

Efficiently Closing Loops in LiDAR-Based SLAM Using Point Cloud Density Maps

Journal Title
XX(X):1-27
©The Author(s) 0000
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/

SAGE

Saurabh Gupta¹ and Tiziano Guadagnino¹ and Benedikt Mersch¹ and Niklas Trekel¹ and Meher V. R. Malladi¹ and Cyrill Stachniss^{1,2}

Abstract

Consistent maps are key for most autonomous mobile robots, and they often use SLAM approaches to build such maps. Loop closures via place recognition help to maintain accurate pose estimates by mitigating global drift, and are thus key for realizing an effective SLAM system. This paper presents a robust loop closure detection pipeline for outdoor SLAM with LiDAR-equipped robots. Our method handles various LiDAR sensors with different scanning patterns, fields of view, and resolutions. It generates local maps from LiDAR scans and aligns them using a ground alignment module to handle both planar and non-planar motion of the LiDAR, ensuring applicability across platforms. The method uses density-preserving bird's-eye-view projections of these local maps and extracts ORB feature descriptors for place recognition. It stores the feature descriptors in a binary search tree for efficient retrieval, and self-similarity pruning addresses perceptual aliasing in repetitive environments. Extensive experiments on public and self-recorded datasets demonstrate accurate loop closure detection, long-term localization, and cross-platform multi-map alignment, agnostic to the LiDAR scanning patterns, fields of view, and motion profiles. We provide the code for our pipeline as open-source software at <https://github.com/PRBonn/MapClosures>.

Keywords

SLAM, Localization, Mapping, LiDAR-based Place Recognition

*

1 Introduction

Mobile robots must navigate their surroundings safely and efficiently. They need to know their location within the environment to successfully navigate to a desired place or explore new areas. Accurate ego-motion estimation helps robots generate accurate maps of the environment, which they can use for navigation. Traditionally, robots often localize themselves using data acquired through exteroceptive sensors like cameras and laser range sensors (LiDAR), proprioceptive sensors like wheel odometers and IMUs, or a combination. Depending on the application and availability, outdoor systems often also exploit GNSS for global positioning.

LiDAR sensors are frequently used in robotics due to their accurate and dense 3D range data. Many previous studies have advanced the field of sequential pose estimation using LiDAR sensors (Vizzo et al. 2023; Guadagnino et al. 2022; Dellenbach et al. 2022; Ferrari et al. 2024; Fontana et al. 2016). Such sequential pose estimation alone, also called sensor odometry, suffers from drifting pose estimates over time due to inherent noise in the robot motion and sensor data, dynamics in the environment, and non-trivial data association problems.

We can compensate for such drift and improve pose estimation by recognizing places previously visited by the robot. This task is referred to as loop closing. It allows the use of geometric information from the LiDAR observations at revisited locations to correct the pose drift, e.g., through pose-graph optimization. Robust loop closure

detection is paramount in simultaneous localization and mapping (SLAM) systems. Robots must recognize that they have returned to a previously visited place to close a loop.

Place recognition is a key sub-task within loop closure detection, which performs data associations between the robot's current view and a database of previously seen places. Generating a database of places is a non-trivial task that requires crafting an as unique as possible description of the environment, invariant to changes in viewpoint. Also, such a database should filter dynamic objects appearing in the sensor data to achieve robust place recognition across long time intervals. Perceptual aliasing is another challenge, as distinct places with similar structures can confuse place recognition algorithms. This could lead to false-positive loop closures that can adversely impact global SLAM estimates (Blanco et al. 2013; Bailey and Durrant-Whyte 2006; Ramos et al. 2007). Furthermore, to perform place recognition for loop closing in a SLAM system, the database should encode the scene's geometry to allow for relative pose estimation between different revisit viewpoints for the pose-graph optimization.

Global feature descriptors computed over the entire point clouds are easy to store in a database and allow for quick

¹University of Bonn, Center for Robotics

²Lamarr Institute for Machine Learning and Artificial Intelligence

Corresponding author:

Saurabh Gupta, University of Bonn, Center for Robotics, Germany.
Email: sgupta2@uni-bonn.de

* Accepted at the International Journal of Robotics Research, April 2026.

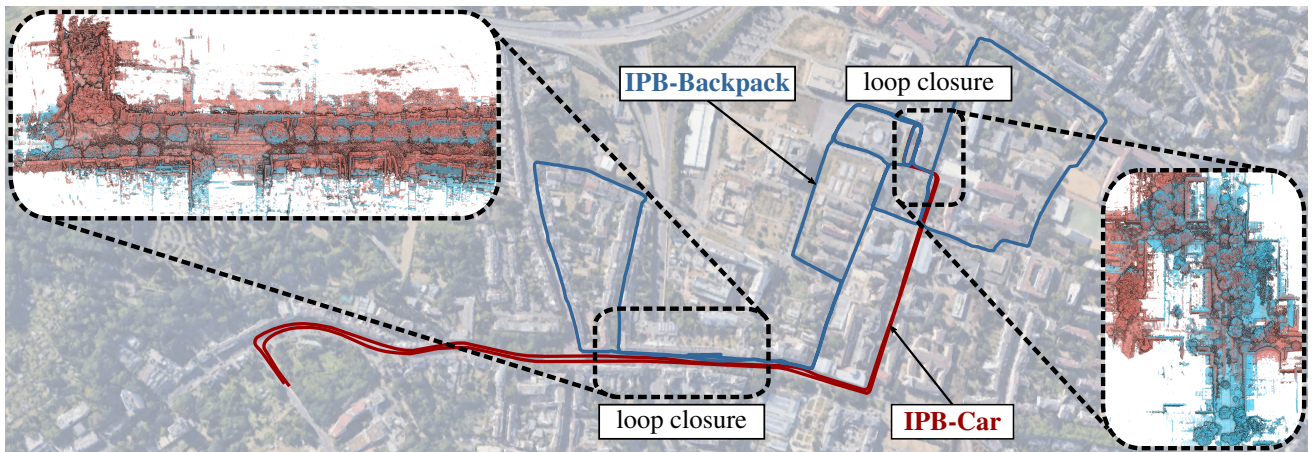


Figure 1. An example of loop closures detected between two sequences recorded with different LiDAR sensor platforms with a revisit interval of two weeks. In blue is the IPB-Backpack sequence, recorded on the campus of the University of Bonn with a Hesai Pandar-128 LiDAR mounted on a backpack. In red is the IPB-Car sequence, recorded in the city of Bonn with an Ouster OS1-128 LiDAR. Both trajectories were individually optimized through a pose-graph with in-session loop closure constraints obtained from our pipeline. We used our loop closure pipeline to detect inter-session loop closures between these two sequences, which have little overlap, and aligned the two trajectories using the multi-session loop closure constraints. The overlapping areas are highlighted by small rectangles, and the corresponding loop closure from these areas is shown in the enlarged rectangles, respectively.

retrieval of revisit candidates (Kim et al. 2021; Uy and Lee 2018; Lu et al. 2022; Xu et al. 2023; Luo et al. 2025). However, such methods often do not provide an initial geometric alignment between the detected loop closures and require an additional point cloud registration step to perform such an alignment. In contrast, feature descriptors computed over local 3D patches within a point cloud can aid the geometric alignment of revisited locations (Blanco et al. 2013; Yang et al. 2017; Steder et al. 2011). However, they require a nuanced approach to store the feature descriptors efficiently in a database.

Due to the algorithmic complexity of processing 3D point clouds to obtain local or global feature descriptors, several methods perform a bird’s-eye-view (BEV) projection (Kim et al. 2021; Li and Li 2021; Luo et al. 2023) or a cylindrical projection (Chen et al. 2020; Steder et al. 2010; Ma et al. 2022) of the point clouds. This results in a 2D representation of the point cloud data, allowing faster feature detection and matching for online loop closure detection in SLAM.

The main contribution of this article is a robust loop closure detection pipeline that works with various LiDAR sensors having different motion profiles, invariant to their scanning pattern, field of view (FoV), and resolution. We generate local maps by aggregating consecutive scans and creating a density-preserving BEV projection of the local maps as an intermediate 2D representation for computing local features describing the structural information in the 3D local maps. Since the ground planes can be consistently identified in outdoor environments during revisits, we use them as a reference plane to ensure consistent BEV projections across revisits. We propose a ground alignment module to identify such a ground plane in each local map and transform the local map such that this ground plane coincides with the local xy-plane of the reference frame. This simplifies the BEV projection and restricts the loop closure alignment to two dimensions on the common ground plane and makes our approach applicable to various motion profiles of LiDAR sensors. We store binary ORB feature descriptors from the BEV projection into a Hamming distance

embedding binary search tree (HBST). We design our approach to be robust against the effects of scene similarity, also known as perceptual aliasing, by performing a self-similarity pruning of the feature descriptors before inserting them in the database. Using a Hamming distance metric, we obtain loop closure candidates by matching feature descriptors from query local maps against this database. Our subsequent random sample consensus (RANSAC) based geometric validation step provides loop closures along with a 2D alignment between BEV projections of the local maps. When combined with the ground alignment estimate of each local map, we obtain a complete 3D estimate of the global alignment between the local maps. We provide an extensive experimental evaluation on multiple datasets, with sequences recorded with a wide range of LiDAR sensors mounted on different mobile platforms, testing the accuracy and robustness of our pipeline in challenging scenarios.

