Tree-Loc: 6-DoF LiDAR Global Localization in Forests via Inter-Tree Geometric Matching

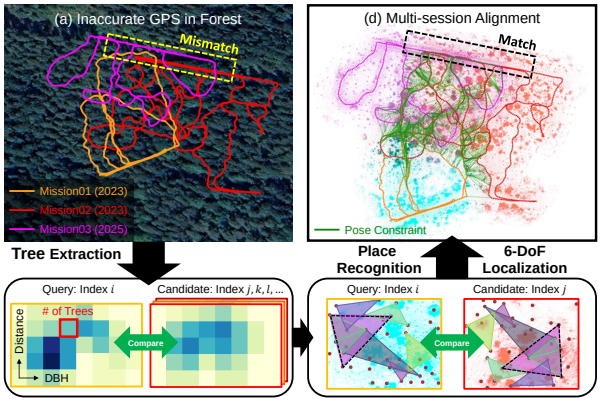
Minwoo Jung[1], Nived Chebrolu[2], Lucas Carvalho de Lima[3], Haedam Oh[2], Maurice Fallon[2] and Ayoung Kim[1*]

***Abstract*— Reliable localization is crucial for navigation in forests, where GPS is often degraded and LiDAR measurements are repetitive, occluded, and structurally complex. These conditions weaken the assumptions of traditional urban-centric localization methods, which assume that consistent features arise from unique structural patterns, necessitating forest-centric solutions to achieve robustness in these environments. To address these challenges, we propose Tree-Loc, a LiDAR-based global localization framework for forests that handles place recognition and 6-DoF pose estimation. We represent scenes using tree stems and their diameter at breast height (DBH), which are aligned to a common reference frame via their axes and summarized using the tree distribution histogram (TDH) for coarse matching, followed by fine matching with a 2D triangle descriptor. Finally, pose estimation is achieved through a two-step geometric verification. On diverse forest benchmarks, Tree-Loc outperforms baselines, achieving precise localization. Ablation studies validate the contribution of each component. We also propose applications for long-term forest management using descriptors from a compact global tree database. Tree-Loc will be open-sourced for the robotics community.**

I. Introduction

Reliable navigation in forests is essential for forestry applications [1] such as digital forest inventory and ecological monitoring. However, complex scenes and unreliable GPS signals cause substantial challenges, leading to inaccurate localization as shown in Fig. 1(a). In these situations, LiDAR-based global localization [2], which uses pre-built maps to estimate global poses, is often employed due to its effectiveness in GPS-denied environments. Nevertheless, these approaches [3] still struggle in forests due to ambiguous structures, seasonal changes, and the storage burden of maintaining dense point cloud maps over long periods.

Despite the need for robust solutions, forest localization is less well studied, with most research [4, 5] focusing on structured urban environments. While existing urban localization methods have been considered for forests, descriptors designed for such environments often suffer from reduced performance in this domain. For instance, global descriptors



(b) Coarse Retrieval using TDH Descriptor (c) Fine Retrieval using 2D Triangle Descriptor

Fig. 1: (a) GPS bias in forests results in misalignment of different LiDAR SLAM trajectories. (b) Tree-Loc performs coarse retrieval using TDH descriptors summarizing tree counts by location and DBH. (c) Fine retrieval with 2D triangle descriptors from tree centers (brown circles); eight matched triangles are highlighted in the same colors. (d) A precise 6-DoF pose enables successful multi-session alignment across datasets captured over different years.

[6, 7] tend to become ambiguous in dense foliage, while local descriptors [8–10] suffer from keypoint inconsistency as the contours of a trunk are hidden under heavy occlusion. These limitations highlight the need for a forest-specific approach that uses stable and interpretable object-level features, such as tree stems, which are standard elements in digital forest inventories. These compact representations are more storage-efficient and easier to maintain than raw point clouds.

Recently, forest localization has achieved more attention with the development of dedicated datasets [11–13] and applications [14, 15]. Unfortunately, many approaches rely on learned features [16–18] or segment-level inputs [19], requiring intensive computation that limits real-time deployment on resource-constrained hardware. Moreover, segment-based methods [19, 20] generate one descriptor per point cloud-level cluster, often capturing non-structural elements like twigs and lacking inter-tree geometry. These descriptors are also sensitive to appearance changes (i.e., foliage growth over time) and require storing large point clouds, making them unsuitable for long-term management. To address these issues, we adopt a learning-free, stem-centric representation that models each tree with a small set of parameters: 3D axis, base location, and diameter at breast height (DBH). These parameters are actually the required elements for building digital forest inventories, which foresters leverage to develop harvesting and maintenance plans. By using such inventories, we aim to exploit the spatial arrangement of nearby trees to define geometry signatures suitable for robust localization.

Our proposed global localization method for forests, Tree-

†This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. RS-2024-00461409), Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT)(No.2022-0-00480) and the Horizon Europe project DigiForest (101070405).

[1]M. Jung and A. Kim are with the Dept. of Mechanical Engineering, SNU, Seoul, S. Korea [moonshot, ayoungk]@snu.ac.kr

[2]N. Chebrolu, H. Oh and M. Fallon are with the Oxford Robotics Institute, University of Oxford, UK {nivedc, haedam, mfallon}@robots.ox.ac.uk

[3]L. Carvalho de Lima is with the Department of Computer Science, University of XYZ lucas@xyz.edu

Loc, is motivated by a perhaps obvious observation: although changes in seasonal foliage growth are significant, the spatial location of the trees themselves does not change, making stem-based features well-suited for persistent mapping. The pipeline first aligns the scene to a consistent local frame using tree orientations, enabling robust 2D inter-tree geometry despite viewpoint variations. It then performs a two-stage search: (i) a tree distribution histogram (TDH) encodes tree counts by DBH and distance into a global descriptor for fast retrieval; (ii) 2D triangle descriptors from tree centers refine candidates by capturing inter-tree geometry. These descriptors and geometric cues, like tree axes and base heights, enable 6-DoF pose estimation without additional feature extraction. Our main contributions are as follows:

- A global localization method for forests that leverages robust and repeatable parametric tree features and inter-tree geometry to estimate accurate 6-DoF poses with a compact, learning-free representation.
- A dual descriptor pipeline: TDH for coarse retrieval and a 2D triangle descriptor using tree centers for fine verification, both stabilized by tree-based scene alignment for rapid candidate retrieval and reliable matching.
- 6-DoF pose estimation without additional feature extraction beyond place recognition, relying solely on 2D triangle geometry and tree measurements to minimize computational overhead.
- Strong performance over state-of-the-art (SOTA) methods, with accuracy comparable to learning-based methods. It enables fast localization within $50\ \mathrm{ms}$, and supports multi-session map maintenance with a compact representation, 3-4 orders of magnitude smaller than methods using dense point clouds or descriptor maps.

