

Effective LiDAR-based Place Recognition



Haedam Oh
St Edmund Hall
University of Oxford

A thesis submitted for the degree of
4-th Year Project

2024

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Challenges in natural environments	2
1.3	Contribution	3
2	Related Work	4
3	Methods	7
3.1	Method	7
4	Experiments	15
4.1	Experimental Evaluation	15
4.1.1	Place Recognition Descriptors	16
4.1.2	Online Place Recognition	17
4.1.3	Offline Multi-Mission SLAM	20
4.1.4	Relocalization	21
4.1.5	Study of ICP inlier-based check	22
5	Conclusion	25

1

Introduction

1.1 Motivation

This report presents a comprehensive exploration of various LiDAR-based place recognition algorithms focusing on their application in long-term navigation across unstructured natural environments. Place recognition is the capability to identify specific revisited locations and correctly determining robot's pose by comparing the current observation with previously visited places.

The primary goal is to design and integrate a LiDAR place recognition system within a Simultaneous Localization and Mapping (SLAM) framework. This integration is key for achieving robust loop closure detection, precise pose optimization, and the creation of accurate 3D maps, which are essential for long-term navigation and mapping systems.

In context of long-term navigation, LiDAR proves to be more robust than RGB cameras, capable of capturing much longer-range 3D scene data, less susceptible to restricted viewpoints and visual change due to lighting conditions. Therefore, LiDAR is used as a main sensor to obtain 3D pointclouds of the environment, then processing 3D pointclouds to extract meaningful features to check previously visited places.

Descriptors are used to represent the environment and compare with each other to determine whether the current observation is a revisited place. There are two

main approaches: handcrafted features and learning-based features. Handcrafted features are manually designed to capture the geometric characteristics of the environment. This approach is more generalizable and robust to changes in the environment, but it is difficult to design features that are both discriminative and invariant to environmental changes. Recently, learning-based features using recent deep-learning became popular in place recognition.

1.2 Challenges in natural environments

Despite the advantages of LiDAR in place recognition, its application in unstructured natural environments like forests presents several notable challenges. Firstly, the task of extracting meaningful features from 3D point clouds in these settings is complex. Natural environments are irregular structures like trees, which not only lack fine structure but also undergo seasonal changes, affecting their shape and density over time. This complicates the creation of consistent geometric representations crucial for accurate place recognition. Secondly, the limitation in the LiDAR's field of view poses a significant challenge. In large-scale natural environment, LiDAR sensors often struggle to capture the complete vertical extent of the environment. This issue is particularly pronounced in areas with tall trees or undulating terrain, where the full scope of the environment is crucial for comprehensive mapping and navigation. Thirdly, processing the voluminous 3D data typical of LiDAR systems, such as point clouds, demands substantial computational resources in terms of memory and processing time. This requirement poses a considerable challenge for robotic applications, where often limited computational resources are available, and real-time performance is essential. Lastly, the dynamic nature of unstructured terrains contrasts sharply with the more predictable environments of urban or structured settings, like roads used in autonomous driving. In natural landscapes, the paths and terrains are subject to continual change. The extreme orientation and movement often encountered in unstructured environments can lead to sparse and unreliable sensor data, exacerbating the issue of frequent odometry drift.

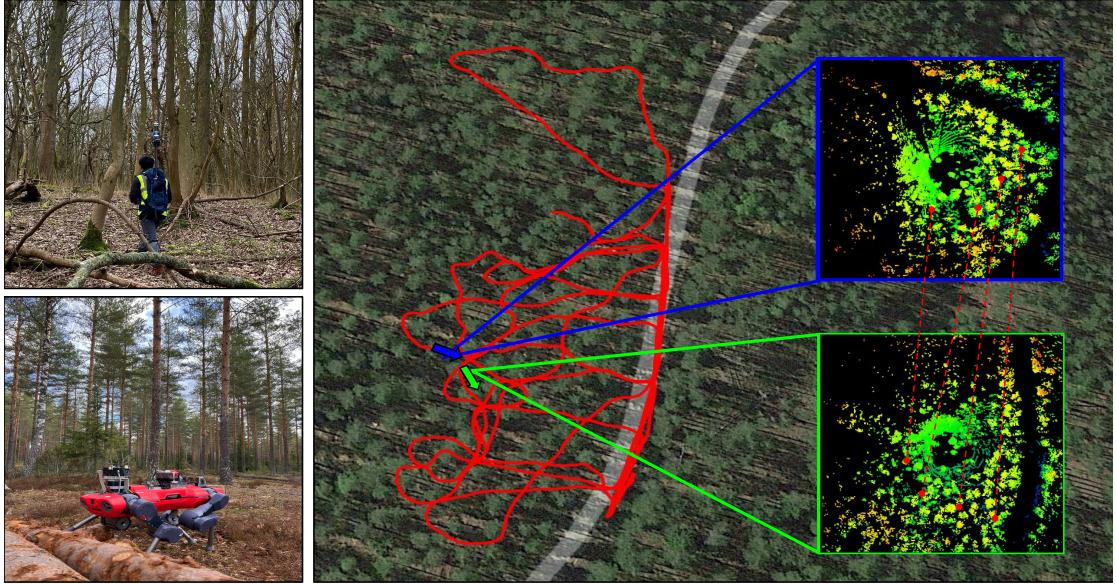


Figure 1.1: Place recognition in dense forest environments using a backpack-mounted LiDAR and legged robot equipped with LiDAR. Right: Illustration of loop closure detection within a deeply forested area, with a baseline distance of xx meters and an orientation offset of yy degrees. Dotted red lines highlight corresponding locations from the two point clouds.

1.3 Contribution

Overall, there is a compelling need for LiDAR place recognition in natural environments. In this report, I showed

- 1. Evaluation of both handcrafted and learning-based LiDAR place recognition models' descriptors in dense forest environments.
- 2. Complete pipeline integrating loop closures into a SLAM system.
- 3. Analysis of loop-closure pairs based on distances and viewpoints differences.
- 4. Demonstration of our system for online and offline SLAM modes, as well as relocalization in previously mapped environments.

2

Related Work

Various LiDAR-based place recognition approaches exist, including handcrafted models that extract geometric features and summary statistics [5, 24], and learning-based approaches that employ Convolutional Neural Networks (CNNs) and Transformers to compute high-level descriptors capable of distinguishing between different places [19, 9]. In particular we are interested in methods which can produce an estimate of relative 6-DoF transform between the matches, as this is essential for integrating loop-closures within a SLAM system.

Among handcrafted approaches, ScanContext[5, 4] stands out as a widely adopted technique that generates a descriptor by encoding the point cloud as a bird’s-eye view representation. It captures height information within defined sectors and integrates them into a 2D descriptor. ScanContext has also been enhanced by incorporating additional information such as intensity [20] and semantics [10] to create more informative descriptors. Another type of handcrafted descriptor, STD [24], encodes boundaries of planes as vertices and connects them to create multiple triangles. STD operates without requiring a 360-degree scan, making it compatible with LiDAR systems with a 90-degree field of view (e.g. Livox Aria). Both ScanContext and STD can estimate a relative transformation between corresponding scans by matching their descriptors. However, despite their impressive

performance in urban scenarios, their effectiveness in capturing the structure of natural environments such as forests is limited.

Recent learning-based models, such as Logg3dNet [19], MinkLoc3D [8], and EgoNN [9] employ discretized representations and contrastive learning schemes to compute point-based local descriptors. This process is followed by generating a global descriptor of aggregated local features using methods like GeM [14], P2O [17], and NetVLAD [1]. For instance, Logg3dNet [19] uses a sparse convolutional network with both local consistency and global scene losses learned in a contrastive manner. Similarly, EgoNN [9] employs a deep CNN architecture to extract local descriptors and key points through regression, subsequently aggregating them using GeM to form a global descriptor. Both models facilitate relative transformation by matching local keypoints with RANSAC, which enables a finer resolution registration using ICP. While Transformer variants [25, 22, 26, 23] are known for their ability to capture long-term dependencies, they have high computational costs and often focus on global-level place recognition without explicitly estimate the relative transformation between candidates. [we need to talk about this - its still too abrupt]

As an alternative to a whole scan descriptor, some methods utilize segments to capture the important elements, effectively compressing the information in the entire point cloud map into a more compact representation. SegMatch [**dube2017icra**] by Dubé et al. computes local segments and extracts geometric features as a descriptor. In their follow-up work, SegMap [**dube2018rss**], these features were learned via a CNN, showing improved overall performance. Tinchev et al. [**tinchev2019ral**, 15] have applied the segment-based approach to natural environments, showing promising results. However, these methods are vulnerable when these segments cannot be reliably detected due to occlusions in dense forest environments, as well as long-term changes in the environment.

