be improved by introducing additional sensor modalities onboard, such as lidar and Inertial Measurement Unit (IMU), and algorithms incorporating and fusing their streamed data are considered the state-of-the-art approach to compensate for unreliable, noisy, and inaccurate measurements. So far, in forestry, these solutions often prioritized accuracy over real-time and online computation (Pierzchała et al., 2018). These include studies where online computations are infeasible due to their non-causality, e.g. by (Kukko et al., 2017) the proposed fusion algorithm improves the post-processed GNSS-IMU trajectory utilizing lidar measurements expecting the availability of the whole dataset. On the other hand, real-time and online solutions have been well-studied and evaluated quantitatively in indoor and urban environments (Shan and Englot, 2018, Xu et al., 2021, Cowley et al., 2021), often including forest only as a case study and leaving its quantitative evaluation an open question targeted by this work.

The main contribution of this paper is a new concept to evaluate lidar-based positioning in forests, where it is difficult to obtain a high-quality reference otherwise. Furthermore, a positioning and mapping solution is proposed based on lidar and IMU sensor fusion capable of real-time operation. The quantitative evaluation showed that on shorter (100-280 meters long) trajectories, it could maintain the positioning error within 2-15 cm.

1.1 Problem Formulation

A moving robotic platform equipped with a lidar and IMU sensor is navigating in forest. The state of this rigid body at time instant k can be described with the following vector:

$$\mathbf{x}_k = \left[\mathcal{X}_k, {}^{\mathcal{E}}\mathbf{v}_k, {}^{\mathcal{X}}\mathbf{b}_k\right] \ , \tag{1}$$

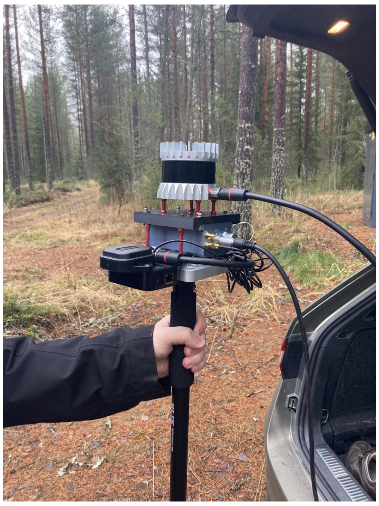

Figure 1. Measurement setup containing an Ouster OS0-128 lidar and Advanced Navigation Certus Evo GNSS/IMU.

https://doi.org/10.5194/isprs-archives-XLVIII-1-W1-2023-145-2023 | © Author(s) 2023. CC BY 4.0 License.

where $\mathcal{X}_k \in \texttt{SE3}$ is the pose and $\mathbf{v}_k \in \mathbb{R}^3$ is the velocity of the body in world frame, while ${}^{\mathcal{X}}\mathbf{b}_k \in \mathbb{R}^6$ is the IMU bias described in body frame. World frame in this context is the reference system of the map that is built from a previously unseen environment online by the robot. This is generally known as the Simultaneous Localization and Mapping (SLAM) problem.

Furthermore, the system receives lidar measurements as a set of 3-dimensional points in sensor frame with their corresponding sampling times (referred to as scan):

$$ {}^{\mathcal{L}}\mathcal{S}_k = \{(\mathbf{p}_i, t_i) | \mathbf{p}_i \in \mathbb{R}^3, t_i \in \mathbb{R}^+\}_{i=1,\ldots,N} \ , \tag{2}$$

where N is the number of points recorded during one complete rotation of the scanner. In addition, auxiliary measurements are streamed from the IMU sensor containing linear acceleration and angular velocity in its own sensor frame and their corresponding sampling times.

All measurements are subjected to noise, and they only indirectly carry information about the elements of the state vector, as there are neither absolute measurements about the pose or velocity of the robot. The task is to utilize lidar and imu modalities to reduce the noise as much as possible and infer the robot's state under real-time computation constraints. Real-time in this scenario means the system can update the state vector before new measurements arrive.