II. Related Work

A. Forestry Applications with LiDAR

LiDAR has become a key tool in forest inventory [1], enabling individual tree segmentation and attribute measurement, such as height and DBH, using algorithmic [14] or deep-learning methods [15]. Public forest datasets [3, 11–13] have further accelerated progress by providing standardized benchmarks. Recently, research on forest-specific localization has grown, leveraging learned descriptors [16–19] and segment-level inputs [19, 20]. ForestLPR [16] extracts multi-channel BEV features from height bins but remains sensitive to clutter in the undergrowth and discretization choices. FGLoc6D [17] employs EgoNN [21] with 3D semantic supervision for cross-view retrieval, which depends on segmentation masks. Segment-based methods such as ESM [19] and NSM [20] exhibit low temporal robustness to seasonal and viewpoint changes, as they include non-tree elements like foliage alongside trees in the segmented point cloud and overlook inter-tree relationships. Moreover, these learning-based methods often incur computational overhead that hinders real-time deployment on resource-constrained platforms. In contrast, our approach leverages RealtimeTrees [14] to directly extract key forest inventory components, such as individual trees and DBH, enabling learning-free, stem-centric features for efficient, real-time forest localization.

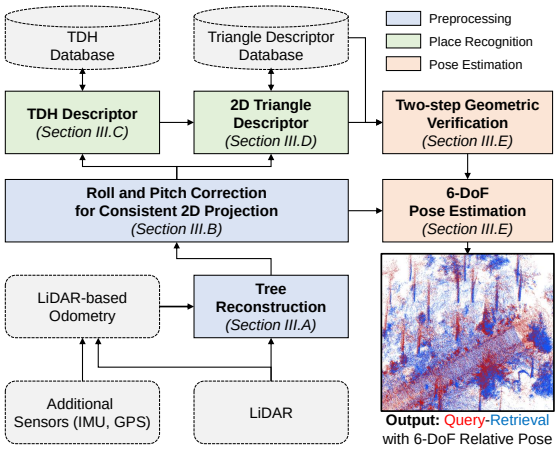


Fig. 2: Tree-Loc converts LiDAR scans into tree instances. After performing place recognition using two descriptors, it leverages existing data to perform 6-DoF pose estimation. At the end of the pipeline, the point cloud is tightly registered without ICP.

B. LiDAR-based Global Localization

Global localization is typically divided into two stages: place recognition and pose estimation, both of which rely on global or local descriptors. Global descriptors [6, 22, 23] compress point clouds into low-dimensional representations for fast retrieval. Scan Context [6, 22] encodes maximum height on a Bird's-Eye-View (BEV) grid, while RING++ [7] enhances rotational invariance using the Radon transform. These methods are usually computationally lightweight but struggle in forests, as repetitive and ambiguous structures can cause over- or under-representation of salient cues, resulting in inadequate information for precise pose estimation.

Local descriptor-based methods aim to overcome these limitations by focusing on sparse yet distinctive geometric cues that facilitate fine-grained pose estimation. MapClosure [8] extracts ORB features [24] from BEV images, enabling feature-based matching. STD [25] and BTC [9] build hash tables from triangle descriptors defined by three keypoints, offering rotation and translation invariance in urban environments. ITD [10] and RE-TRIP [26] leverage LiDAR intensity and reflectivity to enhance the distinctiveness of these keypoints. However, in the forest, these local approaches face several challenges: structural noise from undergrowth, heavy occlusion caused by tree trunks, and a lack of high-intensity objects. These factors degrade keypoint repeatability and descriptor discriminability, leading to failures in both place recognition and pose estimation. To address this, we generate a dual descriptor using tree stems and their DBH, enabling robust performance and fast retrieval. Moreover, projecting the tree layout onto a 2D plane ignores vertical ambiguity. Nevertheless, it still supports accurate 6-DoF pose estimation with tree axes and base heights, without requiring an expensive terrain model or DEM (Digital Elevation Model).

III. Methodology

An overview of the Tree-Loc pipeline is presented in Fig. 2, comprising three main stages: preprocessing, place

recognition via a dual descriptor, and 6-DoF pose estimation.

A. Tree Reconstruction

To enable robust tree reconstruction in forest environments, our pipeline aggregates a number of k LiDAR scans into a single payload to achieve sufficient point density, following recent submap-based LiDAR place recognition [9, 25]. Each payload shares a number of v scans with the previous one, ensuring the overlap with adjacent payloads. Payloads are only accumulated when in motion to ensure meaningful aggregation. The poses for these scans can be obtained from LiDAR-based SLAM [27, 28], which provides local consistency between scans. Let $\mathcal{P}_u$ denote the point cloud payload at index u. We define a window centered at t as $\mathcal{W}_t = \{\, t - \lfloor (s-1)/2 \rfloor, \ldots, t + \lfloor (s-1)/2 \rfloor \,\}$, where a set of s payloads in a sliding window are expressed in the local frame of the index t. The aggregated payloads in the center frame:

$$\mathcal{Z}_t \;=\; \bigcup_{u \in \mathcal{W}_t} \mathbf{T}_{t \leftarrow u}\, \mathcal{P}_u, \tag{1}$$

where $\mathbf{T}_{t \leftarrow u}$ is the pose from frame u to the center frame t.

We reconstruct trees from the aggregated payloads $\mathcal{Z}_t$ using RealtimeTrees [14]. The method first estimates a local terrain surface via a cloth simulation filter and height-normalizes the cloud to suppress slope-induced variance. On the normalized cloud, a Voronoi-inspired clustering method segments vertically coherent stem structures by enforcing spatial compactness in the xy-plane and continuity along the z-axis. Stems meeting these criteria are classified as reconstructed trees; others are candidate stems. For each reconstructed tree, the pipeline recovers (i) the 3D tree axis from principal analysis, (ii) the stem center, (iii) DBH from a circular fit on a near-ground slice, and (iv) the base point from tree–terrain intersection. More details are available in RealtimeTrees [14]. This procedure runs per window $\mathcal{W}_t$.

We summarize the result at index t as $\mathbf{M}_t$, a compact tuple of the pose and a per-tree attribute set, tree database, that serves as the input to TDH and 2D triangle descriptor:

$$\mathbf{M}_t = (\mathbf{T}_t,\ \mathcal{I}_t), \qquad \mathcal{I}_t = \big\{\, (\mathbf{A}_j,\ \mathbf{p}'_j,\ d_j) \,\big\}_{j=1}^{n_t}. \tag{2}$$

$\mathbf{T}_t$ is the pose at index t, $\mathbf{A}_j \in \mathrm{SO}(3)$ is the tree axis orientation, $\mathbf{p}'_j \in \mathbb{R}^3$ is the 3D stem position, combining its center's x, y coordinates ($\mathbf{c}'_j$) with the base height b'_j (the z-coordinate of the tree-ground contact), and d_j is the DBH. If reconstructed trees are insufficient for generating enough descriptors, we supplement them with candidate stems selected by observation count across payloads.