Several LiDAR point cloud datasets are available for benchmarking place recognition models in urban scenarios [**behley2019iccv**, 11, 6]. However, there are few datasets for natural environments [16, 7]. In particular, the Wild-Places dataset [7] is tailored to large-scale place recognition in forests. This dataset provides point

clouds and ground truth poses collected in a forest national park in Australia using hand-held spinning LiDAR at various times of the year.

In this paper, we assess the performance of four different place recognition models including both handcrafted (ScanContext and STD) and learning-based (Logg3dNet and EgoNN) models on our dense forest datasets ¹ collected across three different countries using a backpack LiDAR mapping device, in contrast to previous methods that often concentrated on access roads.

¹Our datasets will be published at <https://ori.ox.ac.uk/labs/drs/datasets-drs/> using the format as Wild-Places

3

Methods

3.1 Method

Our objective is to evaluate the capacity of existing LiDAR place recognition models to successfully produce robust loop closure candidates within dense forest environments. Our evaluation considers three distinct tasks:

- *Task A: Online SLAM*: the proposed loop candidates contributing to a globally-consistent pose graph mapping system in an incremental manner.
- *Task B: Offline multi-mission SLAM*: loop candidates used to link different physically overlapping missions collected at different times.
- *Task C: Relocalization*: place recognition in a prior map made up of individual scans enabling autonomy within the map such as longer term monitoring or harvesting.

Our system infrastructure is shown in Fig. 3.1. For state estimation, we use a LiDAR-inertial odometry system—VILENS [21]—, in conjunction with a pose graph SLAM framework [13]. Additionally, we implemented a *place recognition & verification server*, which not only provides a common interface for the different LiDAR-based place recognition models but also multi-stage verification procedures for its use in the different proposed tasks.

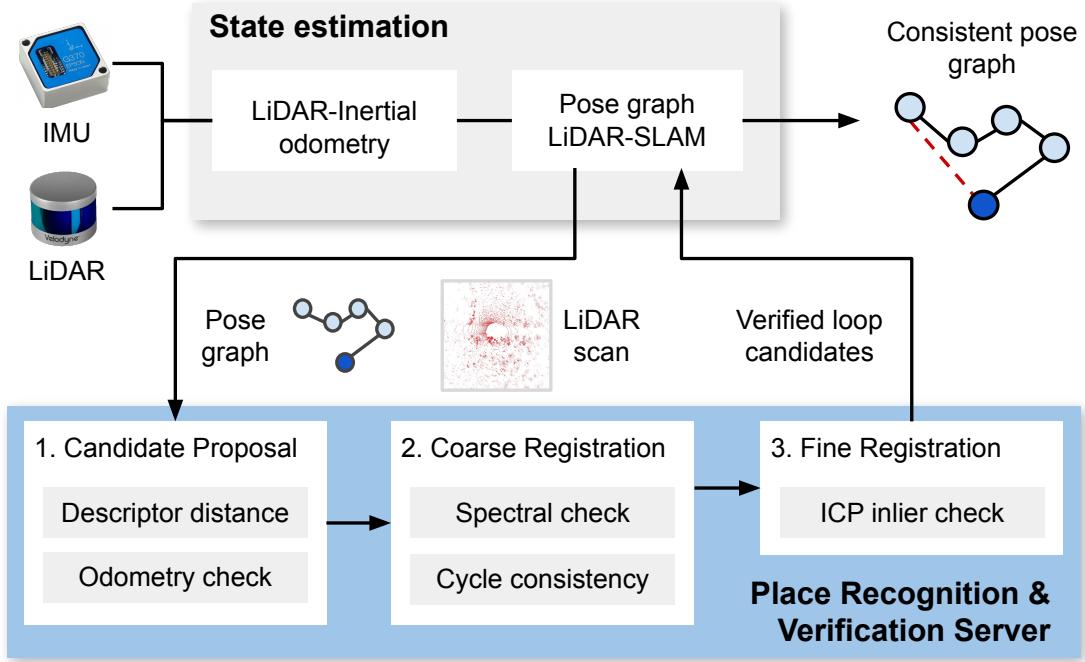


Figure 3.1: Our place recognition pipeline. VILENS provides a continuous odometry estimate at 10 Hz. Pose graph SLAM is used to optimize poses after successful loop closure verification. The place recognition module consists of three steps: Loop candidate proposal, coarse registration, and fine registration. We verify loop candidates at the global descriptor-level, local feature-level consistency, and finally fine registration level. A loop candidate is integrated in the pose graph only if it passes these three stages.

In the following sections, we present the technical details of the place recognition server, and then we present its integration to solve the three aforementioned tasks.

Place Recognition & Verification Server

[Haedam: reference to method figure which step is which] Our place recognition pipeline consists of three steps: loop candidate proposal, coarse registration, and a final fine-registration. At each step, we perform appropriate checks to filter out incorrect loop closures.

The main inputs are the pose graph, with corresponding LiDAR scans attached to each pose, as well as the single query scan. The query scan is provided by different sources depending on the task we are solving. For example, in the relocalization task it will be a live scan directly from the LiDAR sensor. Further details are provided in the corresponding sections.

Step 1: Loop candidate proposals

Initial loop closure candidates are obtained by comparing global descriptors extracted from the pose graph scans as well as the query scan. In this paper, we evaluate four state-of-the-art methods for descriptor extraction: the learning-based Logg3dNet [19] and EgoNN [9], as well as the handcrafted ScanContext [5] and STD [24].

Given the reference pose graph and the query scan, we compute a database of descriptors using all the scans in the pose graph, given by the matrix $\mathbf{D} \in \mathbb{R}^{N \times M}$, where N is the number of poses in the pose graph and M the descriptor dimension. Additionally, we compute the descriptor for the query scan, denoted by $\mathbf{d}_q \in \mathbb{R}^{M \times 1}$.

To obtain candidates, we simply compute the pairwise distances of the scan to the database using the cosine similarity:

$$\mathbf{S} = \mathbf{D} \cdot \mathbf{d}_q \in \mathbb{R}^{N \times 1} \quad (3.1)$$

The vector of descriptor distances \mathbf{S} is sorted by increasing distance, and only the top- k candidates are selecting using a distance threshold τ_s .

[we should discuss this comment here - I dont know what it means:] If a spatial prior is available, for example from LiDAR-inertial odometry, we also perform an additional spatial check discarding all the candidates that are more than 20 m away from the query scan. The output is a set of candidate nodes $\{n_c\}$.

Step 2: Coarse Registration

Next, we estimate the relative 6DoF transformation that expresses the pose associated to the query scan w.r.t each candidate node, which we denote $\Delta \mathbf{T}$. For the handcrafted methods (ScanContext and STD), the relative transformation is directly an output of descriptor computation. For the learning-based approaches, we use the point-wise feature vectors outputted in the forward pass of Logg3dNet and EgoNN for point matching, which is used in a RANSAC-based pose estimation scheme [3] to obtain the estimated relative transformation.