In sum, we make six key claims, which we support with the paper and our experimental evaluation. Our approach

1. detects loop closures between local maps generated from various LiDAR sensors with different scanning patterns, FoVs, and resolutions;
2. performs multi-session loop closure detection and alignment with long-term revisits;
3. works with handheld platforms having non-planar motion in the LiDAR sensor frame;
4. is robust against perceptual aliasing in environments with repetitive structures;
5. provides a complete 3D rigid-body transform to align the detected loop closures;
6. detects loop closures between sequences having minimal overlap, recorded with different LiDAR sensor platforms, enabling cross-platform multi-map alignment.

This article extends our previous conference paper (Gupta et al. 2024), which proposed loop closure detection using local maps by detecting local feature descriptors from their BEV projections and generating a binary search tree

database. It extends the earlier work in the following critical ways: (1) relaxation of the planar-motion assumption through a new ground alignment module; (2) a complete 3D alignment estimate of the detected loop closures; (3) a pruning strategy on the ORB feature descriptors computed from the BEV density images to mitigate the issues with perceptual aliasing; (4) an extensive experimental evaluation on multiple datasets, with sequences recorded with a wide range of LiDAR sensors mounted on different mobile platforms, testing the accuracy and robustness of our pipeline in challenging scenarios.

The open-source implementation of our previous approach and the proposed approach are available at <https://github.com/PRBonn/MapClosures>.

2 Related Work

Zhang et al. (2024) provide a thorough literature review on place recognition for loop closures in 3D LiDAR SLAM and Yin et al. (2024) on LiDAR-based global localization. This section briefly overviews key approaches for LiDAR-based loop closure detection in SLAM.

The most straightforward approach to loop closure detection in SLAM is using the individual scans and their poses obtained from odometry. The similarity of poses between non-consecutive scans within a search radius is used to detect loop closures (Rottmann et al. 2019; Shan et al. 2020). S4-SLAM (Zhou et al. 2021) and the approach by Mendes et al. (2016) use the odometry information to compute the overlap between point clouds to check if they are recorded from the same location. Chen et al. (2020) use convolutional neural networks to calculate the overlap between the range image representations of point clouds to detect loops. These methods primarily perform a place retrieval task, often requiring a subsequent point cloud registration step to validate the loop closures. Furthermore, they can be sensitive to the magnitude of drift present in the odometry pose estimates, requiring a search radius proportional to the drift.

2.1 Methods Using 3D Features From Point Clouds

Place recognition has been widely studied in the context of camera images (Cummins and Newman 2008; Galvez-Lopez and Tardos 2012; Vysotska and Stachniss 2019, 2016; Mur-Artal et al. 2015). Consequently, much research on LiDAR place recognition for loop closing has been motivated by approaches in the camera vision domain. In particular, many approaches have extended the idea of local feature descriptors to 3D point clouds (Rusu et al. 2009; Salti et al. 2014; Johnson and Hebert 1999), discretizing the neighborhood around 3D features into a geometrical grid. They compute a descriptor from the points in this neighborhood based on their height (Bosse and Zlot 2013), density (Tombari et al. 2010; Frome et al. 2004), or distance and angle (Rusu et al. 2008). These local feature descriptor-based methods estimate a complete 3D relative pose between the loop closures. However, they have a high computational cost to identify such feature descriptors from 3D point clouds. They are sensitive to the radially increasing sparsity of point clouds obtained from a standard spinning LiDAR.

Several methods (Magnusson et al. 2009; Kim et al. 2024; Uy and Lee 2018; Röhling et al. 2015) take an orthogonal approach by computing one global descriptor per LiDAR scan. Magnusson et al. (2009) superimpose a voxel grid on the point clouds and approximate a normal distribution over points in each grid cell. They compute a global descriptor for the point clouds by computing a histogram over these normal distributions. More recently, Kim et al. (2024) proposed a lightweight global descriptor with SOLiD by representing point clouds in polar coordinates and discretizing them into range-elevation and azimuth-elevation directions, respectively, with each discrete bin storing the number of points. Such global descriptors are easy to store in a simple database, resulting in a faster matching process to retrieve loop closure candidates. However, global descriptor-based methods typically require revisits within small error margins around the reference pose to limit the variation in the global descriptor. They often do not generalize when detecting loop closures across LiDARs with different scanning patterns and resolutions.

2.2 Two-Dimensional Projection-Based Methods

Many recent works focus on speeding up the recognition by describing 3D point clouds by a 2D projection (Kim and Kim 2018; Li and Li 2021; Luo et al. 2023; Yuan et al. 2024; Xu et al. 2023). Even though such a projection results in loss of geometric information, these methods perform comparably and sometimes even better than their full 3D counterparts, especially in outdoor scenes. Among the two-dimensional projection-based methods, the cylindrical projection into range images (Steder et al. 2010, 2011; Shan et al. 2021; Chen et al. 2020) and the orthogonal projection into BEV images (Luo et al. 2021; Kim et al. 2021; Lu et al. 2022; Yuan et al. 2023; Wang et al. 2020) are widely used 2D projections.

Range image projections are equivariant to azimuthal rotations due to the underlying cylindrical projection. They can recover the relative yaw between loop closures but suffer from scale distortions due to lateral shifts in the LiDAR viewpoint. On the other hand, BEV projections preserve 2D geometry along the local ground plane, which is vital for autonomous ground robots. Among the BEV projection methods, the elevation map is widely used (Kim and Kim 2018; Kim et al. 2021; Yuan et al. 2023; Luo et al. 2023). An elevation map preserves the maximum elevation within each discrete pillar in the BEV representation.

Scan Context by Kim and Kim (2018) is a popular LiDAR-based loop closure approach. It computes a global descriptor for each LiDAR scan using the elevation map in polar coordinates. The polar coordinates make the descriptor equivariant to in-plane rotations, thereby allowing the loop closure alignment module to estimate the relative yaw between LiDAR scans from the same location. However, the polar coordinate representation can be sensitive to in-plane translations. Scan Context++ (Kim et al. 2021) improves upon this drawback of Scan Context and computes elevation maps in Cartesian coordinates to augment the polar elevation maps. However, preserving the maximum elevation of the points after the BEV projection makes the two approaches sensitive to the vertical resolution and FoV of the LiDAR.

In contrast, BVMATCH (Luo et al. 2021) proposes a density-preserving BEV projection of point clouds and a Log Gabor filtering step. It requires training a bag-of-words model to retrieve loop closure candidates, making it less suitable for real-time applications such as SLAM. Our approach also uses a density-preserving BEV projection. It generates a database of local feature descriptors from the BEV image online using the HBST data structure (Schlegel and Grisetti 2018) for efficient operation. The use of local feature descriptors for loop closure detection has also been previously explored in the 2D LiDAR domain with traditional occupancy grid maps (Blanco et al. 2013).

Yuan et al. (2023) propose utilizing a set of geometric primitives, i.e., triangles with unique side lengths, to describe a point cloud. They obtain the feature vertices of such triangles by a local BEV projection of points within voxels that lie at the boundaries of large planar regions in the environment. BTC (Yuan et al. 2024) improves upon this work by proposing a binary descriptor for a detailed representation of local geometry. They combine the triangle descriptors and the newly proposed binary descriptors to perform loop closure retrieval. The works by Yuan et al. (2023, 2024) use a local map representation of the environment by accumulating a fixed number of consecutive scans. This helps them tackle the sparsity issue present with rotating 3D LiDAR. The local maps also make their approach generalizable to LiDAR sensors with non-repetitive scanning patterns and small FoVs. We also use a local map representation to detect loop closures. However, unlike STD and BTC, we accumulate scans until a minimum displacement of the platform to ensure that the local maps capture a sufficient portion of the scene.