B. Roll and Pitch Correction for Consistent 2D Projection

We project the trees onto a 2D plane as a preprocessing step to generate descriptors for place recognition using $\mathbf{M}_t$. Projecting 3D tree structures onto 2D ignores height ambiguity, focusing on stable inter-tree geometry for localization. However, viewpoint-induced roll and pitch variations distort tree center projections on their local xy-plane, causing inconsistent center relations. Basic ground-plane fitting can be used for correction, but it is unreliable on uneven terrain. To address this, we align scenes to a fixed reference direction using tree axes, estimating a rotation to correct roll and pitch while excluding yaw to ensure consistent 2D projections. For each tree j, we use the unit vector $\mathbf{a}_j$, the third column of $\mathbf{A}_j$. The optimal rotation $\mathbf{R}_t^A \in \mathrm{SO}(3)$ is estimated using RANSAC-based robust least-squares optimization, aligning $\{\mathbf{a}_j\}$ to a reference direction. Any unit vector can be used as direction, but we select $\mathbf{e}_z = (0,0,1)^\top$ for convenience:

$$\mathbf{R}_t^A = \underset{\mathbf{R} \in \mathrm{SO}(3)}{\arg\min} \sum_j \Big(1 - \big|\mathbf{e}_z^\top \mathbf{R} \mathbf{a}_j\big|\Big)^2. \tag{3}$$

$\mathbf{R}_t^A$ aligns each frame to a common reference, ensuring consistent 2D projection across viewpoints. Even under an additional rotation $\mathbf{R}_{\text{view}}$, with $\bar{\mathbf{a}}_j = \mathbf{R}_{\text{view}} \mathbf{a}_j$, compensation is achieved with $\mathbf{R}_t^A \mathbf{R}_{\text{view}}^{-1}$ via (3). As $\mathbf{R}_t^A$ encodes roll and pitch relative to the reference frame, it also contributes to 6-DoF pose estimation. Finally, the stem positions are rotated as $\mathbf{p}_j = \mathbf{R}_t^A \mathbf{p}'_j$ and projected onto the plane perpendicular to the reference direction (local xy-plane) as

$$\mathbf{c}_j = \Pi_{\perp \mathbf{e}_z}(\mathbf{p}_j) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \big(\mathbf{p}_j - (\mathbf{e}_z^\top \mathbf{p}_j)\mathbf{e}_z\big)\,. \tag{4}$$

C. Tree Distribution Histogram Descriptor

We generate a tree distribution histogram (TDH) from the tree database at index t to summarize the spatial layout for fast retrieval. Using aligned 2D centers $\{\mathbf{c}_j\}$ and their DBH values $\{d_j\}$, we assign each tree to a radial distance bin (2D distance from the origin in the aligned frame) and a DBH bin, as illustrated in Fig. 1(b).

Radial bins are defined with a fixed resolution r_{res} and boundary overlap w_{range}. DBH spans $[r_{\min}, r_{\max}]$ and is divided into n_{sections} bins of width w_{dbh} with overlapping intervals. These overlaps mitigate measurement noise and minor pose errors. The resulting histogram $\mathbf{H}_t \in \mathbb{R}^{n_{\text{spatial}} \times n_{\text{sections}}}$ counts the number of trees in each radial and DBH bin.

To enhance stability, $\mathbf{H}_t$ is smoothed using a 2×2 uniform filter, leveraging adjacency among neighboring bins. A 40-dimensional descriptor is obtained with $n_{\text{spatial}} = 5$ and $n_{\text{sections}} = 8$. For comparison, the chi-square distance between histograms is computed to rank database candidates. Thanks to its compact design, TDH enables fast retrieval, and the top 100 candidates are passed to the fine retrieval stage.

D. 2D Triangle Descriptor Using Tree Centers

To refine TDH matches, we use a 2D triangle descriptor derived from projected centers, which exhibits rotation and translation invariance [9], as shown in Fig. 1(c). For each scene $\mathcal{I}_s$, given the refined center set $\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{n_t}\}$, every unordered triple (i, j, k) defines a descriptor

$$\mathbf{D}_{ijk} = \big\{\mathbf{c}_i, \mathbf{c}_j, \mathbf{c}_k,\ \ell_{ij}, \ell_{jk}, \ell_{ik},\ \mathbf{q}_{ijk}, i\big\}, \tag{5}$$

where $\ell_{ij} = \|\mathbf{c}_i - \mathbf{c}_j\|_2$ are the side lengths and $\mathbf{q}_{ijk}$ is the triangle centroid; i denotes the scene index.

Permutation invariance is achieved by sorting the side lengths in ascending order and hashing the sorted tuple to form a key: $h_{ijk} = \mathrm{hash}(\mathrm{sort}(\ell_{ij}, \ell_{jk}, \ell_{ik}))$. Each scene stores its keys and descriptors in a hash table $\mathcal{H}_s$. The

similarity between a query Q and a candidate C is defined as: $S(Q, C) = |\mathcal{K}_Q \cap \mathcal{K}_C|$, computed as the number of shared keys, where $\mathcal{K}_Q$ and $\mathcal{K}_C$ represent the set of keys extracted from $\mathcal{H}_Q$ and $\mathcal{H}_C$. This integer lookup-based comparison is more discriminative than TDH, enabling fine retrieval across hundreds of 2D triangle descriptors. The ten candidates with the highest $S(Q, C)$ are passed to geometric verification.

E. Geometric Verification with 6-DoF Localization

We conduct a two-step geometric verification for each query scene $\mathcal{I}_Q$ and each of the ten candidate scenes $\mathcal{I}_C$, calculating a 4-DoF transformation and ranking candidates by geometric consistency, as illustrated in Fig. 3.

Initial Planar Alignment: For each pair of triangles with matching hash keys, we identify a centroid correspondence $(\mathbf{q}_u^Q, \mathbf{q}_u^C)$. Given N such pairs, we estimate a planar transform $\mathbf{T}^c_{C \leftarrow Q} = (\mathbf{R}_c, \mathbf{t}_c) \in \mathrm{SE}(2)$ via SVD-based alignment:

$$\min_{\mathbf{R}_c, \mathbf{t}_c} \sum_{u=1}^{N} \left\| \mathbf{q}_u^C - (\mathbf{R}_c \mathbf{q}_u^Q + \mathbf{t}_c) \right\|^2. \tag{6}$$

Refined Alignment and Vertical Offset: let $\mathcal{C}_Q = \{\mathbf{c}_i\}$ and $\mathcal{C}_C = \{\mathbf{c}_j\}$ be the aligned tree centers, with $\mathcal{B}_Q = \{b_i\}$ and $\mathcal{B}_C = \{b_j\}$ as their base heights. We transform query centers to $\widetilde{\mathbf{c}}_i = \mathbf{R}_c \, \mathbf{c}_i + \mathbf{t}_c$. Using a KD-tree built on $\mathcal{C}_C$, we find nearest neighbors for $\widetilde{\mathbf{c}}_i$, retaining pairs with planar distance below $0.4\,\mathrm{m}$ and DBH difference below $0.2\,\mathrm{m}$. For these pairs, we compute the vertical offset Δz by minimizing the base height differences $|b_i - b_j - \Delta z|$ via RANSAC.