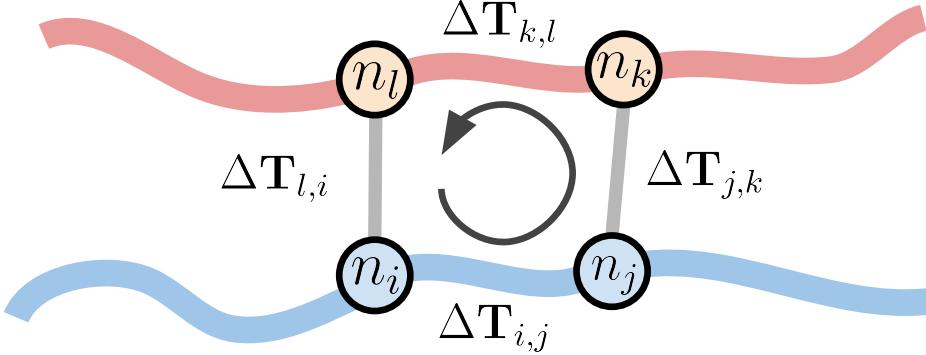


Figure 3.2: Our proposed cycle consistency check is general and applies to the online and offline multi-mission SLAM case, as well as relocalization tasks. We only need the relative transformation estimates and loop candidates between four nodes n_i, n_j, n_k, n_l to verify the validity of a loop. Please refer to Sec. 3.1 for technical details.

We additionally verify the inlier matches using the *Spectral Geometric Verification* [18] method, which provides an additional measure of the quality of the feature matches.

Lastly, we carry out a *cycle consistency* verification, which checks whether the relative transformations between pairs of nodes are mutually consistent with one another. In brief, when given four pose graph nodes n_i, n_j, n_k, n_l , we test how close the following equivalence holds:

$$\Delta\mathbf{T}_{i,j} \Delta\mathbf{T}_{j,k} \Delta\mathbf{T}_{k,l} \Delta\mathbf{T}_{l,i} \approx \mathbf{I}_{4 \times 4} \quad (3.2)$$

If the difference around a cycle is more than a threshold of 10 cm and 1° we reject the candidate. This is shown in Fig. 3.2, please refer to the corresponding sections for further details.

The interpretation of these transformations change if we discussing online SLAM (Sec. 3.1), offline multi-mission SLAM (Sec. 3.1), or pure relocalization (Sec. 3.1).

Step 3: Fine Registration

Finally, we employ the Iterative Closest Point (ICP) algorithm [2] for fine registration of the proposed candidates. We use the *libpointmatcher* implementation [12], which also provides information on the quality of the registration, such as the proportion inlier points and the Residual Error of each point to access the alignment.

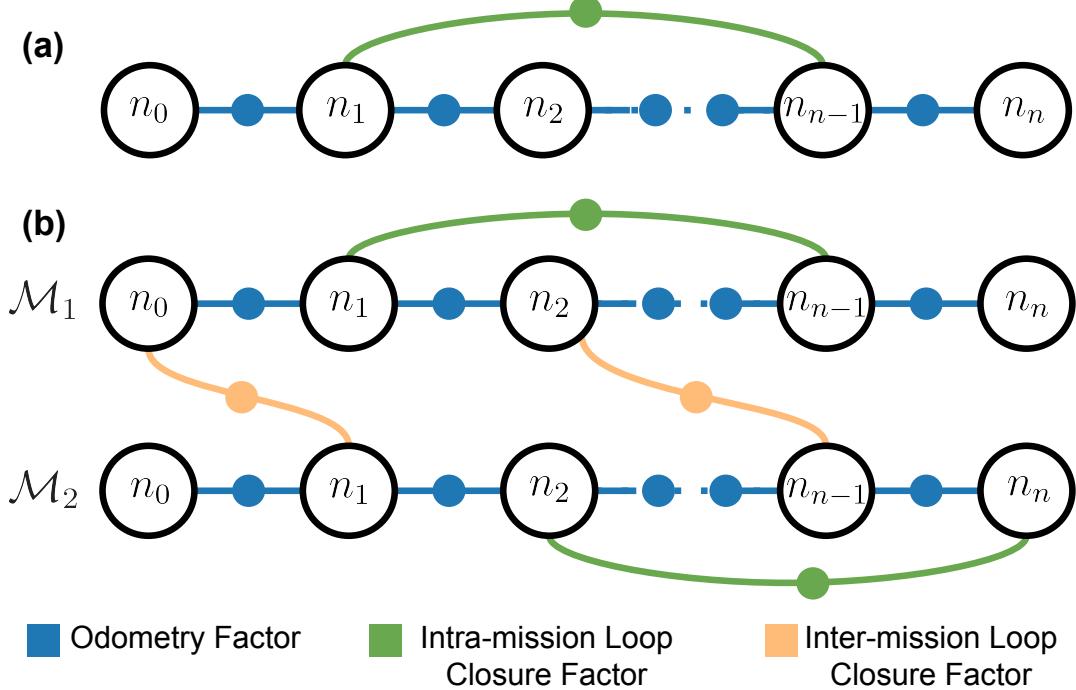


Figure 3.3: Pose graph formulation used for (a) online, and (b) offline multi-mission SLAM optimization. Each node n_i has a 6DOF pose \mathbf{x}_i , which correspond to the main variables estimated on each case.

We use the proportion of inliers and the residual error of 20 cm as a final verification step to reject loop candidates. The verified candidates are then used for SLAM or relocalization tasks, which are detailed in the following sections.

Task A: Online Single-mission SLAM

The first task we consider is LiDAR-based online SLAM. Our implementation defines it as an incremental pose graph estimation problem (see Fig. 3.3,(a)). Consider consecutive loop closures at nodes n_i , n_{i+1} and n_j , n_{j+1} . Edges are provided by relative estimates from our LiDAR-inertial odometry system (odometry factors, denoted by $\Delta \mathbf{T}_{i,i+1}, \Delta \mathbf{T}_{j,j+1}$), and verified loop closure candidates from our place recognition server (loop closure factors, $\Delta \mathbf{T}_{i+1,j}, \Delta \mathbf{T}_{i,j+1}$). [a pose graph is a type of factor graph. no need to say factor graph here]

For the cycle consistency verification described in Sec. 3.1, we consider the relative transformation change between consecutive loop closure candidates w.r.t

the pose graph poses and the odometry change (Fig. 3.2 i, j, k, l , replaced by $i, i+1, j, j+1$). Again, a cycle consistency needs to be satisfied:

$$\Delta \mathbf{T}_{i,i+1} \Delta \mathbf{T}_{i+1,j} \Delta \mathbf{T}_{j,j+1} \Delta \mathbf{T}_{i,j+1}^{-1} \approx \mathbf{I}_{4 \times 4} \quad (3.3)$$

[Haedam: exp:cycle consistency works well in this scenario,]

Task B: Offline Multi-Mission SLAM

Offline multi-mission SLAM addresses the challenge of merging multiple pose graph SLAM missions $\mathcal{M}_{1,\dots,n}$, collected over time, with partly overlapping area. The goal is to find inter-mission loop candidates to construct a unified map in a common reference frame. This application is relevant for forestry applications, where it is required to map larger areas by integrating surveys conducted over multiple missions or campaigns.

Unlike the scenario of on-road navigation, where similar routes (hence locations) are revisited, we considered off-road scenarios where the missions are collected in dense forests, where it is often unfeasible to retrace the same paths on each sequence. To avoid inefficiently retracing our steps, we wish to identify loop candidates when passing no closer than about 10 m, providing the flexibility needed to merge two roughly overlapping missions.