2.3 Learning-Based Approaches

The recent popularity of deep neural networks, attributed to the widespread availability of specialized hardware and software to train such networks, has generated much interest from a LiDAR-based loop closing perspective (Dubé et al. 2018; Chen et al. 2020; Ma et al. 2024; Komorowski 2021; Ma et al. 2022). PointNet (Qi et al. 2017b) and PointNet++ (Qi et al. 2017a) use multi-layer perceptron networks to detect local feature descriptors from point clouds. PointNetVLAD (Uy and Lee 2018) proposes an end-to-end trained neural network that combines the local feature descriptors obtained from PointNet into a global descriptor using NetVLAD (Arandjelovic et al. 2016). SegMap (Dubé et al. 2018) segments point clouds using a region-growing algorithm and computes data-driven descriptors of these segments to detect loop closures.

BEV projections are also popular with learning-based approaches (Kim et al. 2019; Luo et al. 2023, 2025; Xu et al. 2021b). Kim et al. (2019) repurpose the Scan Context (Kim and Kim 2018) image as a three-channel image using a jet colormap over a range of structural heights. They train a convolutional neural network using these Scan Context images and perform localization as a classification task. Xu et al. (2021b) propose an encoder-decoder network to extract descriptors from differentiable Scan Context images.

Vidanapathirana et al. (2022) use a global descriptor obtained from a sparse convolutional neural network to perform loop closures. They propose a local-consistency loss

to train their approach to obtain consistent local features across revisits, which they later combine into a global descriptor using a pooling and normalization strategy.

BEVPlace (Luo et al. 2023) applies group convolutions corresponding to the $\mathbb{SO}(2)$ rotational group to obtain rotation equivariant local features on BEV density images and computes a global descriptor for place retrieval using NetVLAD. BEVPlace++ (Luo et al. 2025) improves upon BEVPlace by proposing a novel rotation equivariant module, which passes rotation-warped BEV density images through a convolutional neural network to extract the local features.

Recently, Ramezani et al. (2024) proposed a place recognition method based on an attentional graph neural network representation. Their approach leverages the topological relationships between consecutive LiDAR scans within a pose-graph SLAM system by generating sub-graphs according to a travel distance criterion. They use a multi-layer perceptron to encode the odometry poses and global descriptor associated with every scan within each of the sub-graphs. For a candidate pair of sub-graphs, they construct a fully connected multiplex graph from the encoded nodes and process it through an attentional graph neural network, P-GAT, to perform place recognition.

Even though learning-based approaches achieve impressive performance regarding loop closure precision and recall metrics, they typically require a computationally expensive offline pre-training step on a GPU, affecting their application to real-time loop closure detection in SLAM.

Our method also exploits the spatio-temporal information across consecutive LiDAR scans by constructing local maps as the primary representation for loop closure detection, rather than relying on individual scans. Following our earlier work (Gupta et al. 2024), we generate local maps based on a travel displacement criteria, in contrast to Yuan et al. (2023, 2024), who accumulate a fixed number of scans. This strategy helps us overcome the sparsity of rotating LiDAR and make our method agnostic to the LiDAR sensor’s scanning pattern, FoV, and resolution. We use a density-preserving BEV projection of the local maps on the local ground plane to reduce dimensionality for computational efficiency. Subsequently, we compute ORB feature descriptors (Rublee et al. 2011) from the BEV density images and store them in an HBST database (Schlegel and Grisetti 2018) for place recognition. We mitigate the negative impact of perceptual aliasing on our feature descriptor-matching algorithm by performing a self-similarity pruning of the ORB feature descriptors (Bosse et al. 2004). Our loop closure pipeline provides a complete 3D global alignment between the local maps.

3 Our Methodology

We propose an approach to detect loop closures between local maps using their BEV representation that works with both vehicle-mounted and handheld LiDAR sensor platforms. It also provides a complete 3D initial guess of the relative pose between local maps involved in the loop closures. We present an overview of our pipeline in Fig. 2.

We explain in Sec. 3.1 the procedure to generate local maps from LiDAR point clouds based on a travel displacement criterion. Then, in Sec. 3.2, we present an

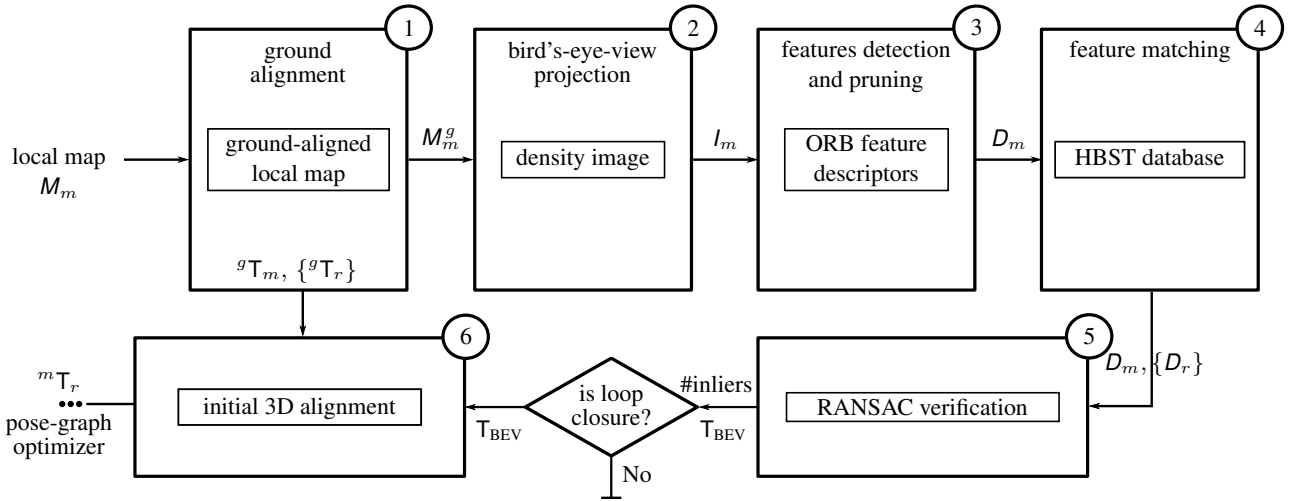


Figure 2. Overview of our pipeline for loop closure detection and alignment. Given a local map M_m generated using a travel displacement criterion: (1) We estimate a ground alignment transform gT_m that aligns the ground plane with the xy -plane of the reference frame. (2) We project the ground-aligned local map M_m^g onto the local xy -plane to obtain a 2D density image I_m . (3) From this density image, we extract ORB feature descriptors D_m and apply self-similarity pruning to remove redundant features. (4) We incrementally insert the pruned feature descriptors into a binary search tree database, which returns feature matches between the query local map M_m^g with feature descriptors D_m and reference local maps $\{M_r^g\}$ with feature descriptors $\{D_r\}$. These matches form the initial loop closure candidates. (5) We verify candidates geometrically using RANSAC, obtaining the number of inliers and the corresponding 2D transform T_{BEV} . We classify candidates as a loop closures if the number of inliers exceeds a threshold. (6) Finally, our pipeline outputs a relative 3D transform mT_r that provides an initial alignment of the two local maps, which a subsequent pose-graph optimizer can refine.

approach to detect and align the ground plane in the local maps to the xy -plane of the local reference frame. This allows us to compute a BEV representation even in scenarios with a non-planar motion of the LiDAR sensor, as explained in Sec. 3.3. We compute binary descriptors for point features on these BEV images, as presented in Sec. 3.4. To enable place recognition using these binary feature descriptors, we store them in an HBST database (Schlegel and Grisetti 2018) for subsequent matching with query local maps as presented in Sec. 3.5. We obtain a set of candidate loop closures for new feature descriptors from a query local map by matching these feature descriptors to the database. We perform loop closure validation and geometric alignment in 2D between the matched feature descriptors using a RANSAC (Fischler and Bolles 1981) scheme described in Sec. 3.6. This 2D alignment, together with the ground alignment for each local map, provides the relative 3D transform aligning each pair of loop-closed local maps.

3.1 Creation of Local Maps

Our approach uses local point cloud maps of the environment to perform loop closures. By aggregating downsampled LiDAR scans over time, we exploit the local consistency of sequential odometry estimates to generate these local maps. We use KISS-ICP (Vizzo et al. 2023) to obtain LiDAR odometry for our pipeline.

Given an online stream of sequential point clouds $\{P_1, \dots, P_n\}$ in the LiDAR frame and their corresponding 3D pose estimates $\{{}^wT_1, \dots, {}^wT_n\}$ in the odometry frame w , we first transform the point clouds into the odometry frame, resulting in point clouds $\{P_1^w, \dots, P_n^w\}$.