If at least two pairs remain, we perform a second SVD alignment on the matched centers $\widetilde{\mathbf{c}}_i$ and $\mathbf{c}_j$, yielding a refined transform $\mathbf{T}^f_{C \leftarrow Q} = (\mathbf{R}_f, \mathbf{t}_f) \in \mathrm{SE}(2)$, where $\mathbf{R}_f \in \mathrm{SO}(2)$ and $\mathbf{t}_f \in \mathbb{R}^2$, using (6) with $\widetilde{\mathbf{c}}_i$ and $\mathbf{c}_j$ as inputs instead of $\mathbf{q}_u^Q$ and $\mathbf{q}_u^C$. The 4-DoF transformation in $\mathrm{SE}(3)$ is composed as $\mathbf{T}^{4D}_{C \leftarrow Q} = (\mathbf{R}_{4D}, \mathbf{t}_{4D})$, where $\mathbf{R}_{4D} \in \mathrm{SO}(3)$ extends the in-plane rotation $\mathbf{R}_f \mathbf{R}_c$ with roll and pitch set to zero, and $\mathbf{t}_{4D} = \left[(\mathbf{R}_f \mathbf{t}_c + \mathbf{t}_f)^\top \quad \Delta z \right]^\top$.

Geometric Consistency and Ranking: Let $\mathcal{M}_{Q,C}$ denote the set of matching tree pairs supporting the refined transform, with $|\mathcal{T}_Q|$ and $|\mathcal{T}_C|$ as the number of trees in each scene. The overlap ratio quantifies geometric consistency:

$$\mathcal{O}(Q, C) = \frac{|\mathcal{M}_{Q,C}|}{|\mathcal{T}_Q| + |\mathcal{T}_C| - |\mathcal{M}_{Q,C}|}. \tag{7}$$

The ten candidates are ranked by $\mathcal{O}(Q, C)$. For the highest scoring candidate, the alignment rotations $\mathbf{R}_Q^A$ and $\mathbf{R}_C^A$ are applied to compute a 6-DoF transformation. This yields $\mathbf{T}^{6D}_{C \leftarrow Q} = (\mathbf{T}_C^A)^{-1} \, \mathbf{T}^{4D}_{C \leftarrow Q} \, \mathbf{T}_Q^A \in \mathrm{SE}(3)$, where $\mathbf{T}_Q^A$ and $\mathbf{T}_C^A$ are the pure rotational transformations induced by $\mathbf{R}_Q^A$ and $\mathbf{R}_C^A$. Although place recognition is constrained to a 2D plane, the 6-DoF transformation $\mathbf{T}^{6D}_{C \leftarrow Q}$ can still be estimated using only tree information, enabling effective loop closure.

IV. Experiment

A. Datasets and Evaluation Metrics

We evaluated Tree-Loc on the Oxford Forest Place Recognition dataset [11] (`Evo` and `Stein am Rhein`, where the backpack recording device has strong roll and pitch motion),

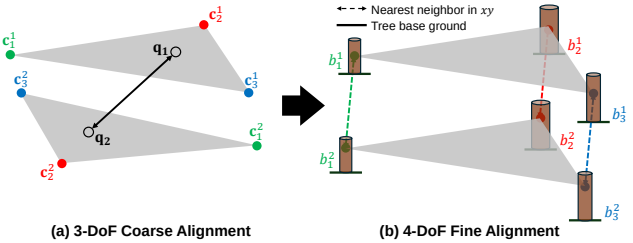


Fig. 3: The coarse stage estimates an initial planar transform from triangle centroids ($\mathbf{q}$) matched between the query and candidate scenes. The fine stage refines this planar transform using 2D tree centers ($\mathbf{c}$) and base heights (b), obtaining the 4-DoF relative pose.

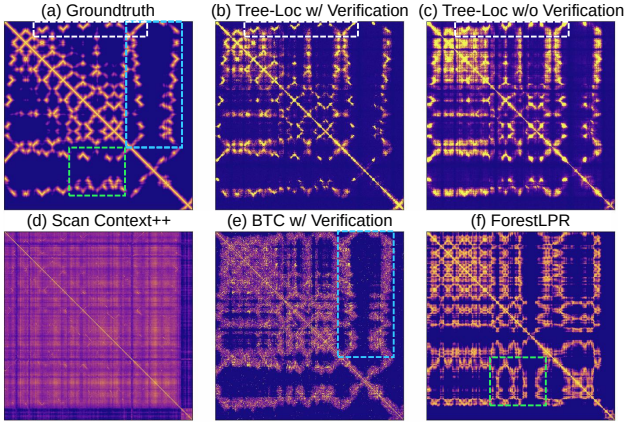


Fig. 4: Feature similarity maps in `Evo` (warmer colors for higher similarity). Tree-Loc focuses similarity in a compact region aligned with the ground truth, enabling clear thresholding, while baselines yield diffuse areas (blue) mixed with false positives (green).

Wild-Places [3] (`Venman01-04` and `Karawatha01-04`, natural forests with long trajectories and seasonal changes), and BotanicGarden [12] (`1008-01`, an urban park). Payloads were built from trajectories using FAST-LIO2 [27] for Oxford and from ground truth for the others. We reported Recall@1 (R@1), maximum F1 score, and AUC under a $10\,\mathrm{m}$ threshold for place recognition. For localization, we used Recall@$50\,\mathrm{cm}$ (R@50), defined as pose estimation with translation error (TE) $\leq 0.5\,\mathrm{m}$ and rotation error (RE) $\leq 5°$, along with the success rate (SR), the ratio of true positives meeting the R@50 criteria. Errors were computed only for successful cases, following the approach used in BTC [9].

Since Tree-Loc exploits multiple payloads to form a tree-based descriptor, we also provided all baselines with accumulated LiDAR scans within a $60\,\mathrm{m} \times 60\,\mathrm{m}$ for both query and database inputs to ensure a fair comparison. We compared against three global descriptor-based methods such as Scan Context++ (SC++) [6], SOLID [23], and RING++ [7], two local descriptor-based methods, BTC [9] and MapClosure [8], and four learning-based methods, TransLoc3D [29], LoGG3D-Net [30], MinkLoc3Dv2 [31], and ForestLPR [16].

B. Intra-session Place Recognition

We first evaluated the intra-session place recognition performance of Tree-Loc across six sequences in Table. I. The most recent 50 queries were excluded from the database to avoid trivial matches arising from temporal proximity. Except for `K-04`, Tree-Loc outperformed all SOTA baselines in

TABLE I: Intra-session place recognition with Multiple Forest Dataset (**Bold** : Best, Normal : Second-Best, V: Venman, K: Karawatha)

Method	Oxford (Evo)			Oxford (Stein)			Wild-Places (V-03)			Wild-Places (V-04)			Wild-Places (K-03)			Wild-Places (K-04)		
	R@1	F1	AUC	R@1	F1	AUC	R@1	F1	AUC	R@1	F1	AUC	R@1	F1	AUC	R@1	F1	AUC
SC++	0.476	0.784	0.883	0.579	0.797	0.854	0.087	0.361	0.321	0.594	0.757	0.670	0.517	0.708	0.734	0.659	0.867	0.906
SOLID	0.428	0.675	0.772	0.527	0.751	0.832	0.081	0.151	0.063	0.434	0.633	0.636	0.324	0.491	0.476	0.554	0.768	0.815
RING++	0.549	0.719	0.847	0.576	0.785	0.878	0.188	0.409	0.303	0.331	0.508	0.305	0.455	0.625	0.510	0.627	0.822	0.875
BTC	0.626	0.804	0.868	0.611	0.774	0.803	0.387	0.660	0.700	0.353	0.661	0.673	0.362	0.596	0.521	0.449	0.747	0.617
MapClosure	0.317	0.481	0.311	0.060	0.114	0.060	0.170	0.291	0.163	0.376	0.546	0.368	0.572	0.728	0.581	**0.756**	0.860	0.792
Tree-Loc	**0.907**	**0.966**	**0.992**	**0.871**	**0.945**	**0.988**	**0.865**	**0.933**	**0.980**	**0.890**	**0.942**	**0.974**	**0.798**	**0.897**	**0.952**	0.694	**0.928**	**0.965**