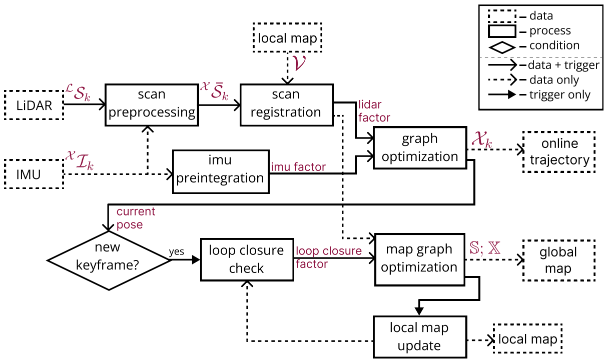

Figure 2. Schematic diagram of proposed solution.

2. METHODS

2.1 Real-time Forest SLAM

The schematic design of the proposed positioning and mapping algorithm can be seen in Figure 2. Its main structure was modeled based on the Lidar-inertial Odomentry via Smoothing and Mapping (LIO-SAM) algorithm (Shan et al., 2020), with a variety of changes to make it more suitable for forest data.

The inference problem was modelled by two factor graphs (Dellaert and Kaess, 2017) constructed in parallel. The first one was updated at each new scan and it is responsible for the estimation of the state vector in Eq. (1). It takes in a lidar factor generated as described in Section 2.2 and a preintegrated IMU factor (Forster et al., 2017). Meanwhile the other graph is responsible for the mapping, where the map is represented as a collection of graph nodes each holding a scan and its corresponding pose in world frame. The mapping graph was fed by relative pose measurements derived from consecutive lidar factors and sparsely occuring loop closure factors, and it was updated on a selective manner based on a travelled distance threshold parameter (set to 0.2 meters during the experiments).

2.2 Lidar Factor Generation

Each incoming scan was first preprocessed including deskewing and downscaling. The deskewing or rectifying process was necessary to remove unwanted motion distortion from the scan that accumulates while the scanner performs a complete rotation (see Figure 3). With our system a revolution took around 0.1 seconds, in which time interval approximately 20 IMU measurements were sampled that were integrated to obtain intermediate pose estimates. Additional pose estimates for each p_i point were then computed by interpolating on manifolds such as:

$$\mathcal{X}_{t_i} = \mathcal{X}_{m-1} \oplus \left[\frac{t_i - t_{m-1}}{t_m - t_{m-1}} (\mathcal{X}_m \ominus \mathcal{X}_{m-1}) \right] \tag{3}$$

where $\mathcal{X}_{m-1}, \mathcal{X}_m \in \texttt{SE3}$ are consecutive IMU poses with their corresponding sampling times t_{m-1} and t_m, that surrounds t_i as $t_{m-1} < t_i < t_m$. At last, a relative pose from the beginning of the revolution was computed using $\mathcal{X}_{t_i}$, which was then applied on p_i for proper placement within the lidar sensor frame.

The reconstructed scan was then downscaled using a voxel filter with uniform 0.4 meter leaf size. Finally, it was registered to a local map using point to distribution based Normal Distributions Transform (NDT) algorithm (Magnusson, 2009). The lidar factor itself was modelled as a Gaussian distribution, where the mean was the resulting transformation from the registration, while its covariance was a preset diagonal matrix scaled with the fitness score obtained during the registration.

Maintenance of the local map was necessary in order to reduce the computation time of the registration. It was constructed after each update of the mapping graph, by selecting elements from the most recently inserted graph nodes, and assemble their scans by transforming them with their corresponding pose estimates. The selection was done in a way that it ensures spatial sparsity of the selected nodes by limiting the minimum distance between two selected nodes.

2.3 Loop Closure Factor Generation

Loop closures can further constrain the mapping graph to reduce drift accumulating over time when the robot revisits a pre-