Each mission \mathcal{M}_i is defined by a pose graph with odometry factors and intra-mission loop closures, obtained during each independent online SLAM run. We aim to provide additional *inter-mission* loop candidates that bridge nodes across missions, as shown in Fig. 3.3 (b). In this case, loop candidates are obtained by incrementally matching each mission of the nodes on matched missions and corresponding LiDAR scans across the missions. [do you actually do all-v-all? I incrementally build the database graph so its 1v1 and then 2v2.] [Haedam: you're right, we incrementally build it] For the loop proposal step, we executed the same procedures described in Sec. 3.1, but with the stringent descriptor distance threshold τ_s . We observed that in the multi-session case, this was required to allow a larger set of candidates that were later verified by stronger procedures such as the cycle

consistency check. For the cycle consistency step, we considered pairs of nodes within the same mission, namely $n_i, n_j \in \mathcal{M}_1$ and $n_k, n_l \in \mathcal{M}_2$. The intra-mission relative transformation were then $\Delta\mathbf{T}_{i,j}, \Delta\mathbf{T}_{k,l}$, while the inter-mission relative transformations between loop candidates were given by $\Delta\mathbf{T}_{i,k}, \Delta\mathbf{T}_{j,l}$:

$$\Delta\mathbf{T}_{i,j} \Delta\mathbf{T}_{j,l} \Delta\mathbf{T}_{k,l}^{-1} \Delta\mathbf{T}_{i,k}^{-1} \approx \mathbf{I}_{4 \times 4} \quad (3.4)$$

Task C: Relocalization

Lastly, we considered the case in which a prior map of the forest was available (from online SLAM). Our place recognition & verification server could then be used as a relocalization module, by using the loop candidate proposals to produce initial pose estimates, then coarse-to-fine registration achieving real-time localization of the LiDAR sensor base B with the prior map's coordinate frame M, denoted by \mathbf{T}_{MB} .

[the following paragraph is confusing and poorly written. If we are doing cycle consistency checking we don't have a 'successful relocalization' we only have a 'possible candidate'] [can you try again?] [Haedam: Okay, I add a sentence above and below] Similarly to the previous tasks, the main difference is in defining the cycle consistency check. For this case, it is between the current and the last successful relocalization: Given the last relocalization estimate $\mathbf{T}_{MB}(t-1)$ and the current estimate $\mathbf{T}_{MB}(t)$, we compared the against the odometry estimates at the same timestamps $\mathbf{T}_{OB}(t-1)$ and $\mathbf{T}_{OB}(t)$, where O indicates the fixed odometry frame. The cycle consistency check is then defined as:

$$\underbrace{\mathbf{T}_{MB}(t)^{-1} \mathbf{T}_{MB}(t-1)}_{\Delta\mathbf{T} \text{ in } M \text{ frame}} \underbrace{\mathbf{T}_{OB}(t-1)^{-1} \mathbf{T}_{OB}(t)}_{\Delta\mathbf{T} \text{ in } O \text{ frame}} \approx \mathbf{I}_{4 \times 4} \quad (3.5)$$

This check is used to verify the relocalization estimate, and if successful, ICP is used to fine-localize the LiDAR sensor within the prior map. [again, we need to point out that localisation into a giant point cloud map would be very inconvenient.]

This relocalization capability facilitates various applications, such as enabling a harvester robot to operate autonomously within a prior map or enabling foresters to visualize a rendering of the virtual forest along with important information on a

screen in real-time. An example demonstrating this capability is later presented in Sec. 4.1.4, where a prior map of the forest is generated using a backpack-based LiDAR mapping system, and a legged robot continuously relocalizes itself within that prior map as part of an inspection task. [please don't over claim - we didnt do this. soften your comments here please]

[But Matias: it would be awesome to do what is described here!] [Haedam: I will try this experiment and see if it works.]

4

Experiments

4.1 Experimental Evaluation

In this section, we rigorously evaluate our place recognition pipeline to assess its performance in dense forest environments. Our evaluation includes four distinct test sites featuring varying forest compositions: Evo (Finland) characterized by coniferous trees; Stein-Am-Rhein (Switzerland) Wytham Woods (UK) and Forest of Dean (UK) containing both broad-leaf and coniferous tree species. We evaluate all three operational modes of our system: Online SLAM, Offline Multi-Mission SLAM, and Relocalization. The experiments conducted are as follows: (i) Evaluation of four different place recognition models at the descriptor-level, tested across multiple forest environments with different LiDAR setups. (ii) Performance assessment during both online and offline SLAM operations within dense forest settings. [somewhere here you need to say you tested only Logg3dNet for some experiments] (iii) Analysis of successful loop closures, based on baseline distance and orientation differences. (iv) Demonstration of the relocalization application in a previously mapped forest environment, showcasing its utility in an inspection task performed by a quadruped robot. These experiments provide insights into the capabilities and limitations of our place recognition pipeline, particularly in challenging forest terrains.

4.1.1 Place Recognition Descriptors

In this experiment, we evaluated the descriptors of four different place recognition models (Logg3dNet, EgoNN, ScanContext, STD) 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. Precision-recall curves (See Fig. 4.1) show how accurately (precision) and frequently (recall) each model detects correct loop candidates within a reasonable distance threshold (here set to 10 m) at various descriptor thresholds τ_s in four different forests. [Haedam: We used a Hesai XT32 LiDAR with 50m range and 30 degree narrow field of view to map Evo and Stein am Rhein, and a Hesai QT64 with 30m range, 100 degree wide field of view to map Wytham Woods.]

From the precision-recall curves Fig. 4.1, it is evident that Logg3dNet consistently outperforms the other models across the four different forests. Particularly, on the Evo and Stein am Rhein datasets, where a longer range with narrow field of view Hesai XT32 LiDAR was employed, Logg3dNet showed the best performance both in terms of precision and recall, without experiencing any sudden drops in precision. In contrast, ScanContext demonstrates a significant decrease in precision, attributed to a limited vertical field of view of LiDAR sensor. In more challenging scenarios such as Wytham Woods characterized by complex terrains featuring hills and valleys with dense tree clutter captured with a wide field of view QT64 LiDAR, handcrafted models show a notable decline in performance. However, Logg3dNet remains robust, successfully retrieving a substantial portion of correct loop-candidates, achieving a 70% precision at a 50% recall rate.

[I get to this bit and there is just a hole in the work because nothing about Logg3dNet’s algorithm is ever discussed. There is no tuning or ablation. It just presented ‘as is’.]

In Fig. 4.2, we present a heatmap of the descriptor distances between all query and database scans computed under a specific τ_s of F_1 -max score. It offers an illustrative visual representation of the discriminative potential of each descriptor. Consistent with the precision-recall curves, Logg3dNet descriptors exhibit high

consistency with the ground truth heatmap (highlighted in white circles), 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 choose Logg3dNet as the place recognition module in our system for the rest of the experiments.

[technically you tested with data from 4 countries - Australia!]

4.1.2 Online Place Recognition

In this experiment, we investigate the online place recognition capability of our system, wherein loop closures from the place recognition module are integrated into the SLAM system. The database D , is incrementally built as the sensor moves through the environment. When matching, we exclude the most recent 30 seconds of data to prevent loop closures with immediately recent measurements.

Fig. 4.3 presents an illustrative example of online SLAM performance using the Evo dataset, depicting the sets of loop candidates after each verification step. Initially, many loop closure candidates are proposed (shown in blue) under a relaxed descriptor matching threshold τ_s of F_1 -max score. Next, loop closures beyond a distance threshold of 20 m are rejected using the odometry information [again, we should discuss this 20m is not the right metric - it should be the marginal covariance]. Following this, a subset of loop closure candidates are identified using RANSAC matching (highlighted in orange), and finally, a refined set of loop closures that pass the consistency and ICP steps are integrated into the SLAM framework.