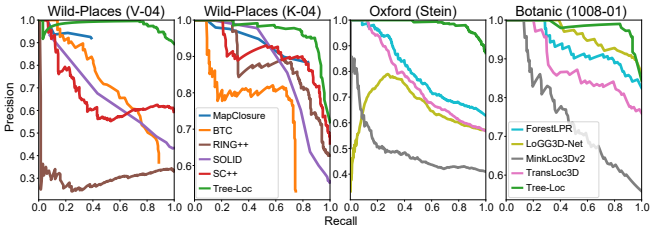


Fig. 5: Precision–Recall curves comparing with algorithmic methods (left) and learning-based methods (right). Tree-Loc achieves a higher AUC and maintains high precision at high recall, thereby avoiding the sharp drop in precision observed in baselines.

R@1, F1, and AUC, with substantial improvements in F1 and AUC. Global descriptor-based methods (SC++, SOLID, RING++) struggled due to weak distinctiveness in forests, often producing similar descriptors for negative scenes. As shown in Fig. 4(d), SC++ is unable to clearly distinguish each place, making threshold selection difficult. Low thresholds allowed many false positives, while high thresholds missed true positives. Local descriptor-based methods (BTC, MapClosure) also underperformed due to the repetitive and irregular forest structure, which degrades feature distinctiveness. Although BTC's similarity map roughly followed the ground truth contour, it produced scattered high-similarity regions near true positives (Fig. 4(e)), reducing localization accuracy.

In contrast, Tree-Loc produces place recognition matches which closely resemble the ground truth, as shown in Fig. 4(a)-(c). Even without a geometric verification, Tree-Loc's descriptor similarity was higher than that of other baselines. While the matched region appeared wider, it was clearly separated from false positives, which simplified the threshold selection process. As shown in Fig. 5, Tree-Loc maintains high precision at high recall, while baselines exhibit a sharp decline in precision. This highlights Tree-Loc's robustness in real-world deployment. Tree-Loc's performance slightly degrades in K-04 due to sparse trees in open spaces, reducing the number of distinct stem-based features for matching. Nevertheless, it outperforms all baselines, confirming the reliability of its tree-based representations.

C. Inter-session Place Recognition

We evaluated inter-session place recognition on Venman and Karawatha sequences 01–04 from Wild-Places, using 24 query-database pairs (12 from each region) and following its protocol for comparing repeated traversals in specific test regions. Sequences were collected on the same day, and 6 and 14 months apart, introducing temporal discrepancies and requiring descriptors that preserve similarity over time. As shown in Table. II, Tree-Loc achieved the highest mean recall, F1, and AUC as well as the lowest standard deviation (σ), demonstrating the temporal stability of the tree-based pipeline. Performance in Karawatha was slightly lower than in Venman due to there being fewer trees in open areas, yet higher F1 and AUC still indicated practical utility.

TABLE III: Localization Performance (Up: 3-DoF, Down*: 6-DoF)

Method	Oxford (Stein)				Wild-Places (K-04)			
	R@50	SR	TE [m]	RE [°]	R@50	SR	TE [m]	RE [°]
RING++	0.227	0.388	0.266	1.031	0.257	0.409	0.281	1.031
MapClosure	0.092	0.806	0.198	0.795	0.411	0.544	0.255	1.089
BTC	0.638	0.780	0.225	0.641	0.548	**0.961**	0.157	0.431
Tree-Loc-f	0.953	0.982	0.100	0.248	0.557	0.765	0.112	0.370
Tree-Loc	**0.970**	**0.987**	**0.053**	**0.137**	**0.653**	0.895	**0.064**	**0.197**
BTC*	0.290	0.398	0.236	1.953	0.414	0.766	0.234	1.576
Tree-Loc-f*	0.876	0.916	0.168	1.055	0.519	0.709	0.199	1.321
Tree-Loc*	**0.908**	**0.940**	**0.126**	**1.013**	**0.609**	**0.832**	**0.153**	**1.192**

In comparison, BTC and MapClosure achieved lower scores because their local descriptors are affected by scene appearance, resulting in less reliable matches. SC++, SOLID, and RING++ often achieved better performance than local descriptor-based methods by compressing scenes, but exhibited larger standard deviations when temporal discrepancies were significant. As shown in Fig. 6, the second-best performing method, SC++, retrieves many true positives on the same day between V-01 and V-02, but failed to detect matches between V-02 and V-03 due to environmental change. Meanwhile, Tree-Loc consistently retrieves correct matches across all time intervals, demonstrating the robustness of the tree-based descriptor to environmental changes.

D. Localization Accuracy

We compared pose estimation across four metrics on three sequences, including K-04, which had the lowest place recognition performance. Since some baselines only supported 3-DoF pose estimation, we reported both 3-DoF and 6-DoF results for Tree-Loc to ensure fair comparison. SC++ was excluded from this evaluation as it only provides yaw angle prediction. An ablation study was also conducted on a variant of Tree-Loc that used only the first step of our two-step geometric verification pipeline, in which triangle centroids $\mathbf{q}$ were used. This is referred to as Tree-Loc-f. As shown in Table. III, Tree-Loc outperformed the baselines in most metrics, achieving centimeter-level accuracy and high success rates, including K-04, even without ICP refinement. RING++'s BEV-based global descriptor was sensitive to viewpoint and resolution, resulting in large translation and rotation errors. MapClosure's ORB features lacked consistency across the forest, hindering robust matching. BTC used triangles and binary vectors, but performed worse than Tree-Loc due to inconsistent observations of triangle vertices across different viewpoints and forest densities.

In 6-DoF evaluations, BTC showed a significant drop in success rate, as tree height variations hindered relative z estimation. In contrast, Tree-Loc maintained strong performance in both 3-DoF and 6-DoF, with lower median error and

TABLE II: Inter-session place recognition performance. Arrows indicate higher/lower is better.