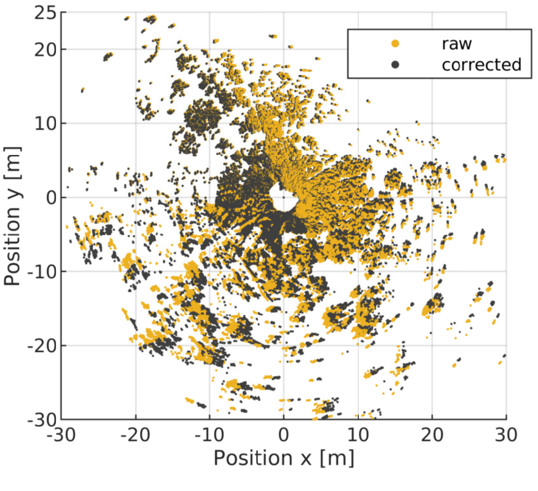

Figure 3. Raw and corrected scans, illustrating the accumulated motion distortion as the scanner completes one revolution.

viously explored area. The implemented strategy was evaluated after insertion of a new node in the mapping graph. First, while excluding the most recent nodes, the nearest graph node to the newly inserted one was identified. Then, in case it was within a parameter set (5 meters during the experiments) distance threshold, its corresponding scan was registered to the currently buffered local map using the same NDT registration algorithm. Finally, the loop closure factor was also modelled as a Gaussian distribution, where the mean was computed as the relative pose between the NDT computed transformation and the newly inserted node. Its covariance matrix was set the same way as for the lidar factor.

Additionally, the evaluation of this strategy was limited to 0.67 Hz not to overload the CPU, and factors where the registration resulted in poor fitness scores were disgarded.

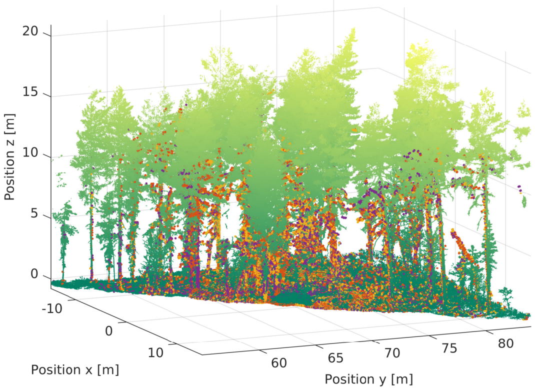

Figure 4. Illustration of individual motion corrected scans matched to high-quality TLS point cloud to generate reference trajectory. Unique colors purple, orange and yellow corresponds to individual scans, while the height-scaled green color corresponds to the reference point cloud.

2.4 Generation of Reference Trajectory

Reference trajectory was computed by registering individual motion corrected scans onto high-quality point cloud models recorded using Terrestrial Laser Scanning (TLS) and Uncrewed Laser Scanning (ULS) techniques. This matching was done offline using the Iterative Closest Point (ICP) algorithm with high number of iterations and small threshold values. An illustration of this process can be seen in Figure 4.

The SLAM estimated trajectory was used for initial guess for the ICP algorithm. However, since the output of the SLAM is in its own local frame, first the combined output point cloud was manually pre-fitted onto the reference one, obtaining a transformation that converted the SLAM trajectory to the same world frame as the reference cloud. The fitness score for each individual matching was also recorded in order to reason about the quality of the generated reference trajectory.

3. EXPERIMENTAL EVALUATION

Data was collected in the Fall of 2021 in managed boreal forest sites in Finland with an Ouster OS0-128 lidar and an Advanced Navigation Certus Evo GNSS/IMU mounted on a stick (see Figure 1.) surveying 3 plots with a walking speed between 2-6 km/h. Further descriptive information of the datasets are summarized in Table 1. The point clouds for the reference matching were recorded with a Leica RTC360 TLS system for the Evo1009 and Evo1045 datasets, while MiniVUX-1UAV drone data for the Evo1002.

Table 1. Description of collected forest datasets.

Dataset	Area	Traj. length	Duration	Forest type
	m^2	m	sec	
Evo1002	6400	279	284	scots pine
Evo1009	1600	96	103	scots pine
Evo1045	1600	154	165	norway spruce

Table 2. Summary of main positioning results.

Dataset	Mean error	RMSE-x	RMSE-y	RMSE-z
	cm	cm	cm	cm
Evo1002	15.526	7.837	16.512	10.502
Evo1009	2.360	1.738	1.802	1.314
Evo1045	4.367	2.477	3.544	1.908

The methods were implemented in C++ utilizing multicore CPU computation and the experimental data was processed simulating a real-time scenario with the Robotic Operating System (ROS) framework. The computations were run on a Dell Workstation with an Intel Core i7-1075H processor running Ubuntu 20.04.