In Fig. 4.4, we conduct a comprehensive analysis of loop closure statistics based on distance and viewpoint angles. Our findings show that the system can successfully identify loop closure pairs across considerable baseline distances (10-20 m). We observe that despite the large distance, a significant portion of initial candidates can be registered using RANSAC-based matching. However, the proportion of candidates verified by ICP decreases as the distance between

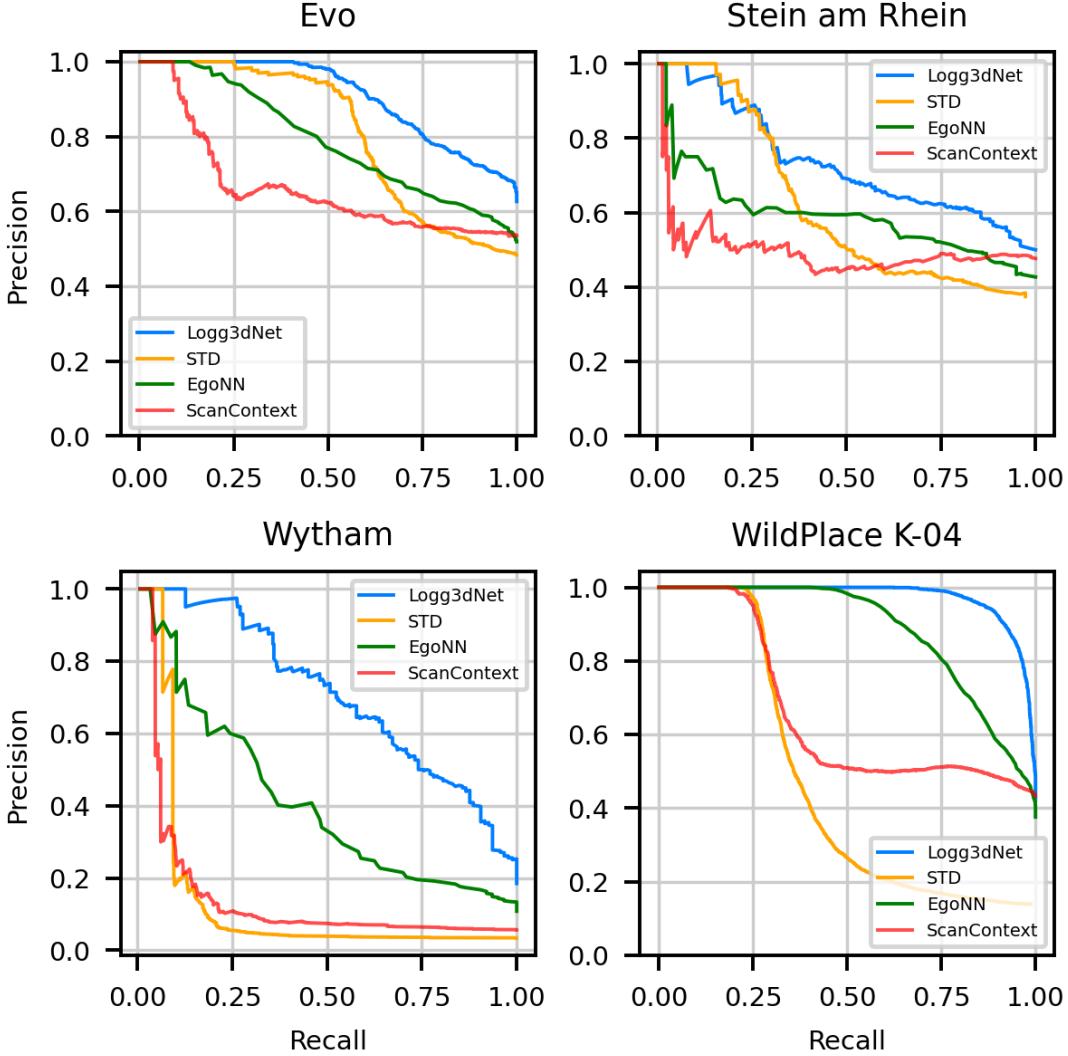


Figure 4.1: Precision-recall curves on four different forest datasets. Evo (Finland, May), Stein am Rhein (Switzerland, Oct), Wytham woods (UK, Feb), WildPlaces [7] (Australia). Evo and Stein am Rhein datasets were collected by Hesai XT32, and Wytham woods dataset was collected by Hesai QT64. Datasets were collected by backpack-LiDAR within dense forests. Only top-1 candidate within 10 m of the ground truth position is regarded as a true positive candidate.

scans increases. Specifically, when scans are more than 10 meters apart, only approximately 50% of RANSAC-registered candidates are successfully verified by ICP. This decrease is due to the diminishing overlap ratio between corresponding scans with increasing distance, making convergence of ICP challenging.

Similarly, in terms of viewpoint difference, most initial candidates have only a 45 degree orientation difference (only counting candidates within 20 m). We note

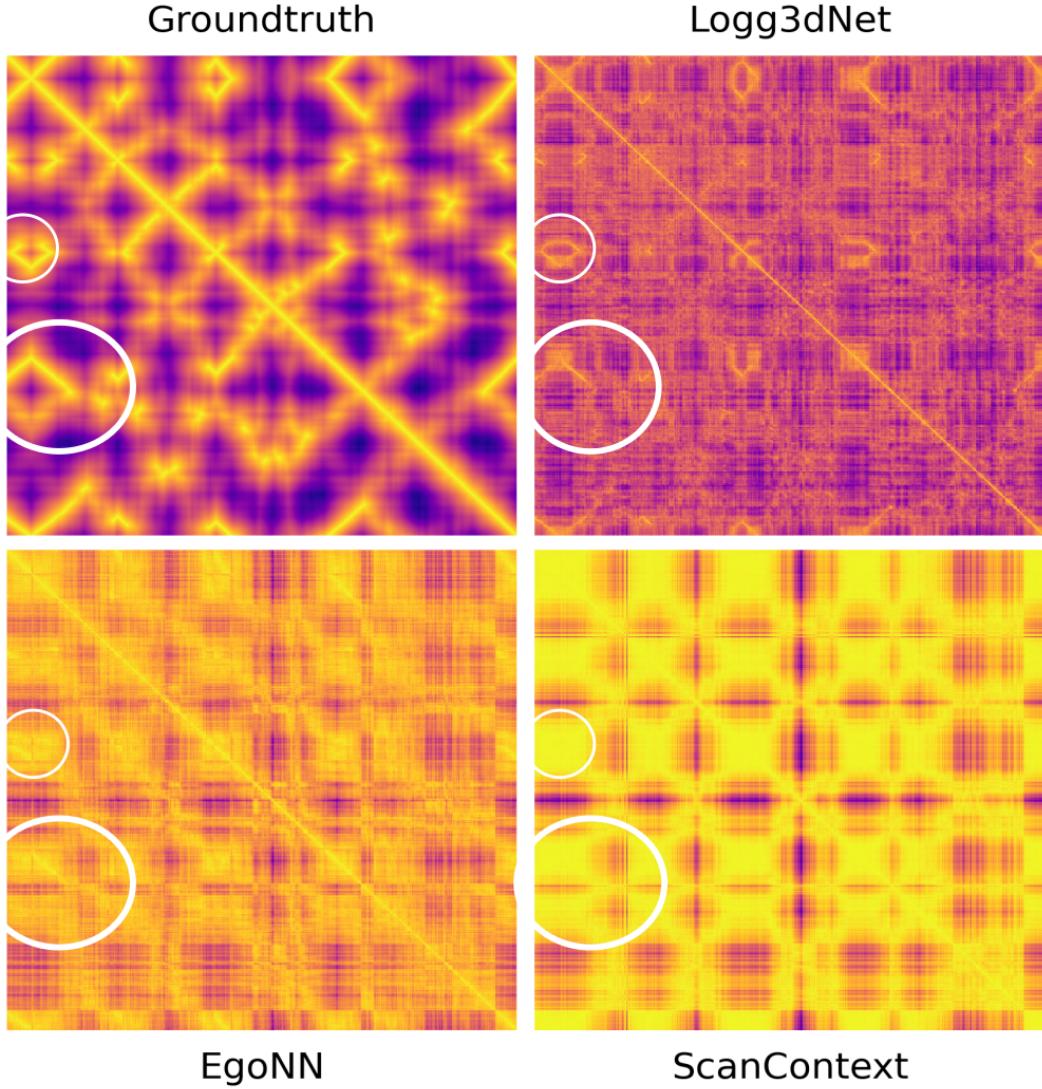


Figure 4.2: Heatmaps depicting descriptor distances for the Evo dataset. Yellow hues denote a high descriptor similarity between scans, whereas purple indicates the low similarity. Patterns more closely resembling the ground-truth (top-left) indicate better descriptor performance. Logg3dNet descriptors show the most similar patterns, whereas ScanContext descriptors are least discriminative among these models. We use τ_s that corresponds to the F_1 -max score in evaluation.

that the proportion of ICP-verified candidates to RANSAC-registered candidates is initially high as $\sim 80\%$ but its proportion decreases $\sim 50\%$ at 90 degree difference. [this is worrying and not explained. It's pretty problematic. We should discuss -THIS IS IMPORTANT. There should be no justification for this but we had this problem with Georgi Tinchev's work also. Furthermore, why doesn't it work with the angle is 180d and also 0d. That's quite bad.]