Method	Wild-Places (Venman01-04, 12 query-database pairs)						Wild-Places (Karawatha01-04, 12 query-database pairs)					
	R@1		F1		AUC		R@1		F1		AUC	
	Mean [↑]	σ [↓]	Mean [↑]	σ [↓]	Mean [↑]	σ [↓]	Mean [↑]	σ [↓]	Mean [↑]	σ [↓]	Mean [↑]	σ [↓]
SC++	0.594	0.152	0.746	0.118	0.812	0.134	0.579	0.138	0.730	0.110	0.675	0.170
SOLID	0.358	0.177	0.511	0.177	0.514	0.156	0.373	0.131	0.557	0.134	0.542	0.128
RING++	0.368	0.115	0.544	0.112	0.546	0.156	0.414	0.073	0.584	0.071	0.458	0.140
BTC	0.302	0.038	0.500	0.039	0.337	0.077	0.086	0.045	0.243	0.125	0.131	0.098
MapClosure	0.172	0.092	0.286	0.125	0.163	0.092	0.136	0.044	0.237	0.068	0.127	0.045
Tree-Loc	**0.920**	**0.036**	**0.984**	**0.006**	**0.993**	**0.005**	**0.771**	**0.034**	**0.978**	**0.002**	**0.989**	**0.001**

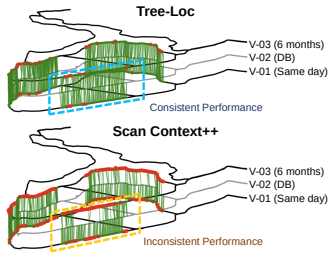


Fig. 6: (Left) Mean and standard deviation (σ) of metrics across 12 pairs. Tree-Loc achieves the highest means with the lowest σ. (Right) While SC++ succeeds for same-day pairs (V-01 and V-02; orange) but fails with temporal discrepancies (V-02 and V-03), Tree-Loc reliably detects true positives across sessions (V-02 and V-03; blue), highlighting its robustness to environmental and temporal variation.

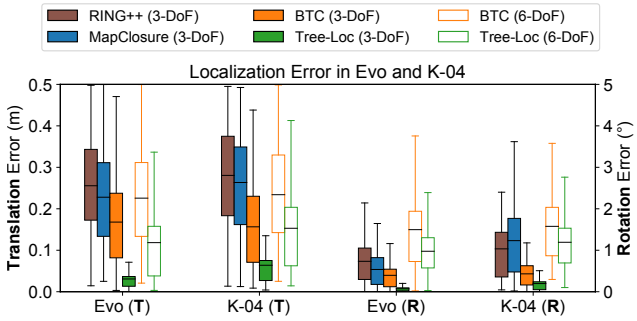


Fig. 7: Localization errors for 3-DoF (Filled) and 6-DoF (Unfilled). Tree-Loc shows lower errors and more stable results in both cases.

narrower interquartile range as shown in Fig. 7. An ablation study showed that the first step in geometric verification already outperformed BTC's 3D triangle matching, while the second step with stem centers further improved accuracy. These reflect Tree-Loc's 2D design, which removes height ambiguity, enables stable geometric matching. Moreover, combined with axis alignment and base height, it allows 6-DoF localization, benefiting practical applications in §IV-H.

E. Learning-based Methods in Forest Place Recognition

We also evaluated Tree-Loc against learning-based methods on Evo, Stein am Rhein, and 1008-01 to highlight its robust generality as an algorithmic method. For the baselines, we used checkpoints from ForestLPR, trained on the Wild-Places, with test sequences beyond the training domain. As shown in Table. IV, Tree-Loc outperformed learning-based methods, which struggled with domain shifts in canopy height and data acquisition systems. Regardless of these shifts, Tree-Loc ensures robust place recognition through stable stem extraction across diverse environments.

In Fig. 4(f), ForestLPR demonstrated strong true positive discrimination but often produced predictions that deviated from the ground truth, despite being designed for forests. The Precision–Recall curves in Fig. 5 further illustrate Tree-Loc's ability to maintain high precision at high recall, consistently outperforming learning-based baselines. Despite the small number of trees in 1008-01, Tree-Loc achieved a moderate R@1 and the highest F1 and AUC scores, highlighting its precision under low tree density. These results suggest that algorithmic approaches based on robust, domain-aligned features can offer superior generalization and efficiency compared to learning-based methods in natural environments.

TABLE IV: Performance Comparison with Learning-based Method

Method	Oxford (Evo)			Oxford (Stein)			Botanic (1008-01)		
	R@1	F1	AUC	R@1	F1	AUC	R@1	F1	AUC
TransLoc3D	0.613	0.760	0.814	0.579	0.725	0.760	0.760	0.863	0.877
LoGG3D-Net	0.703	0.826	0.872	0.593	0.744	0.642	**0.864**	0.927	0.955
MinkLoc3Dv2	0.424	0.595	0.475	0.409	0.581	0.497	0.558	0.717	0.749
ForestLPR	0.672	0.804	0.839	0.628	0.771	0.799	0.825	0.904	0.923
Tree-Loc	**0.907**	**0.966**	**0.992**	**0.871**	**0.945**	**0.988**	0.773	**0.945**	**0.980**

F. Ablation Studies

We evaluated the Tree-Loc configuration through ablations on Stein am Rhein and V-04, reporting R@1, F1, R@50, and total computation time in Table. V. The experiments isolated the contributions of TDH, DBH, and the alignment method.

Role of TDH: TDH is essential for reducing the candidate set during fine retrieval while maintaining, and often enhancing, place recognition performance. Without TDH, triangles formed at locations that overlap spatially but are distant from true positives can match in the hash space, resulting in inaccurate retrievals that are farther from the actual location. TDH reduces search time and stabilizes the matching process by establishing a location-centered distribution that prioritizes nearby candidates. As shown in Table. V, configurations using TDH achieved similar or higher R@1 and F1 score with substantially shorter runtime.

Effect of DBH: To verify whether DBH provides meaningful information, we compared TDH with and without DBH. Applying DBH improved all metrics, with particularly notable improvements in R@1 and R@50. DBH enriches the representation by leveraging forest-specific cues and helps resolve ambiguity between similar geometric structures.

Effect of Scene Alignment: Viewpoint variations cause roll and pitch distortions that affect the dual-descriptor used for place recognition. To evaluate the impact of our preprocessing step, we compared our proposed tree-based scene alignment with ground alignment (as used in ForestLPR) and a no-alignment variant that feeds raw centers directly to TDH and the triangle hash. The results in Table. V demonstrate that using tree axis alignment yields a more stable inter-tree geometry, leading to consistent descriptor extraction and matching, and thereby improving all metrics. While ground alignment is beneficial on relatively flat terrain, it is less reliable on irregular terrain. Without alignment, roll and pitch incline distort in-plane distances and reduce triangle-key consistency, resulting in overall performance degradation.

Runtime Analysis: We evaluated real-time feasibility

TABLE V: Ablation Studies on Tree-Loc (Alignment using Tree (**T**) and Ground (**G**); Time in milliseconds)

Configuration			Oxford (Stein)				Wild-Places (V-04)			
# TDH	DBH	Align	R@1	F1	R@50	Time	R@1	F1	R@50	Time
✗	✗	**T**	0.842	0.930	0.904	252.6	0.842	0.926	0.569	508.1
50	✗	**T**	0.819	0.932	0.874	26.7	0.727	0.850	0.454	26.8
50	✓	**T**	0.859	0.938	0.896	26.9	0.837	0.913	0.534	27.1
100	✗	**T**	0.846	0.936	0.903	45.3	0.822	0.904	0.549	46.0
100	✓	✗	0.748	0.878	0.701	45.5	0.825	0.904	0.469	47.4
100	✓	**G**	0.856	0.940	0.888	45.7	0.882	0.937	0.566	45.9
100	✓	**T**	**0.871**	**0.945**	**0.908**	45.3	**0.890**	**0.942**	**0.599**	45.3