Starting from a point cloud with index i , we consider k subsequent scans until the displacement $\|{}^w\mathbf{t}_{i+k-1} - {}^w\mathbf{t}_i\|_2$ exceeds a threshold τ_c , where ${}^w\mathbf{t}_i \in \mathbb{R}^3$ is the translational

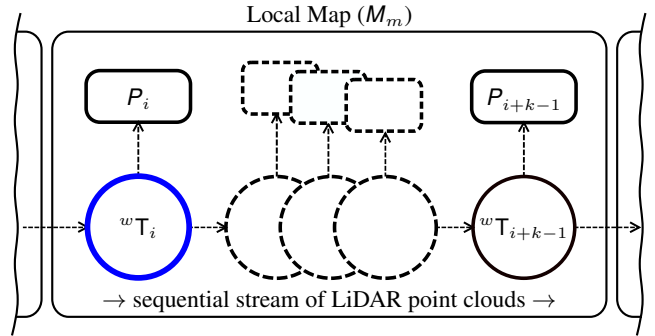


Figure 3. A block diagram showcasing the composition of a local map M_m , generated using point clouds $\{P_i, \dots, P_{i+k-1}\}$ registered with corresponding odometry pose estimates $\{{}^wT_i, \dots, {}^wT_{i+k-1}\}$. The local map is referenced to the local coordinate frame of the first scan in this local map, as highlighted in blue.

component of the pose estimate ${}^wT_i \in \mathbb{SE}(3)$. Once we reach the displacement threshold τ_c , we aggregate this subset of scans $\{P_i^w, \dots, P_{i+k-1}^w\}$ into a local map M_m^w where m is the local map index. Specifically, we use a voxel grid-based downsampling strategy to generate the local map, using a resolution of ν_{map} meters per voxel. Such a downsampling strategy also ensures a uniform spatial distribution of scanned points by retaining at most 20 points per voxel. We repeat this process of local map generation sequentially, thereby generating mutually exclusive sets of consecutive point clouds, as shown in Fig. 3.

We transform each local map M_m^w into the local reference frame wT_i of the first scan in the sub-sequence as follows:

$$M_m = {}^wT_i^{-1} M_m^w. \quad (1)$$

Aggregating consecutive LiDAR scans into local maps provides richer structural information than individual scans,

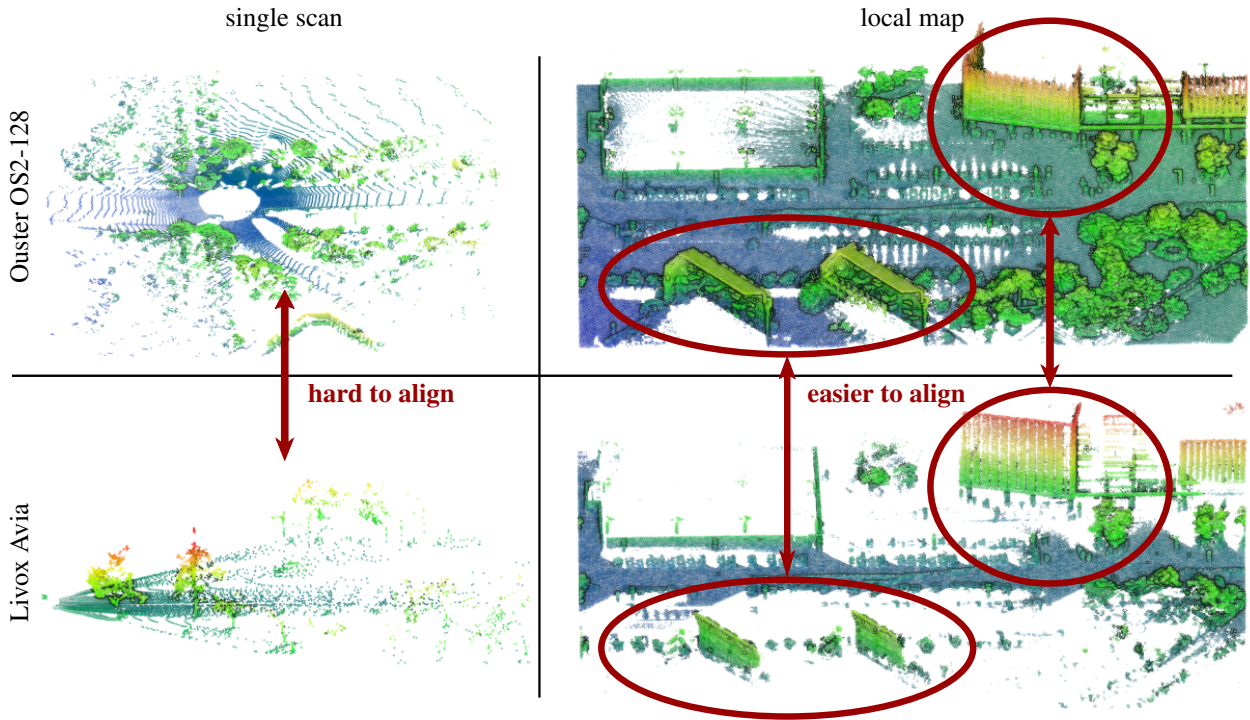


Figure 4. A comparison of data from LiDARs with different scanning patterns and FoVs. The first row shows a single scan from an Ouster OS2-128 LiDAR with a $360^\circ \times 22.5^\circ$ FoV and its corresponding local map. The second row shows a Livox Avia LiDAR with a $70^\circ \times 77^\circ$ FoV and its corresponding local map. The similar structural features across local maps are highlighted with ellipses.

as illustrated in Fig. 4. This larger spatial context is particularly beneficial in multi-session scenarios, where local environmental changes may otherwise hinder place recognition. Furthermore, local maps generated at the same location exhibit higher similarity across different LiDAR resolutions and scanning patterns, whereas inter-LiDAR place recognition with individual scans remains a challenging data association problem, even for humans, as illustrated in Fig. 4. As a result, the use of local maps improves robustness for multi-session and cross-platform loop closure detection.

3.2 Ground Plane Detection and Alignment

After generating the point cloud local map M_m , we perform a BEV projection. To ensure consistency across multiple revisits, we must project the point cloud *onto a consistent physical plane* in the environment. The ground plane provides a natural choice for this plane and enables the computation of a stable basis for the BEV projection. In our earlier work (Gupta et al. 2024), we assumed planar sensor motion so that the ground plane remained parallel to the xy-plane of the local map’s reference frame. However, when the LiDAR follows a non-planar trajectory, the local ground plane tilts relative to the xy-plane, as shown in Fig. 5. To address this, we explicitly identify the ground plane and align it with the xy-plane of the reference frame.

The first step identifies candidate ground points in the local map. Semantic segmentation networks can label ground in point clouds (Xu et al. 2021a; Paigwar et al. 2020), but they require offline GPU training and add significant computational costs. Real-time CPU-based methods such as Patchwork (Lim et al. 2021) and Patchwork++ (Lee et al. 2022) operate on individual scans, so they require per-scan

segmentation and careful parameter tuning. In contrast, our BEV projection requires only one approximate estimate of the local ground plane to align the entire local map at once, not a precise segmentation for every scan.

We adopt a simple sampling strategy: in a 3D LiDAR local map, the lowest-elevation points usually correspond to the ground (Hu et al. 2014). This assumption generally holds for typical outdoor sensing platforms with known extrinsics, where the ground plane lies within the LiDAR’s field of view. It also remains valid when the local map’s xy-plane is not perfectly aligned with the ground plane, since a LiDAR cannot scan below the ground surface.

We divide the local map M_m into a 2D grid over the xy-plane with 1.0 m cells. In each cell, we retain only the point with the minimum z-coordinate, yielding a sparse set of candidate ground points, as shown in Fig. 5. This set may still contain non-ground points where the ground is occluded or where the xy-plane is misaligned with the ground plane. To filter these, we run principal component analysis (PCA) on the normal vectors associated with the sampled points and retain only samples whose normals have a cosine similarity larger than 0.95 with the first principal component. This step removes most non-ground points and yields the ground sample set G_m . The first principal component also provides a strong initial estimate of the ground orientation.