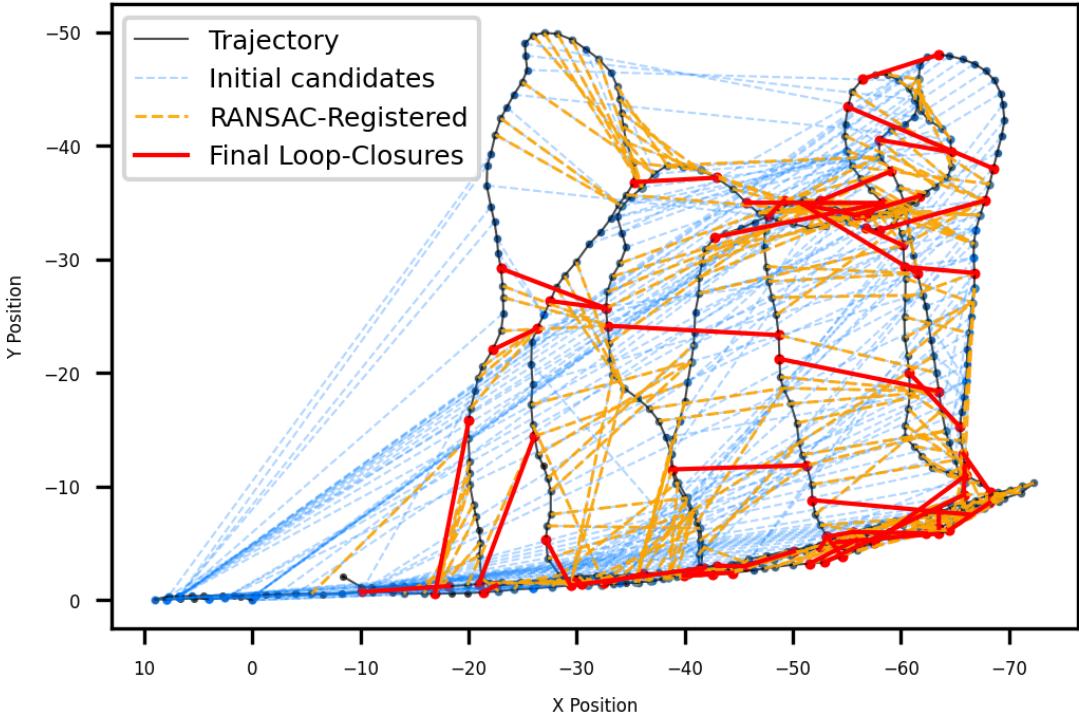


Figure 4.3: Online place recognition with the Evo dataset. Dataset covers about 25 minutes of walk with the backpack mapping system (Hesai-XT32). Bold red lines show the loop closures integrated into the SLAM system, successfully identified up to distance of 17m within dense forest areas.

[I'm interested in the last comment about greater distances. How much?]

4.1.3 Offline Multi-Mission SLAM

In this experiment, we showcase the ability of our approach to obtain loop closures between different mapping missions and to merge those missions into a common map. In this scenario, unlike online-SLAM, we employ stringent descriptor thresholds (τ_s) to highly reduce the number of false positive loop candidates. [why is odometry not available? We literally do store it to file! Reviewers would find the comment confusing] Figure 4.5 presents the results of merging maps from three datasets: Wytham Woods (UK), Evo (Finland), and the Forest of Dean (UK). Each individual mission covers approximately one hectare, with merged map areas ranging from three to five hectares.

We tested the robustness of our place recognition approach by testing different LiDAR sensors and inclinations. The Wytham Woods and Forest of Dean datasets

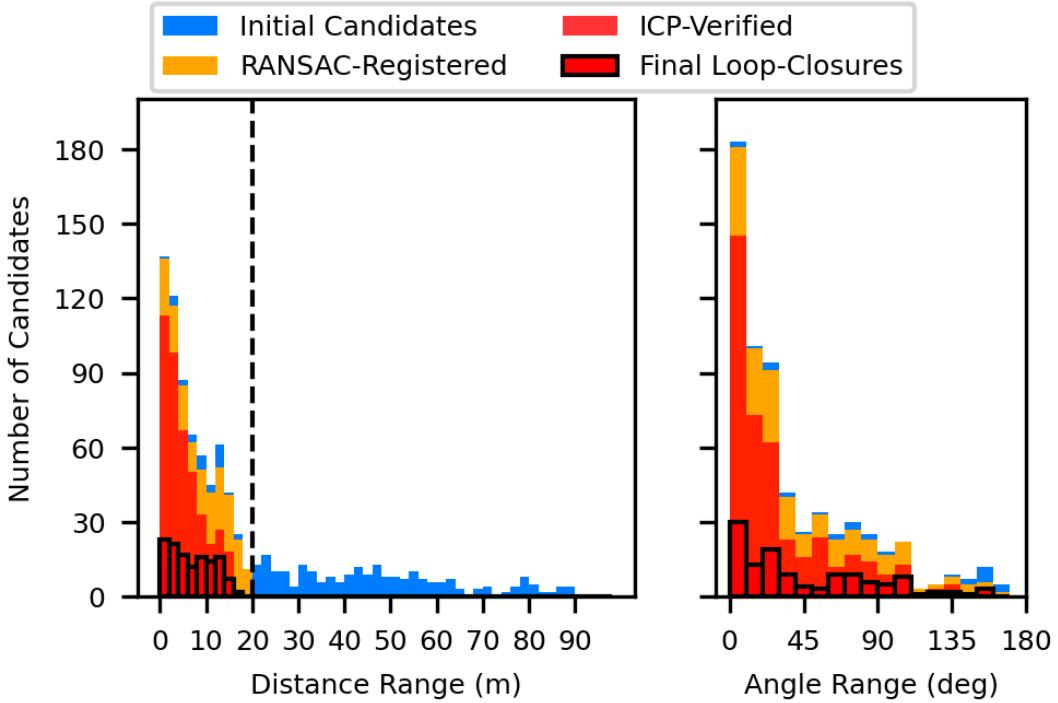


Figure 4.4: Loop closures distribution by distance and angle at various stages of the pipeline. Initial candidates based on descriptor distance are shown in blue. Candidates beyond 20 m are rejected as False Positives using odometry information. Candidates within 20 m undergo RANSAC pre-registration with additional verification using SGV[18] and pairwise checks (yellow). Pre-registered candidates are refined using ICP fine-registration (red), and integrated as constraints in the pose graph. [Haedam: height should be reduced]

were mapped using a Hesai QT64 LiDAR. In contrast, XT32 LiDAR was used and placed at a 45-degree inclination aimed at capturing the forest canopy in Evo. Despite the asymmetry in point clouds caused by this inclination, which primarily captures points in the forward direction, our approach successfully identifies loop closures.