Quantitative evaluation of the results are based on positioning error metrics, that were computed after fitting the local SLAM trajectory onto the reference one. Positioning error on its own in the following refers to the Euclidean distance between corresponding trajectory elements. Additionally Root Mean Square Error (RMSE) is presented along individual axes. Finally, the time that was taken to perform one update of the whole SLAM pipeline as in Figure 2 was considered.

3.1 Results

General overview of the positioning results are shown in Table 2. The average error during the experiments were observed between 2-15 cm. Furthermore, the longer the trajectory was, the higher the average error, which is a common behaviour in SLAM systems that has no mean for global correction.

The positioning errors throughout the measurements can be seen in Figures 5-7, where the green shaded area represents the fitness score of the reference matching. The larger the surrounding area, the worse the quality of the reference. It is mainly visible in the case of the Evo1002 dataset, where the SLAM trajectory was not fully covered by the reference point cloud in the middle of the measurement. The reference trajectory otherwise had great fitness, and the positioning errors are generally well-bounded around a constant value with a few outliers.

The estimated SLAM and reference trajectories for the Evo1002 dataset are shown in Figure 8. Additionally, the GNSS trajectory that was recorded using a single antenna with the Certus Evo GNSS/IMU unit is also included in the plot. The estimated trajectory tracks well the reference one, while the GNSS one is significantly worse, further showcasing the degrading effects on the signal inside the forest. Trajectories of the other datasets are very similar, therefore, excluded. They all return to the neighbourhood of the initial position closing one big loop in the trajectory as done in the original surveying.

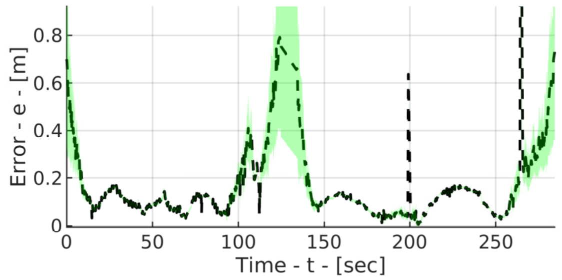

Figure 5. Positioning error for dataset Evo1002.

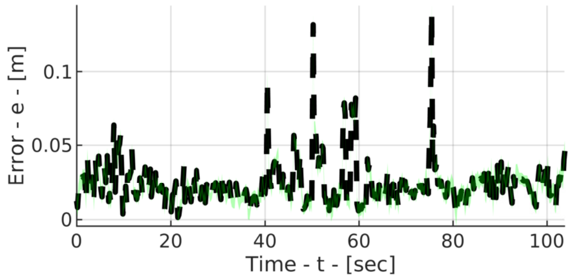

Figure 6. Positioning error for dataset Evo1009.

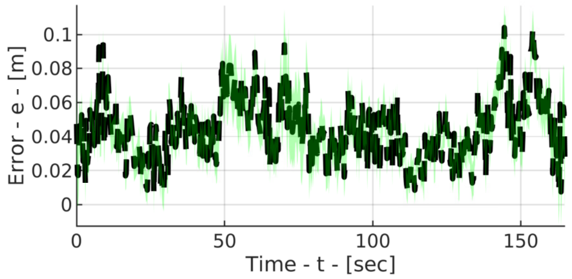

Figure 7. Positioning error for dataset Evo1045.

The distributions of computational times for each dataset are shown in Figure 9. As scans are arriving at 10 Hz with the system under consideration, the target limit to perform one SLAM update is 100 ms. It can be seen that the majority of updates were executed within this limit, therefore the method is capable of real-time execution.

3.2 Discussion

Real-time computation is primarily feasible due to the local mapping strategy and the compressing capabilities of the NDT registration algorithm. In theory, the robot relies on less information to position itself and accuracy is expected to decrease. Figure 10 showcases the playoff between the density of the local map controlled by the limiting distance threshold between selected nodes at reconstruction. As the distance threshold is increasing, less nodes are selected and therefore less scans have to be processed, drastically improving the computation speed till it converges to approximately 40 ms. On the other hand, the positioning accuracy is less sensitive for this, enabling adequate accuracy in real-time.