We convert this PCA estimate into an initial transform $T_{\text{init}} \in \mathbb{SE}(3)$ for ground alignment. Specifically, we compute an axis-angle rotation that aligns the first principal component with the z-axis of the local map frame:

$$\mathbf{a} = \mathbf{v}_{\text{pc}} \times [0 \ 0 \ 1]^T, \quad (2)$$

$$\theta = \arccos(\mathbf{v}_{\text{pc}} \cdot [0 \ 0 \ 1]^T), \quad (3)$$

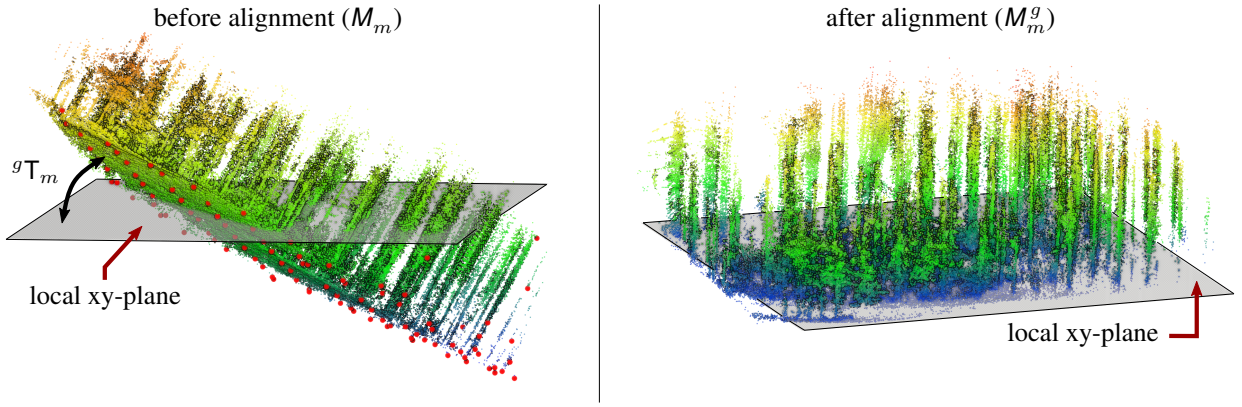


Figure 5. Effect of ground alignment on a local map generated from a handheld LiDAR sensor, recorded in a forest, with non-planar motion. The colors of the points represent their z-coordinates in the local reference frame. *Left:* Our ground alignment approach samples ground points G_m (highlighted in red) from the local map M_m and computes an alignment gT_m to the xy-plane. *Right:* The ground-aligned local map M_m^g .

where, v_{pc} denotes the first principal component, \mathbf{a} the axis of rotation, and θ the rotation angle. We convert this axis-angle pair into a rotation matrix $R_{init} \in \mathbb{SO}(3)$ using Rodrigues’ formula. We also compute a mean z-shift:

$$z_{init} = \mathbf{r}_2 \cdot \bar{\mathbf{g}}_m, \quad (4)$$

where, \mathbf{r}_2 is the last row of R_{init} and $\bar{\mathbf{g}}_m$ is the mean of G_m . We combine R_{init} and z_{init} into the initial transform T_{init} .

To make the ground alignment robust to outliers, we refine T_{init} with a least-squares optimization:

$${}^gT_m = \underset{T_{init}}{\operatorname{argmin}} \sum_{\forall \mathbf{g}_k \in G_m} w_k (\hat{\mathbf{n}}_{xy} \cdot (T_{init} \mathbf{g}_k))^2, \quad (5)$$

where, $T_{init} \mathbf{g}_k = [x'_k \ y'_k \ z'_k]^T$ denotes the $\mathbb{SE}(3)$ group action on \mathbb{R}^3 , $\hat{\mathbf{n}}_{xy} = [0 \ 0 \ 1]^T$ is the normal vector of the xy-plane, and w_k is a Gaussian weight that reduces the influence of spurious outliers. We iterate this optimization, stopping after ten iterations or once the pose correction between iterations falls below a threshold.

We compute the Jacobian J of the residual term $\hat{\mathbf{n}}_{xy} \cdot (T_{init} \mathbf{g}_k)$ in Eq. (5) with respect to a perturbation $\Delta \mathbf{x} = [\Delta t^T \ \Delta \omega^T]^T$ around T_{init} on the $\mathbb{SE}(3)$ manifold:

$$J = [0 \ 0 \ 1 \ x'_k \ -y'_k \ 0]. \quad (6)$$

The Jacobian indicates that the optimization only updates the z-axis translation and the roll (x-axis) and pitch (y-axis), which directly correct the ground alignment.

Finally, we apply the ground-aligning transform gT_m to the local map M_m to obtain the aligned local map M_m^g whose ground plane lies on the xy-plane. The alignment step increases robustness to non-planar LiDAR motion profiles, particularly for handheld platforms. It establishes the ground plane as a robust reference plane for BEV projections during long-term revisits, since the ground surface typically remains consistent over time. We show an example of this alignment in Fig. 5 and provide a detailed analysis in Sec. 5.3.

3.3 Density-Preserving Bird’s-Eye-View Projection

After establishing the primary representation of the environment in Sec. 3.1 and Sec. 3.2, the next step is to

extract informative yet distinct local feature descriptors to perform place recognition for loop closures. Rather than computing features directly on the 3D point clouds of local maps, which is computationally expensive given the spatial extent of such point clouds, we use a 2D BEV projection of the local maps as an intermediate representation for loop closure detection.

We perform the BEV projection by simply dropping the z-coordinate of all the individual points in the ground-aligned local maps M_m^g . Furthermore, we discretize the \mathbb{R}^2 projection space with a resolution of ν_b meters. This results in a 2D Cartesian grid $N_m(u, v)$ of size $W_m \times H_m$ where,

$$W_m = \left\lfloor \frac{x_m^u}{\nu_b} \right\rfloor - \left\lfloor \frac{x_m^l}{\nu_b} \right\rfloor + 1, \quad (7)$$

$$H_m = \left\lfloor \frac{y_m^u}{\nu_b} \right\rfloor - \left\lfloor \frac{y_m^l}{\nu_b} \right\rfloor + 1, \quad (8)$$

$$\left[\begin{array}{c} x_m^u \\ y_m^u \end{array} \right] = \max_{x,y} M_m^g; \quad \left[\begin{array}{c} x_m^l \\ y_m^l \end{array} \right] = \min_{x,y} M_m^g. \quad (9)$$

However, the dimensionality reduction comes at the cost of losing the complete 3D information about the scene. Many traditional BEV projection approaches store the maximum elevation of the points in each cell (Kim et al. 2021; Kim and Kim 2018; Li and Li 2021; Luo et al. 2022). The maximum elevation, however, is sensitive to the distance between the scanner and the surface being scanned, as well as the LiDAR sensor’s FoV. In our pipeline, we instead store the *point density* in each discrete 2D cell after the projection, which is less sensitive to viewpoint changes (Luo et al. 2021).

Therefore, each cell in this grid $N_m(u, v) \in \mathbb{N}_0$ stores the count of projected points within that cell. The BEV image $I_m(u, v)$ of the local map M_m^g is then defined as the relative point density in each cell of the grid as follows:

$$I_m(u, v) = \frac{N_m(u, v) - N_{min}}{N_{max} - N_{min}} \in \mathbb{R}^{W_m \times H_m}, \quad (10)$$

$$N_{max} = \max N_m(u, v); \quad N_{min} = \min N_m(u, v). \quad (11)$$

To mitigate the undesired influence of dynamic objects that accumulate during local map generation, we explicitly set all image pixels $N_m(u, v)$ with a relative density lower than 5% to zero. This choice reflects the fact that most

dynamic objects commonly found in urban environments, such as vehicles, pedestrians, and cyclists, have small vertical footprints and therefore produce low-density values.

3.4 Feature Detection and Pruning Strategy

The BEV projection reduces the dimensionality of the local map, making feature detection for place recognition computationally efficient. The image-like BEV density representation enables us to apply well-established computer vision techniques to extract distinctive feature descriptors. Since these BEV density images preserve the geometrical, floorplan-like structure of the environment, we use ORB (Rublee et al. 2011) feature descriptors to capture relevant features. Unlike camera images, BEV images generated via orthographic projection have no scale ambiguity. We take advantage of this property by computing the ORB feature descriptors D_m without applying scale-invariance adjustments, further improving computational efficiency.

Fig. 6 shows an example of ORB features extracted from a BEV density image. We observe that most ORB features concentrate around high-density regions with strong gradients. These strong responses typically arise from static structures with large vertical footprints, such as building facades, trees, and light poles. In contrast, low-density regions, often corresponding to small bushes, vegetation, or dynamic objects that occasionally pass through the low-density filter, contribute few ORB features to the database. This leads to a robust behavior of our algorithm as it is not adversely affected by dynamic objects in the environment.

ORB feature descriptors remain salient within a local neighborhood of the density image, but not necessarily across the entire image. As a result, environments with repetitive structures can generate self-similar feature descriptors that can confuse the feature-matching stage and cause false positives.

To improve robustness against perceptual aliasing, we prune the ORB feature descriptors within each density image using a self-similarity check inspired by Bosse et al. (2004). Specifically, we compute the nearest-neighbor match for each ORB feature descriptor based on the Hamming distance and discard any feature whose closest match lies below a threshold of $\tau_{pr} = 35$ bits. This procedure eliminates self-similar feature descriptors within the same density image. This approach resembles Lowe’s ratio test (Lowe 2004), but we apply it intra-image rather than inter-image, which avoids the cost of finding second-best matches across a larger feature descriptor database.