4.1.4 Relocalization

This experiment showcases a relocalization demonstration in a dense forest using our approach. In this scenario, our system demonstrates the ability to continuously track the position within a prior SLAM map once an initial loop closure is established. Figure 4.6 presents an illustrative example of this demonstration,

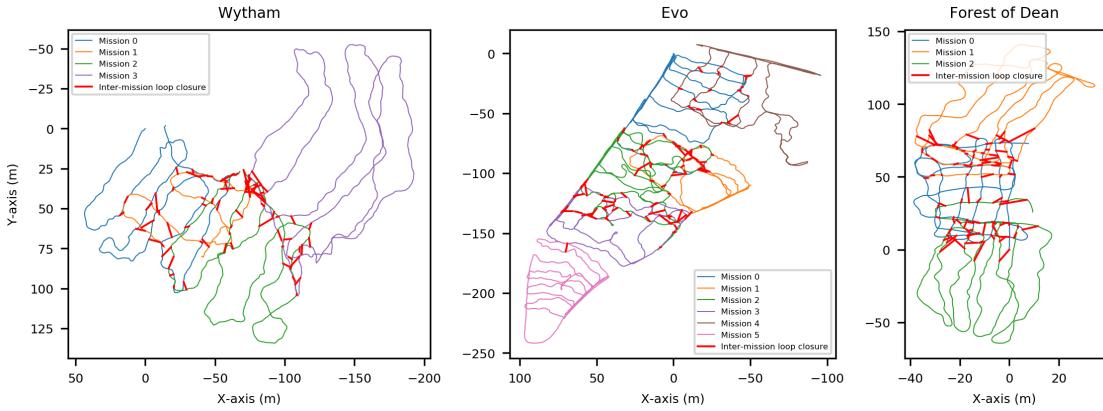


Figure 4.5: Offline multi-mission SLAM. Left: Wytham - a densely wooded area with uneven terrain, including hills. Center: Evo - featuring a LiDAR setup on an incline, with loop closures occurring primarily when viewpoints are closely aligned. Right: Forest of Dean - flatter terrain compared to Wytham, with a sparser plantation, allowing for loop closures to be captured at greater distances.

where the quadruped robot shown in [Fig 1] is localized with respect to a prior SLAM map generated using a backpack LiDAR system. This capability enables the real-time rendering of a virtual forest map overlaid with associated data, such as Diameter at Breast Height (DBH) and species information, onto the camera images. Such feedback is highly beneficial for tasks such as taking forest inventories by foresters or enabling autonomous harvesting.

4.1.5 Study of ICP inlier-based check

Our final experiment investigates the ICP inlier-based check on loop closure integration within the final pose graph optimization process. This analysis is crucial for ensuring that incorrect loop closures are not introduced into the optimized pose graph. Fig. 4.7 illustrates the corrections (on top of the initial transformation prior) as estimated by the ICP registration at various distances. Additionally, the figure color-codes the points based on the number of inlier points obtained during the registration process, with blue indicating a large number of inliers and red indicating a smaller number. Our observation suggests that loop closures occurring beyond 10 m, which propose a substantial transformation correction often have fewer inlier ICP points and are thus less reliable. Based on this analysis, we establish an inlier threshold to ensure that corrections are limited to ICP corrections of less than 1

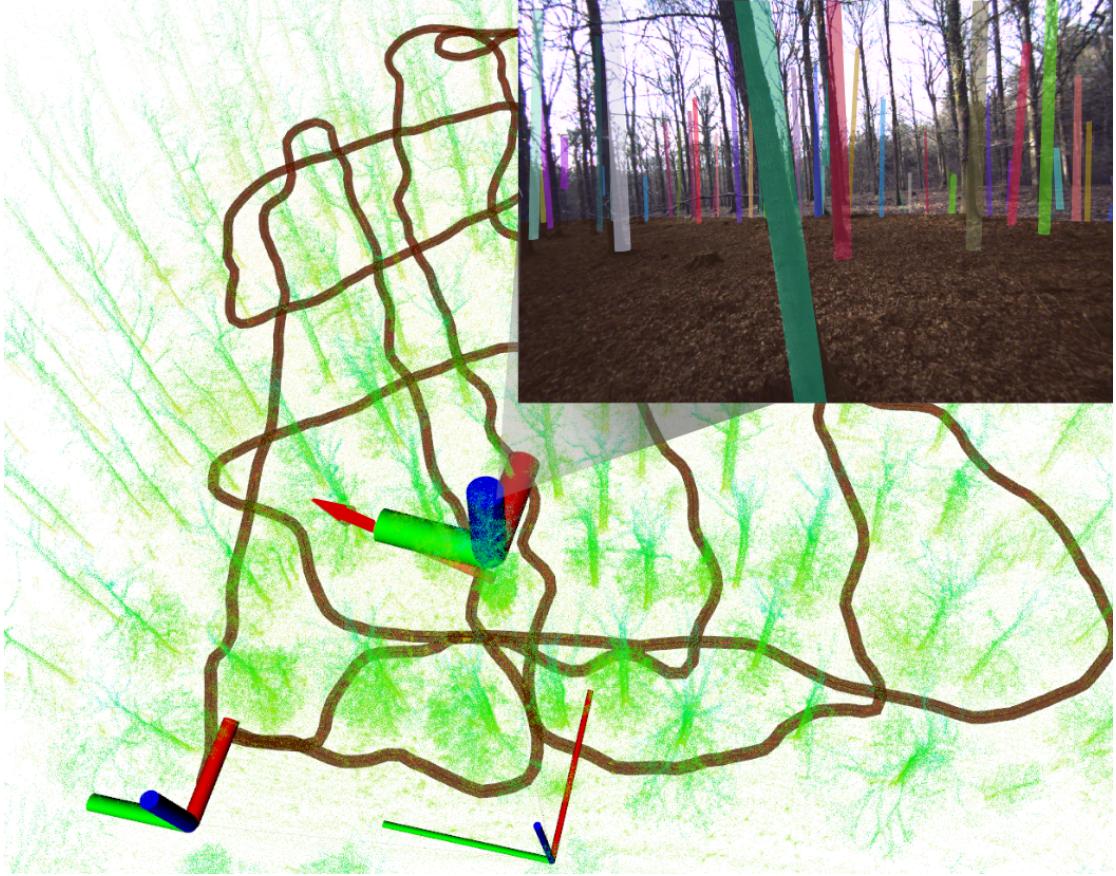


Figure 4.6: Demonstration of relocalization capability. LiDAR (illustrated by large marker) is relocalized in a prior map. We render a virtual view of the forest digital map synchronized with images from our camera (right). Red arrow shows a successful localization at that pose.

meter (shown in dotted red line). This threshold further reduces the number of incorrect loop closures from being integrated into the pose graph.

[I'd like to discuss this also, but it reads fine]

[this caption is too brief for a reader to parse what is meant by all the variables.]

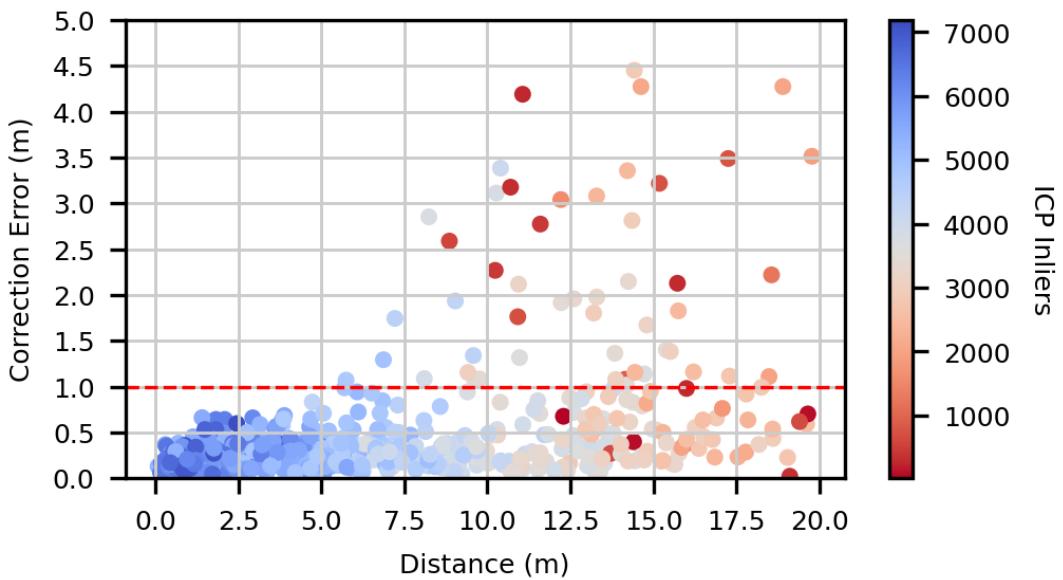


Figure 4.7: Analysis of final ICP registration check. X-axis shows the distribution of loop-candidates by distance after ICP. Y-axis shows ICP correction error w.r.t. the coarse-registration from RANSAC. Color indicates the number of ICP inliers, 30 iterations and RMSE= 0.01m, where clouds are cropped 50 by 50m, then downsampled to 20k points. [Haedam: reduce size]