TABLE VI: Comparison of Storage Requirements for Localization

Mission	# Scenes	Raw PCD	BTC Desc.	SC++ Desc.	**Tree-Loc**
Mission01	452	1.7 GB	478.1 MB	3.6 MB	**69.9 KB**
Mission02	788	3.0 GB	857.6 MB	6.5 MB	**136.7 KB**
Mission03	222	0.2 GB	241.2 MB	1.8 MB	**61.0 KB**

TABLE VII: Multi-session Alignment Performance with Runtime

Method	Time A [ms]	Time B [ms]	# Matching Pairs	ATE [m]	ARE [°]
BTC (TP)	527.000	0.946	283	2.491	3.484
BTC (SR)			73	6.085	13.378
BTC (200)			200	25.580	59.910
Tree-Loc	**1.377**	**-**	**1110**	**0.248**	**0.492**

Time A: descriptor generation time, Time B: time to load descriptors from storage.

by measuring the end-to-end time required for descriptor generation, candidate search, and pose estimation on an 11th Gen Intel Core i7-11700 @ 2.50 GHz. All TDH-enabled configurations achieve inference in under 50 ms, enabling low-latency operation suitable for practical deployment.

G. Application 1: Efficient Forest Map Management

Tree-Loc previously generated sets of reconstructed trees ($\mathcal{I}_t$) and dual-descriptors per scene using aggregated payloads ($\mathcal{Z}_t$). While this approach enabled localization at runtime, it lacked scalability due to duplicated storage and fragmented map management. To address this, we aggregate all observations into a single global tree database. At each location, descriptors are dynamically computed from nearby trees, removing the need for scene-specific processing.

To evaluate its efficiency, we compared storage requirements across three sequences collected at the same site as `Evo`: two missions in 2023 (`Mission01` and `Mission02`) using Hesai XT32, and one in 2025 (`Mission03`) using Hesai QT64. As shown in Table. VI, the global tree database required only tens of kilobytes while encoding rich spatial information for each tree, including the stem axis, center, DBH, base height, and observation count. This representation was 97.8% smaller than SC++ global descriptors and 99.9% smaller than BTC local descriptors. Unlike prior experiments, Tree-Loc does not store precomputed descriptors but generates them on demand in about 1.4 ms per location, as shown in Table. VII. This is faster than BTC descriptor generation, which is slowed by processing heavy point clouds, and only marginally slower than loading such descriptors.

A key advantage of the global tree database is that its size scales with the number of unique trees rather than trajectory length, making it more suitable for long-term mapping. Moreover, its tree-level representation supports incremental updates: new missions can be aligned and integrated into the existing map. This is further demonstrated through multi-session alignment across temporally distinct sequences.

H. Application 2: Lightweight Multi-session Alignment

We conducted multi-session alignment across the three missions using the global tree database. Ground truth was obtained via ICP applied to manually matched scan pairs with spatial overlap. Since true positives are not known a priori, inter-session constraints were added only when the overlap ratio $\mathcal{O}(Q, C)$ exceeded 0.2. For BTC, we evaluated three variants: BTC (TP), which retains only true positives; BTC (SR), which retains successful localizations; and BTC (200), which retains the top 200 matches based on feature similarity and reflects a realistic usage scenario. All constraints were inserted into a pose graph and optimized using GTSAM [32], allowing loop closures to refine the SLAM trajectories [28]. The initial trajectories, which do not incorporate GPS measurements, are illustrated in Fig. 8(a).

To ensure fairness, descriptors were computed from equivalent local information. Tree-Loc queried nearby trees from the global tree database, while BTC extracted local crops from point clouds of full scans. As summarized in Table. VII and visualized in Fig. 8(c), Tree-Loc produced the lowest absolute translation error (ATE) and absolute rotation error (ARE), and the most valid constraints. The point cloud overlays in Fig. 1(d) and Fig. 2 further demonstrate accurate alignment without relying on ICP refinement, confirming that the overlap ratio effectively filters false positives. In contrast, BTC (TP) preserved enough links for graph optimization but introduced larger translation errors, as seen in Fig. 8(b) and (d). BTC (SR) produced accurate matches but insufficient constraints for multi-session alignment. BTC (200), while denser, suffered from false positives. Despite differences in LiDAR hardware and acquisition years, Tree-Loc achieved consistent multi-session alignment. These results confirm the practicality of maintaining a single global tree database for compact, updatable, and accurate forest map management, as further illustrated in the accompanying video.

V. Conclusion

We present Tree-Loc, a global localization framework for forests using tree stems as geometric primitives. It combines consistent 2D projection, TDH for coarse retrieval, and 2D triangle descriptors with geometric verification to estimate 6-DoF poses. Tree-Loc outperforms SOTA global and local descriptor-based methods across diverse datasets, demonstrating superior localization accuracy. Ablation studies confirm that TDH, DBH, and 2D projection based on tree axes enhance overall performance. Furthermore, a compact global tree database supports lightweight multi-session alignment and scalable map management. Although Tree-Loc's reliance on tree stems may limit performance in open areas, it still outperforms all baselines across forested regions, including those with sparse trees, in most metrics. Future work will explore additional primitives, such as BEV contours, to enhance localization in open areas or environments with lower tree density. Overall, our results suggest that structure-aware, stem-centric geometry provides a robust and scalable foundation for long-term forest global localization.

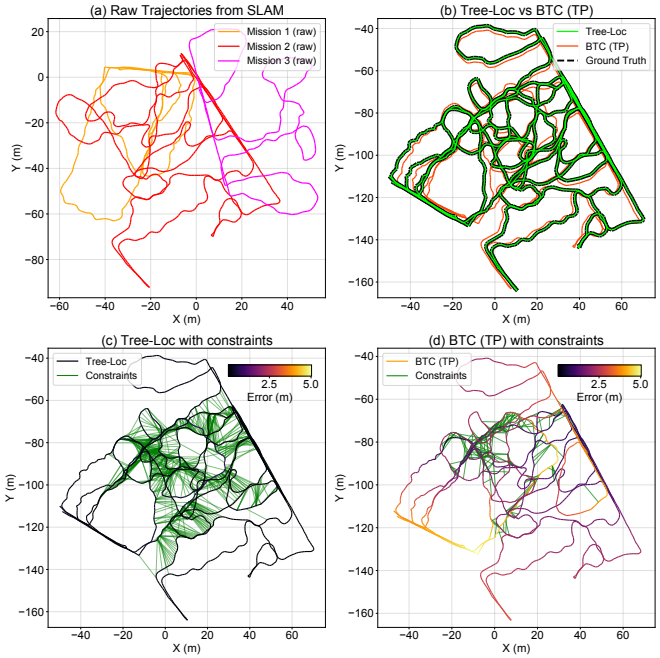


Fig. 8: (a) Initial trajectories before the pose graph optimization. (b) Alignment quality of Tree-Loc and BTC (TP) with ground truth. Tree-Loc shows tighter alignment. (c, d) Optimized trajectories, color-coded based on a translation error. Thanks to the diverse and reliable constraints, Tree-Loc consistently reduces errors across the combined map, while BTC (TP) exhibits larger errors.