The compression ratio within the NDT registration can be controlled by the grid size used within the algorithm, and its effects on the performance is shown in Figure 11. The larger the grid size, the more compressed the local map becomes, carrying less information about the surroundings. It would also suggest higher computation speed, however the algorithm might require more iterations to converge therefore the computation speed is less effected by this. The system remains robust until around 3 meter grid size, from which point the positioning error exponentially increases.

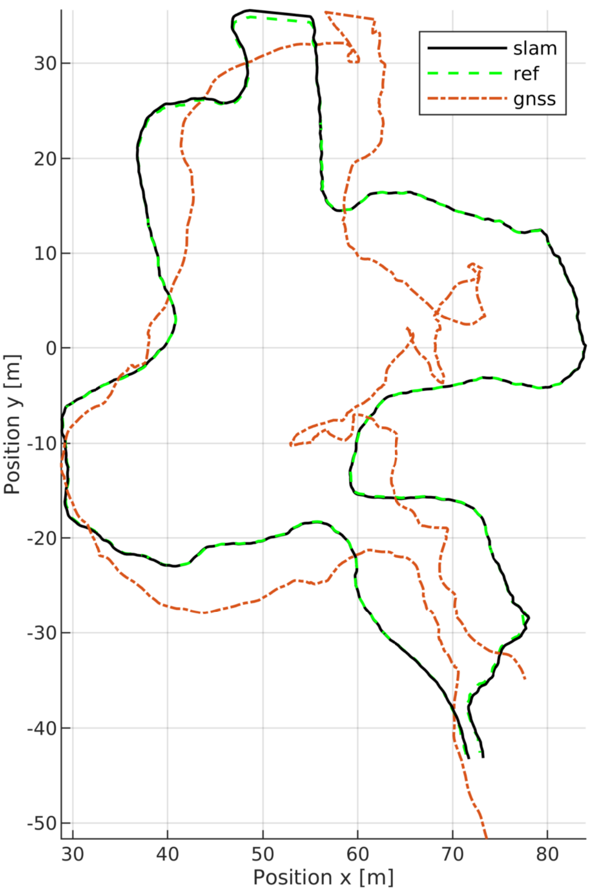

Figure 8. Trajectories for dataset Evo1002.

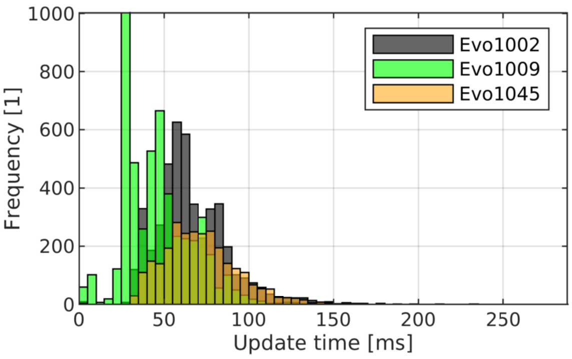

Figure 9. Distribution of the computational times required for a SLAM update.

A segment of the SLAM built forest point cloud model is shown in Figure 12, while a side-by-side comparison with the reference point cloud is illustrated in Figure 13. The resulting point cloud well resembles the reference, and the trees appear correctly placed and can be distinguished without any shadowing effects. However, a widening effect can be observed on the individual tree level, as the trunks in the SLAM-built cloud ap-

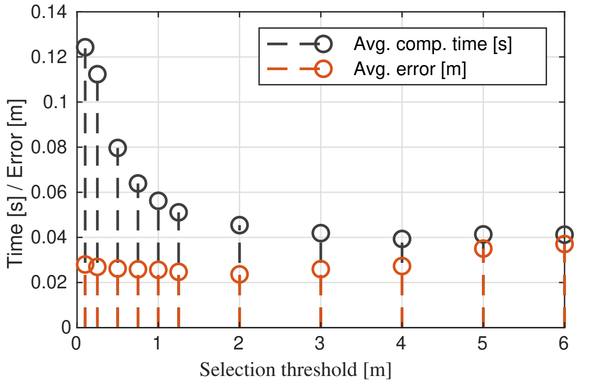

Figure 10. Performance with different distance thresholds used during node selection for the local map construction. Computed for dataset Evo1009.