This pruning step filters out features from repetitive structures in the environment, such as repeating trusses on a bridge, as shown in Fig. 7. This helps reduce false-positive feature matches across density images resulting from perceptual aliasing, thereby lowering the risk of incorrect loop closure detections. We present a detailed analysis of this pruning strategy and its impact on loop closure detection in Sec. 5.4.

3.5 Feature Database

Once we have the unique features within a BEV density image, we create a database to serve as a reference for place recognition. We leverage the binary domain of the ORB

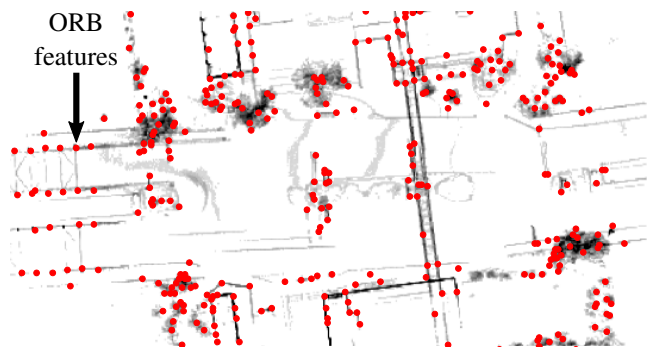


Figure 6. A BEV density image of a local map, where darker pixels indicate higher point density. Red dots mark the ORB features extracted from the image.

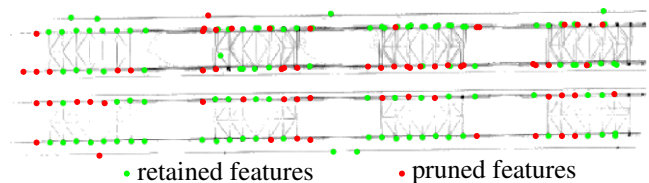


Figure 7. An example of self-similarity feature pruning applied to ORB feature descriptors computed on a BEV density image of a bridge with repetitive mechanical structures. Our algorithm prunes the features shown in red and retains those in green.

descriptors, using the Hamming distance embedding binary search tree (HBST) (Schlegel and Grisetti 2018) to store the set of feature descriptors D_m obtained from each density image I_m , along with their corresponding map index m .

The depth of the HBST is limited by the number of bits in the binary descriptor. As a result, a query descriptor will require at most 256 bitwise comparisons with the tree’s nodes before reaching a leaf node. Each leaf node can hold a maximum of 100 descriptors, ensuring efficient feature matching. These design choices constrain the computational time for feature matching without imposing practical limitations on the usability of our approach. Unlike a clustering-based bag of words or a neural representation-based database, we use the HBST as an incrementally updated database, requiring no offline pre-training.

After obtaining a set of descriptors D_m from the query local map’s BEV density image I_m , we search the HBST database to find the closest match for each descriptor $d_{j,m} \in D_m$. Matches are determined using a Hamming distance threshold of 50 bits between ORB descriptors. Since the binary tree links each stored descriptor to an index m corresponding to its local map, we compile a list of feature matches between the query local map and the reference local maps in the database. These matches are subsequently verified geometrically. Fig. 8 visualizes the corresponding set of feature matches between two density images, as obtained from the HBST database.

3.6 Loop Detection and Map Alignment

The geometric validation step entails performing a 2D alignment of the matched ORB features. This involves computing a 2D rigid-body transformation that optimally aligns the matched features from the binary search

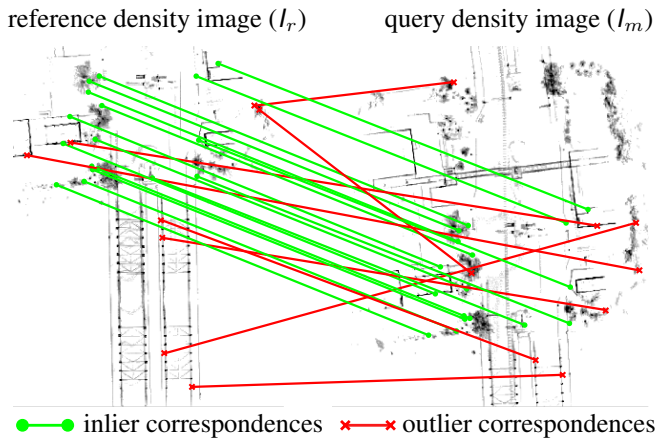


Figure 8. ORB descriptor matches between features from a reference and a query BEV density image obtained from the HBST database. The inlier correspondences obtained from the RANSAC-based geometric verification are shown with green lines, and the outlier correspondences with red lines.

tree using a distance metric. Unlike the general image-alignment problem, this process is constrained to an $\mathbb{SE}(2)$ transformation rather than a homography. To handle outlier associations caused by the limitations of binary tree-based matching, we employ a RANSAC-based alignment strategy.

We design a RANSAC-based approach that randomly selects two feature pairs from the set of matches between a query (I_m) and a reference (I_r) density image. We use the Kabsch-Umeyama algorithm (Kabsch 1976; Umeyama 1991) to compute the relative 2D alignment between the feature pairs. We use this alignment to compute the point-wise Euclidean distance between each pair of feature matches. Feature matches with distances larger than 1.5 m (3 pixels for $\nu_b = 0.5$ m) are considered outliers. The RANSAC process runs for a fixed number of iterations N_{ransac} . We require a minimum number of inliers γ from the RANSAC alignment stage to decide whether two local maps belong to the same location.

Finally, after the N_{ransac} iterations, we compute a Kabsch-Umeyama 2D alignment over the entire set of inlier correspondences. It provides us with a rotation matrix $R \in \mathbb{SO}(2)$ and a translation vector $t \in \mathbb{R}^2$, which we represent as a homogenous transformation $T_{\text{BEV}} \in \mathbb{SE}(3)$, having translation components only along the xy-plane, and a rotation component only about the z-axis. We scale the translation vector by the voxel size (ν_b) since the features are computed from the discretized density images.

Although T_{BEV} only aligns the ground-aligned local maps, we can recover the complete $\mathbb{SE}(3)$ pose to align the original local maps by composing T_{BEV} with the ground-aligning transforms of the individual local maps, as shown in Eq. (12).

$${}^m T_r = {}^g T_m^{-1} T_{\text{BEV}} {}^g T_r. \quad (12)$$

This 3D alignment serves as an initial estimate for a fine-grained registration of local maps during typical pose-graph optimization in a SLAM pipeline. A qualitative example demonstrating the effectiveness of this initial alignment is illustrated in Fig. 9. Notably, one of the local maps (in blue) in this visualization contains a large trace from a dynamic object (highlighted by the green ellipse), showing that our

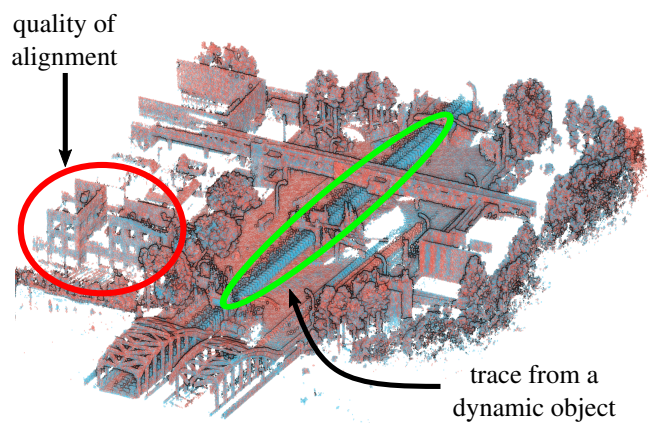


Figure 9. Two local maps (in red and blue) detected as loop closure and aligned using the initial estimate ${}^m T_r$ provided by our pipeline. Dynamic objects in the scene, as seen in the area highlighted by a green ellipse, do not affect the alignment quality provided by our approach. The red ellipse highlights the overlap of windowpanes on a building.

pipeline can also detect and align loop closures even in the presence of such strong dynamics.

4 Experimental Setup for Evaluation

This section presents our experimental setup for evaluating our method and comparing it to existing loop closure detection approaches. We begin by introducing the datasets we use for evaluation and benchmarking. Next, we provide details about the implementation of the baseline methods we include in our comparison. Finally, we describe the quantitative metrics we use to assess the performance of our loop closure detection pipeline and the competing baselines.