REFERENCES

[1] M. Mattamala, N. Chebrolu, J. Frey, L. Freißmuth, H. Oh, B. Casseau, M. Hutter, and M. Fallon, "Building forest inventories with autonomous legged robots–system, lessons, and challenges ahead," *IEEE Trans. of Field Robot.*, 2025.

[2] H. Yin, X. Xu, S. Lu, X. Chen, R. Xiong, S. Shen, C. Stachniss, and Y. Wang, "A survey on global lidar localization: Challenges, advances and open problems," *Intl. J. of Comput. Vision*, vol. 132, no. 8, pp. 3139–3171, 2024.

[3] J. Knights, K. Vidanapathirana, M. Ramezani, S. Sridharan, C. Fookes, and P. Moghadam, "Wild-places: A large-scale dataset for lidar place recognition in unstructured natural environments," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2023, pp. 11 322–11 328.

[4] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *Intl. J. of Robot. Research*, vol. 36, no. 1, pp. 3–15, 2017.

[5] M. Jung, W. Yang, D. Lee, H. Gil, G. Kim, and A. Kim, "Helipr: Heterogeneous lidar dataset for inter-lidar place recognition under spatiotemporal variations," *Intl. J. of Robot. Research*, vol. 43, no. 12, pp. 1867–1883, 2024.

[6] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Trans. Robot.*, vol. 38, no. 3, pp. 1856–1874, 2021.

[7] X. Xu, S. Lu, J. Wu, H. Lu, Q. Zhu, Y. Liao, R. Xiong, and Y. Wang, "Ring++: Roto-translation invariant gram for global localization on a sparse scan map," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4616–4635, 2023.

[8] S. Gupta, T. Guadagnino, B. Mersch, I. Vizzo, and C. Stachniss, "Effectively detecting loop closures using point cloud density maps," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2024, pp. 10 260–10 266.

[9] C. Yuan, J. Lin, Z. Liu, H. Wei, X. Hong, and F. Zhang, "Btc: A binary and triangle combined descriptor for 3-d place recognition," *IEEE Trans. Robot.*, vol. 40, pp. 1580–1599, 2024.

[10] Y. Zhang, Y. Tian, G. Yang, Z. Li, M. Luo, E. Li, and F. Jing, "Intensity triangle descriptor constructed from high-resolution spinning lidar intensity image for loop closure detection," *IEEE Robot. and Automat. Lett.*, 2024.

[11] H. Oh, N. Chebrolu, M. Mattamala, L. Freißmuth, and M. Fallon, "Evaluation and deployment of lidar-based place recognition in dense forests," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2024, pp. 12 824–12 831.

[12] Y. Liu, Y. Fu, M. Qin, Y. Xu, B. Xu, F. Chen, B. Goossens, P. Z. Sun, H. Yu, C. Liu *et al.*, "Botanicgarden: A high-quality dataset for robot navigation in unstructured natural environments," *IEEE Robot. and Automat. Lett.*, vol. 9, no. 3, pp. 2798–2805, 2024.

[13] M. V. Malladi, N. Chebrolu, I. Scacchetti, L. Lobefaro, T. Guadagnino, B. Casseau, H. Oh, L. Freißmuth, M. Karppinen, J. Schweier, S. Leutenegger, J. Behley, C. Stachniss, and M. Fallon, "DigiForests: A Longitudinal LiDAR Dataset for Forestry Robotics," 2025.

[14] L. Freißmuth, M. Mattamala, N. Chebrolu, S. Schaefer, S. Leutenegger, and M. Fallon, "Online tree reconstruction and forest inventory on a mobile robotic system," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2024, pp. 11 765–11 772.

[15] M. Wielgosz, S. Puliti, B. Xiang, K. Schindler, and R. Astrup, "Segmentanytree: A sensor and platform agnostic deep learning model for tree segmentation using laser scanning data," *Remote Sensing of Environment*, vol. 313, p. 114367, 2024.

[16] Y. Shen, T. Tuna, M. Hutter, C. Cadena, and N. Zheng, "Forestlpr: Lidar place recognition in forests attentioning multiple bev density images," in *Proc. IEEE Conf. on Comput. Vision and Pattern Recog.*, 2025, pp. 6659–6669.

[17] L. C. de Lima, E. Griffiths, M. Haghighat, S. Denman, C. Fookes, P. Borges, M. Brünig, and M. Ramezani, "Online 6dof global localisation in forests using semantically-guided re-localisation and cross-view factor-graph optimisation," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2025.

[18] M. Ramezani, E. Griffiths, M. Haghighat, A. Pitt, and P. Moghadam, "Deep robust multi-robot re-localisation in natural environments," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2023, pp. 3322–3328.

[19] G. Tinchev, A. Penate-Sanchez, and M. Fallon, "Learning to see the wood for the trees: Deep laser localization in urban and natural environments on a cpu," *IEEE Robot. and Automat. Lett.*, vol. 4, no. 2, pp. 1327–1334, 2019.

[20] G. Tinchev, S. Nobili, and M. Fallon, "Seeing the wood for the trees: Reliable localization in urban and natural environments," in *Proc. IEEE/RSJ Intl. Conf. on Intell. Robots and Sys.*, 2018, pp. 8239–8246.

[21] J. Komorowski, M. Wysoczanska, and T. Trzcinski, "Egonn: Egocentric neural network for point cloud based 6dof relocalization at the city scale," *IEEE Robot. and Automat. Lett.*, vol. 7, no. 2, pp. 722–729, 2021.

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

[23] H. Kim, J. Choi, T. Sim, G. Kim, and Y. Cho, "Narrowing your fov with solid: Spatially organized and lightweight global descriptor for fov-constrained lidar place recognition," *IEEE Robot. and Automat. Lett.*, 2024.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. IEEE Intl. Conf. on Comput. Vision*, 2011, pp. 2564–2571.

[25] C. Yuan, J. Lin, Z. Zou, X. Hong, and F. Zhang, "Std: Stable triangle descriptor for 3d place recognition," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2023, pp. 1897–1903.

[26] Y. Park, G. Pak, and E. Kim, "Re-trip: Reflectivity instance augmented triangle descriptor for 3d place recognition," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, May. 2025.

[27] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2053–2073, 2022.

[28] D. Wisth, M. Camurri, and M. Fallon, "Vilens: Visual, inertial, lidar, and leg odometry for all-terrain legged robots," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 309–326, 2022.

[29] T.-X. Xu, Y.-C. Guo, Z. Li, G. Yu, Y.-K. Lai, and S.-H. Zhang, "Transloc3d: point cloud based large-scale place recognition using adaptive receptive fields," *Commun. Inf. Syst.*, vol. 23, no. 1, pp. 57–83, 2023.

[30] K. Vidanapathirana, M. Ramezani, P. Moghadam, S. Sridharan, and C. Fookes, "Logg3d-net: Locally guided global descriptor learning for 3d place recognition," in *Proc. IEEE Intl. Conf. on Robot. and Automat.*, 2022, pp. 2215–2221.

[31] J. Komorowski, "Improving point cloud based place recognition with ranking-based loss and large batch training," in *Proc. Intl. Conf. Pattern Recog.*, 2022, pp. 3699–3705.

[32] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," *Georgia Institute of Technology, Tech. Rep*, vol. 2, no. 4, 2012.