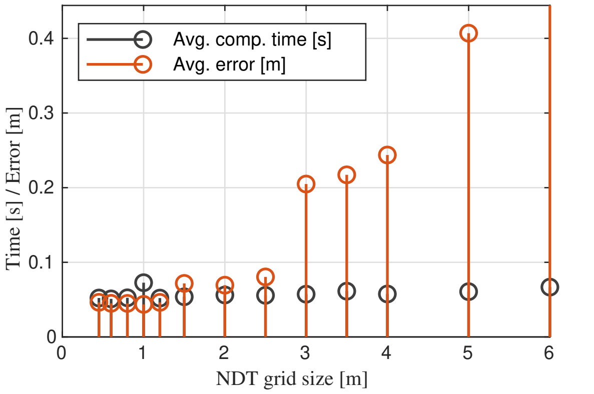

Figure 11. Performance with different grid cell sizes used within the NDT registration algorithm. Computed for dataset Evo1045.

pear much more extensive. This is most likely derived from the Ouster lidar's high beam divergence compared to other remote-sensing oriented lidar sensors. On the one hand, it can be useful for automation tasks as thinner objects are more likely to be detected for obstacle avoidance and navigation purposes while less applicable for field surveying tasks.

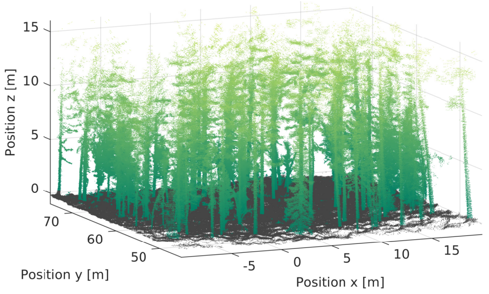

Figure 12. Segment of the resulting point cloud model from dataset Evo1009.

4. CONCLUSION

A real-time positioning and mapping algorithm is proposed using lidar and inertial sensors. Additionally, a new approach to generate a reference trajectory for lidar-based positioning evaluation in the forest is conceptualized, where other popular means are less applicable. The SLAM method can provide high-frequency and accurate estimates in a previously unseen and challenging environment while maintaining a map of the surroundings. This online positioning and mapping algorithm can provide a basis for further forestry automation applications as input for planning and control algorithms.

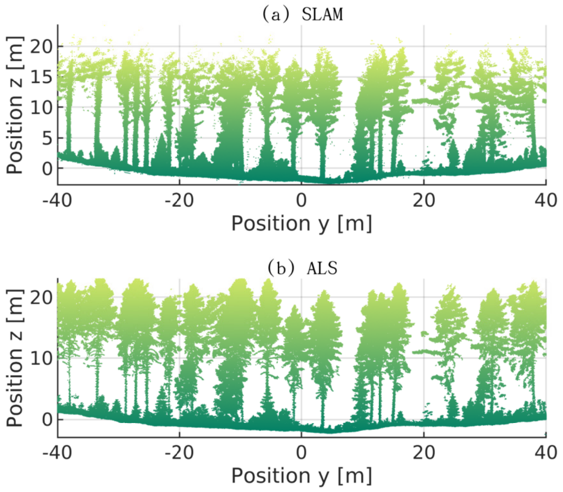

Figure 13. Segment from SLAM generated and ALS reference cloud of the same are from Evo1002 dataset.

In future work, the method should be tested on more extended datasets, in different forest sites, and potentially with additional sensor units. Further developments in reference data generation for forest positioning evaluation are of interest. Especially alternative proposals on providing the initial guess for the reference matching, as in this scenario, it is only applicable if the SLAM optimized trajectory is relatively consistent globally, that might not be the case for longer measurements.