5

Conclusion

Bibliography

Bibliography

- [1] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. “NetVLAD: CNN Architecture for Weakly Supervised Place Recognition”. In: 40.6 (2018), pp. 1437–1451. DOI: [10.1109/TPAMI.2017.2711011](https://doi.org/10.1109/TPAMI.2017.2711011) (page 5).
- [2] Paul J. Besl and Neil D. McKay. “A Method for Registration of 3-D Shapes”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 14.2 (1992), pp. 239–256. DOI: [10.1109/34.121791](https://doi.org/10.1109/34.121791). URL: <https://doi.org/10.1109/34.121791> (page 10).
- [3] Martin A. Fischler and Robert C. Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. In: *Commun. ACM* 24.6 (1981), 381–395. ISSN: 0001-0782. DOI: [10.1145/358669.358692](https://doi.org/10.1145/358669.358692). URL: <https://doi.org/10.1145/358669.358692> (page 9).
- [4] Giseop Kim, Sunwook Choi, and Ayoung Kim. “Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments”. In: (2021) (page 4).
- [5] Giseop Kim and Ayoung Kim. “Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map”. In: IEEE. 2018, pp. 4802–4809 (pages 4, 9).
- [6] Giseop Kim, Yeong Sang Park, Younghun Cho, Jinyong Jeong, and Ayoung Kim. “MulRan: Multimodal Range Dataset for Urban Place Recognition”. In: 2020, pp. 6246–6253. DOI: [10.1109/ICRA40945.2020.9197298](https://doi.org/10.1109/ICRA40945.2020.9197298) (page 5).
- [7] Joshua Knights, Kavisha Vidanapathirana, Milad Ramezani, Sridha Sridharan, Clinton Fookes, and Peyman Moghadam. “Wild-Places: A Large-Scale Dataset for Lidar Place Recognition in Unstructured Natural Environments”. In: IEEE. 2023, pp. 11322–11328 (pages 5, 18).
- [8] Jacek Komorowski. “MinkLoc3D: Point Cloud Based Large-Scale Place Recognition”. In: 2021, pp. 1789–1798. DOI: [10.1109/WACV48630.2021.00183](https://doi.org/10.1109/WACV48630.2021.00183) (page 5).
- [9] Jacek Komorowski, Monika Wysoczanska, and Tomasz Trzcinski. “EgoNN: Egocentric Neural Network for Point Cloud Based 6DoF Relocalization at the City Scale”. In: 7.2 (2022), pp. 722–729. DOI: [10.1109/LRA.2021.3133593](https://doi.org/10.1109/LRA.2021.3133593) (pages 4, 5, 9).
- [10] Lin Li, Xin Kong, Xiangrui Zhao, Tianxin Huang, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. “SSC: Semantic Scan Context for Large-Scale Place Recognition”. In: 2021, pp. 2092–2099. DOI: [10.1109/IROS51168.2021.9635904](https://doi.org/10.1109/IROS51168.2021.9635904) (page 4).
- [11] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. “1 year, 1000 km: The oxford robotcar dataset”. In: 36.1 (2017), pp. 3–15 (page 5).

- [12] François Pomerleau, Francis Colas, Roland Siegwart, and Stéphane Magnenat. “Comparing ICP Variants on Real-World Data Sets”. In: *Autonomous Robots* 34.3 (Feb. 2013), pp. 133–148 (page 10).
- [13] Alexander Proudman, Milad Ramezani, Sundara Tejaswi Digumarti, Nived Chebrolu, and Maurice Fallon. “Towards real-time forest inventory using handheld LiDAR”. In: 157 (2022), p. 104240. ISSN: 0921-8890. DOI: <https://doi.org/10.1016/j.robot.2022.104240>. URL: <https://www.sciencedirect.com/science/article/pii/S0921889022001397> (page 7).
- [14] Filip Radenović, Giorgos Tolias, and Ondřej Chum. “Fine-Tuning CNN Image Retrieval with No Human Annotation”. In: 41.7 (2019), pp. 1655–1668. DOI: [10.1109/TPAMI.2018.2846566](https://doi.org/10.1109/TPAMI.2018.2846566) (page 5).
- [15] Georgi Tinchev, Simona Nobili, and Maurice Fallon. “Seeing the Wood for the Trees: Reliable Localization in Urban and Natural Environments”. In: 2018, pp. 8239–8246. DOI: [10.1109/IROS.2018.8594042](https://doi.org/10.1109/IROS.2018.8594042) (page 5).
- [16] Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. “Tartandrive: A large-scale dataset for learning off-road dynamics models”. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 2546–2552 (page 5).
- [17] Kavisha Vidanapathirana, Peyman Moghadam, Ben Harwood, Mumeng Zhao, Sridha Sridharan, and Clinton Fookes. “Locus: Lidar-based place recognition using spatiotemporal higher-order pooling”. In: IEEE. 2021, pp. 5075–5081 (page 5).
- [18] Kavisha Vidanapathirana, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. “Spectral Geometric Verification: Re-Ranking Point Cloud Retrieval for Metric Localization”. In: 8.5 (2023), pp. 2494–2501 (pages 10, 21).
- [19] Kavisha Vidanapathirana, Milad Ramezani, Peyman Moghadam, Sridha Sridharan, and Clinton Fookes. “LoGG3D-Net: Locally guided global descriptor learning for 3D place recognition”. In: IEEE. 2022, pp. 2215–2221 (pages 4, 5, 9).
- [20] Han Wang, Chen Wang, and Lihua Xie. “Intensity Scan Context: Coding Intensity and Geometry Relations for Loop Closure Detection”. In: 2020, pp. 2095–2101 (page 4).
- [21] David Wisth, Marco Camurri, and Maurice Fallon. “VILENS: Visual, Inertial, Lidar, and Leg Odometry for All-Terrain Legged Robots”. In: *IEEE Transactions on Robotics* 39.1 (2023), pp. 309–326. DOI: [10.1109/TR.2022.3193788](https://doi.org/10.1109/TR.2022.3193788) (page 7).
- [22] Yan Xia, Yusheng Xu, Shuang Li, Rui Wang, Juan Du, Daniel Cremers, and Uwe Stillä. “SOE-Net: A Self-Attention and Orientation Encoding Network for Point Cloud Based Place Recognition”. In: 2021, pp. 11348–11357 (page 5).
- [23] Tian-Xing Xu, Yuan-Chen Guo, Zhiqiang Li, Ge Yu, Yu-Kun Lai, and Song-Hai Zhang. “TransLoc3D: Point cloud based large-scale place recognition using adaptive receptive fields”. In: (2021) (page 5).
- [24] Chongjian Yuan, Jiarong Lin, Zuhao Zou, Xiaoping Hong, and Fu Zhang. “STD: Stable Triangle Descriptor for 3D place recognition”. In: IEEE. 2023, pp. 1897–1903 (pages 4, 9).

- [25] Wenxiao Zhang and Chunxia Xiao. “PCAN: 3D attention map learning using contextual information for point cloud based retrieval”. In: 2019, pp. 12436–12445 (page 5).
- [26] Zhicheng Zhou, Cheng Zhao, Daniel Adolfsson, Songzhi Su, Yang Gao, Tom Duckett, and Li Sun. “NDT-Transformer: Large-Scale 3D Point Cloud Localisation using the Normal Distribution Transform Representation”. In: 2021, pp. 5654–5660. DOI: [10.1109/ICRA48506.2021.9560932](https://doi.org/10.1109/ICRA48506.2021.9560932) (page 5).