4.1 Datasets

We conduct a comprehensive evaluation of our approach on datasets collected using a variety of LiDAR sensors with different resolutions and scanning patterns. These datasets include multiple sequences recorded in diverse urban environments across various mobile platforms.

4.1.1 Public Datasets: We use the MulRan (Kim et al. 2020) and HeLiPR (Jung et al. 2024) public datasets, both recorded in urban environments using LiDAR sensors mounted on car rooftops. These datasets are well-suited for intra-session loop closure evaluation due to frequent in-sequence revisits with varying orientations and lane shifts. In addition, both datasets include at least three sequences per environment, making them ideal for evaluating inter-session loop closure detection.

The MulRan dataset was recorded using a 64-beam rotating LiDAR (Ouster OS1-64) operating at 10 Hz. Its horizontal FoV is limited to 290° due to occlusion from a radar sensor mounted behind it. We use sequences from three environments: KAIST, Riverside, and Sejong.

The HeLiPR dataset features multiple LiDAR sensors with different FoVs, scan patterns, and resolutions mounted on a single vehicle, enabling the evaluation of inter-LiDAR loop closure detection. From this dataset, we use three of the four available LiDARs: (i) a 128-beam spinning LiDAR (Ouster

OS2-128) with $360^\circ \times 22.5^\circ$ FoV, (ii) a hybrid solid-state LiDAR (Livox Avia) with $70^\circ \times 77^\circ$ FoV and an unusual non-repetitive scanning pattern, and (iii) a solid-state LiDAR (Aeva Aeries II) with $120^\circ \times 19.2^\circ$ FoV. We exclude the fourth sensor (Velodyne VLP-16) due to self-occlusion caused by surrounding sensors. The Bridge sequences in this dataset pose an additional challenge due to strong perceptual aliasing from repetitive structures in certain regions.

Moreover, both MulRan and HeLiPR datasets include sequences from the KAIST and Riverside environments, recorded approximately 4 years apart. This allows for a realistic evaluation of long-term inter-session loop closure detection, including cross-dataset comparisons between sequences captured with different LiDAR sensors. The KAIST and Riverside sequences also have some spatial overlap, providing an opportunity to evaluate multi-map alignment in a more challenging setting.

Additionally, we use two sequences from the NCLT dataset (Carlevaris-Bianco et al. 2016), recorded using a Velodyne HDL-32E LiDAR mounted on a Segway robot. The motion of the LiDAR in these sequences is non-planar due to the inverted pendulum-like dynamics of the Segway platform. We select two sequences recorded in the same environment but one year apart to assess the robustness of our loop closure detection approach under viewpoint and temporal variation.

4.1.2 Self-Recorded Datasets: In addition to the public datasets described above, we evaluate our approach on sequences recorded using our own instrumented vehicle and sensor platform. The IPB-Car sequence was recorded using an Ouster OS1-128 LiDAR mounted on the rooftop of an instrumented vehicle. We collected this data by driving through a hilly, semi-urban area with forest-lined roads and performing multiple revisits from varying viewpoints.

The IPB-Backpack sequence was captured using a sensor platform mounted on a backpack and carried through urban streets. This setup uses a Hesai Pandar-128 LiDAR mounted approximately 2m above ground level. Unlike vehicle-mounted systems, the walking motion introduces non-planar sensor trajectories and dynamic changes in the viewpoint. This sequence presents a challenging test case for evaluating our ground alignment strategy under non-planar motion.

The two sequences share a small physical overlap in their environments, which we utilize to evaluate inter-session loop closure detection.

We generate near-ground-truth pose information using the offline LiDAR bundle adjustment method by Wiesmann et al. (2024), with manual inspection of the results to ensure alignment accuracy. This method incorporates RTK-GNSS data for geo-referencing; an information not provided to any loop closure system under evaluation. The process begins with initial pose estimates obtained from pose-graph fusion of LiDAR odometry (Vizzo et al. 2023), aided by either manually added loop closures (for the IPB-Backpack sequence) or RTK-GNSS data (for the IPB-Car sequence) to enforce global consistency. The final bundle adjustment refines these initial poses by aligning all scans with each other in an offline fashion.

4.2 Baseline Methods Used for Comparison

We compare our approach against seven baseline methods, all of which provide publicly available implementations that we use for evaluation. To ensure a fair comparison, we modify their source code only to return multiple loop closure candidates per query scan rather than just the top-ranked one. Unless stated otherwise, we retain the default parameter settings in each baseline’s implementation.

4.2.1 Our Approach: For our approach, we limit the maximum range of each LiDAR scan to 100m. We set the travel displacement threshold τ_c for generating local maps to 100m as well. For KISS-ICP (Vizzo et al. 2023) odometry, we use their latest open-source version (v1.2.3). We use a voxel size of $\nu_{\text{map}} = 0.5$ m for the voxel grid used to generate the local maps, and the resolution of the BEV density images ν_b is set to 0.5m. We use the default parameters for ORB feature descriptors, except that we turn off the scale invariance since the BEV images are orthographic projections. We obtain feature matches from HBST using a Hamming distance threshold of 50 bits and leave all other HBST parameters at their defaults. Finally, we classify two local maps as loop closures if the RANSAC-based alignment yields more than $\gamma = 5$ inliers.

For multi-session experiments, we save the HBST database from the reference session and use it as the database for the query session. We apply the same configuration to evaluate our previous approach, referred to as MapClosure (Gupta et al. 2024).

4.2.2 Scan Context (SC): Scan Context (Kim and Kim 2018) is a widely used, state-of-the-art approach for LiDAR-based loop closure detection. However, it is specifically designed for LiDARs with a large horizontal FoV and a cylindrical scanning pattern. Therefore, we omit evaluation on the Livox Avia and Aeva Aeries II sensors from the HeLiPR dataset.

4.2.3 Stable Triangle Descriptors (STD) and Binary Triangle Combined Descriptors (BTC): STD by Yuan et al. (2023) introduces a triangle descriptor for point cloud local maps, which BTC (Yuan et al. 2024) extends with a binary descriptor for efficient storage and fast retrieval of loop closure candidates. Both methods accumulate 10 consecutive scans into a local map to compute descriptors. We use KISS-ICP to obtain the pose estimates for constructing these local maps. We adopt the default parameter settings from their implementations. For a fair comparison, we also evaluate STD and BTC on the same local maps used in our approach, generated with a 100m travel displacement criterion. We refer to these variants as STD-100 and BTC-100.

4.2.4 Spatially Organized and Lightweight Global Descriptor (SOLiD): SOLiD (Kim et al. 2024) is a recent loop closure detection method for 3D LiDAR sensors that is designed to operate across a wide variety of sensors, irrespective of their scan pattern, FoV, or resolution. The method employs a kd-tree-based search strategy to identify candidate loop closures. However, the authors do not provide an implementation of this search procedure in the publicly available codebase. Consequently, we approximate the candidate set by selecting all scans at least hundred frames prior to the current query scan. Following the procedure

outlined in their manuscript, we compute the evaluation metrics using cosine distances between the descriptors of the query scan and each candidate.

4.2.5 LoGG3D-Net: LoGG3D-Net (Vidanapathirana et al. 2022) is a learning-based method that employs a 3D sparse convolutional network to extract consistent local features across different viewpoints. For our evaluation, we use the checkpoint trained on sequences from the MulRan dataset, as provided by the authors, and we adopt the default parameter settings from their implementation.

4.2.6 BEVPlace++: BEVPlace++ by Luo et al. (2025) is a learning-based method that computes a global descriptor for the BEV projection of the input scan using a rotation-equivariant module. In our evaluation, we use the checkpoint trained on sequence 00 from the KITTI dataset (Geiger et al. 2013), as provided by the authors, and we adopt the default parameter settings from their implementation.

4.3 Reference Loop Closures Between Local Maps

Since our method computes loop closures between local maps, we also define reference loop closures at the map level for evaluation. In contrast to prior works (Jiang and Shen 2023; Kim and Kim 2018; Kim et al. 2021; Yuan et al. 2023; Kim et al. 2024), which typically rely on simple distance-based criteria, such heuristics are insufficient in our setting. The distance-based criterion breaks down in scenarios where other objects occlude the previously seen area, the sensor has a limited FoV, or the revisits are from a significantly different viewpoint. Additionally, there is inherent ambiguity in selecting the appropriate reference frame for measuring distances between local maps.

To overcome these limitations, we define reference loop closures based on the volumetric overlap of local maps, similar to Gupta et al. (2024) and Yuan et al. (2024). We generate reference local maps using the ground-truth pose information provided by the respective datasets. We ensure that each reference local map contains the same set of scans as the corresponding local maps produced by our method. We then identify reference loop closures by selecting all pairs of local maps that exhibit more than 25% voxel-level overlap. To avoid trivial short-range closures, we skip the three consecutive local maps preceding the query local map.