AUTHOR CONTRIBUTIONS

T. Faitli: design and implementation of proposed method, measurement setup and data collection, data processing and evaluation, writing; **T. Hakala**: measurement setup and data collection; **H. Kaartinen**: reference point cloud data; **J. Hyyppä**: supervision; **A. Kukko**: supervision and review of the manuscipt;

ACKNOWLEDGEMENTS

The work was supported by the funds and infrastructure granted by Academy of Finland through UNITE Flagship (337656) and projects ‘Feasibility of Inside-Canopy UAV Laser Scanning for Automated Tree Quality Surveying’ (334002), ’Estimating Forest Resources and Quality-related Attributes Using Automated Methods and Technologies’ (334829), ’Understanding Wood Density Variation Within and Between Trees Using Multispectral Point Cloud Technologies and X-ray Microdensitometry’ (331708), and the Ministry of Agriculture and Forestry project ‘Future forest information system at individual tree level’ (VN/3482/2021). Academy of Finland Research infrastructure (346382) was applied for the study.

Furthermore, the authors would like to thank the help of Heikki Hyyti and Aimad El Issaoui throughout the collection of the reference point cloud data.

REFERENCES

Aguiar, A. S., Santos, F. N. D., Cunha, J. B., Sobreira, H., Sousa, A. J., 2020. Localization and mapping for robots in agriculture and forestry: A survey. *Robotics*, 9(4), 1–23.

Cowley, A., Miller, I. D., Taylor, C. J., 2021. UPSLAM: Union of Panoramas SLAM. http://arxiv.org/abs/2101.00585.

Dellaert, F., Kaess, M., 2017. Factor Graphs for Robot Perception. *Foundations and Trends in Robotics*, 6(1-2), 1–139. http://www.nowpublishers.com/article/Details/ROB-043.

Forster, C., Carlone, L., Dellaert, F., Scaramuzza, D., 2017. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. *IEEE Transactions on Robotics*, 33(1), 1–21.

Jaakkola, A., Hyyppä, J., Yu, X., Kukko, A., Kaartinen, H., Liang, X., Hyyppä, H., Wang, Y., 2017. Autonomous collection of forest field reference—The outlook and a first step with UAV laser scanning. *Remote Sensing*, 9(8), 1–12.

Kaartinen, H., Hyyppä, J., Vastaranta, M., Kukko, A., Jaakkola, A., Yu, X., Pyörälä, J., Liang, X., Liu, J., Wang, Y., Kaijaluoto, R., Melkas, T., Holopainen, M., Hyyppä, H., 2015. Accuracy of Kinematic Positioning Using Global Satellite Navigation Systems under Forest Canopies. *Forests 2015, Vol. 6, Pages 3218-3236*, 6(9), 3218–3236. https://www.mdpi.com/1999-4907/6/9/3218/htm https://www.mdpi.com/1999-4907/6/9/3218.

Kukko, A., Kaijaluoto, R., Kaartinen, H., Lehtola, V. V., Jaakkola, A., Hyyppä, J., 2017. Graph SLAM correction for single scanner MLS forest data under boreal forest canopy. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132, 199–209.

Magnusson, M., 2009. *The Three-Dimensional Normal-Distributions Transform.* 10, Örebro University.

Pierzchała, M., Giguère, P., Astrup, R., 2018. Mapping forests using an unmanned ground vehicle with 3D LiDAR and graph-SLAM. *Computers and Electronics in Agriculture*, 145(December 2017), 217–225.

Proudman, A., Ramezani, M., Digumarti, S. T., Chebrolu, N., Fallon, M., 2022. Towards real-time forest inventory using handheld LiDAR. *Robotics and Autonomous Systems*, 157, 104240.

Shan, T., Englot, B., 2018. LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain. *IEEE International Conference on Intelligent Robots and Systems*, 4758–4765.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., Rus, D., 2020. LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. *arXiv*. http://arxiv.org/abs/2007.00258.

Visser, R., Obi, O. F., 2021. Automation and robotics in forest harvesting operations: Identifying near-term opportunities. *Croatian Journal of Forest Engineering*, 42(1), 13–24.

Xu, W., Cai, Y., He, D., Lin, J., Zhang, F., 2021. FAST-LIO2: Fast Direct LiDAR-inertial Odometry. http://arxiv.org/abs/2107.06829.