We adopt the same volumetric overlap strategy for cross-sequence evaluation, but reduce the overlap threshold to 10% to accommodate potential misalignments in the global pose information between sequences.

4.4 Scan-Level to Map-Level Conversion of Loop Closures for Baselines

The baseline methods in our evaluation operate at different scales; some detect loop closures at the individual scan level (Scan Context and SOLiD), while others work on local maps with a fixed number of scans (STD and BTC). As a result, we cannot directly evaluate them using the reference loop closures defined between local maps generated based on a travel displacement criterion. Therefore, we convert their outputs into equivalent map-level loop closures.

For scan-level approaches, we treat a loop closure between scans P_i and P_j as a loop closure between local maps M_k and M_l , where scan P_i belongs to map M_k and scan P_j belongs to map M_l .

For methods such as STD and BTC, which use smaller local maps, we first extract all scan pairs between these local maps, S_m and S_n , that form a loop closure. We treat each scan pair as an individual scan-level loop closure and convert it into a corresponding map-level loop closure using the same method described above.

4.5 Evaluation Metrics

4.5.1 Precision, Recall, and F1 Score: We use the reference closures computed in Sec. 4.3 to evaluate our approach and the baseline methods quantitatively. Specifically, we compute precision-recall curves by varying the threshold γ on the number of inlier feature descriptor matches from our pipeline’s RANSAC-based geometric validation stage. For the baseline methods, we generate their respective precision-recall curves by varying the key thresholding parameter described in their publications. In addition to the precision-recall curves, we report the average precision (AP) (area under the precision-recall curve), the maximum recall at 100% precision (R@1), and the maximum F1 score (F1_m). We specifically choose to report R@1 as the maximum recall at 100% precision to emphasize the need to avoid false loop closures in a SLAM pipeline that could lead to catastrophic failures (Lowry et al. 2016).

4.5.2 Absolute Pose Error: We evaluate the effectiveness of our approach in correcting drift within a SLAM pipeline through an offline pose-graph optimization using the g2o optimizer (Kümmerle et al. 2011). We directly incorporate the detected loop closures between local maps as constraints in the pose-graph, using the initial transformation estimates from our pipeline.

We assess performance by computing the root mean square (RMS) absolute pose error (APE) in translation, with respect to ground-truth poses, both before and after the pose-graph optimization. To directly evaluate the accuracy of the detected loop closures and their alignments, we do not apply any robust kernel for outlier rejection during optimization.

We further evaluate the accuracy of the initial alignment estimate (mT_r) between loop-closed local maps using the RMS APE in translation and rotation.

5 Experimental Evaluation

The primary focus of this work is an accurate and effective loop closure detection pipeline that works with various LiDAR sensors, invariant to their scanning pattern, FoV, and resolution. We present our experiments to show the capabilities of our method and support our key claims. Our approach (1) detects loop closures between local maps generated from various LiDAR sensors with different scanning patterns, FoVs, and resolutions; (2) performs multi-session loop closure detection and alignment with long-term revisits; (3) works with handheld platforms having non-planar motion in the LiDAR sensor frame; (4) is robust against perceptual aliasing in environments with repetitive structures; (5) provides a complete 3D rigid-body transform

to align the detected loop closures; (6) detects loop closures between sequences having minimal overlap, recorded with different LiDAR sensor platforms, enabling cross-platform multi-map alignment.

5.1 Intra-Session Loop Closure Detection

In this experiment, we evaluate the performance of our approach on intra-session loop closure detection. We compare against several state-of-the-art baselines by converting their detected closures to local map-level closures, as described in Sec. 4.4. In addition, we include a comparison with our previous conference publication, MapClosure (Gupta et al. 2024), which forms the foundation of this work. We also evaluate the local map-based baselines STD and BTC using the same 100m travel displacement-based local maps as in our approach, referred to as STD-100 and BTC-100. However, we do not evaluate the scan-based baselines using similar local maps since scan-based methods like Scan Context and SOLiD depend on a *single scan center* as the reference frame, which local maps inherently lack. Handling large lateral shifts would require sampling many artificial scan centers, changing the methods’ intended use.

The quantitative results in Tab. 1 and Tab. 2 show the effectiveness of our approach across multiple public and in-house datasets. Our method achieves the highest average precision in fifteen of the twenty-four sequences and ranks second in eight of the remaining nine. In terms of recall, it achieves the best performance at 100% precision in fifteen sequences and the second-best in nine others. This balance between precision and recall is also reflected in the F1 scores, where our method ranks first in eighteen sequences and second in six others.

Our method provides a robust solution for loop closure detection within a SLAM pipeline, where false positives must be minimized (Blanco et al. 2013; Bailey and Durrant-Whyte 2006; Lowry et al. 2016). The precision-recall curves in Fig. 10 show that our approach (red) consistently outperforms baselines across datasets with varied sensor setups. It maintains high precision over a broad range of recall values, eliminating the need for careful tuning of the inlier threshold in RANSAC or additional outlier rejection schemes in the pose-graph optimization.

The evaluations on the HeLiPR dataset demonstrate the versatility of our method across different LiDAR sensors in varied urban environments. Our method consistently ranks among the top two methods across all metrics. In several sequences, our current approach ranks second only to MapClosure. However, the precision-recall curves in Fig. 10 reveal a key advantage: our method maintains a higher precision than MapClosure, ensuring that detected loop closures remain accurate.

The HeLiPR Bridge sequences, which exhibit strong perceptual aliasing, highlight the effectiveness of our pruning strategy. While overall metrics in Tab. 1 may appear comparable between our method and MapClosure, our precision-recall curves in the first row of Fig. 10 terminate earlier at a higher precision value than MapClosure, which also holds in comparison to other baselines. This behavior reflects our deliberate choice to prioritize precision over recall in perceptually ambiguous scenes. By doing so, we improve robustness, reduce sensitivity to the inliers

threshold (γ), and lower the risk of false loop closures. The Town sequences present additional challenges with narrow alleyways, wide boulevards, and numerous dynamic objects such as pedestrians and cars. Even under these conditions, our approach ranks among the top two methods, showing its resilience in complex urban environments.

Additionally, the results on HeLiPR Bridge sequences provide insight into how the baselines perform under different levels of dynamic activity in the scene. The Bridge01 sequence, recorded at night, contains relatively few dynamic objects, whereas Bridge02, recorded during the day, includes substantially more dynamic participants. Since both sequences capture the same driving environment, we can directly compare their results to assess how dynamic objects affect loop closure detection. Our method performs consistently in terms of average precision and F1 score, highlighting its robustness to such dynamic elements. As expected, scan-based methods also show little to no variation in the metrics across these sequences, since individual scans are not significantly affected by dynamic objects due to the short temporal window of a single LiDAR scan.

The ground alignment strategy introduced in Sec. 3.2 further strengthens performance on datasets with non-planar LiDAR motion, such as the NCLT dataset and our self-recorded IPB-Backpack sequence. As shown in Tab. 2, our method outperforms Scan Context, BEVPlace++, and MapClosure, which assume planar motion for the BEV projection. We achieve the best values across all three metrics, except for average precision on the IPB-Backpack dataset, where the performance gap to BEVPlace++ remains marginal. Consistent BEV projection onto the local ground plane during revisits, even under varying pitch and roll, drives this performance. On the IPB-Car dataset, recorded in hilly terrain with vegetation and elevation changes, our method achieves almost 100% average precision, with a near-perfect precision-recall curve, confirming its robustness to complex ground plane variations.

The results also confirm the importance of local maps for extracting meaningful structural information for place recognition and loop closure detection. Among the baselines, only STD, BTC, and MapClosure perform competitively, and all rely on local maps. Nonetheless, our evaluation of STD and BTC with the same local maps as our pipeline, referred to as STD-100 and BTC-100, shows that our method’s advantage does not stem solely from larger local map sizes, but from the full algorithmic design.

In contrast, scan-based global descriptor approaches such as Scan Context and SOLiD fail to achieve consistent performance. Scan Context, designed for rotating LiDAR sensors, performs poorly in the HeLiPR Ouster OS2-128 and MulRan sequences due to its strong dependence on viewpoint. SOLiD, designed to handle LiDARs with different FoVs, performs better on the HeLiPR Aeva Aeries II and Livox Avia sequences but still falls short of local map-based methods. These trends highlight the inherent difficulty in constructing global descriptors for single LiDAR scans.

The weak performance of LoGG3D further illustrates this limitation. Without spatio-temporal aggregation or dimensionality reduction, LoGG3D remains highly sensitive to the point density within each scan. Trained exclusively on sequences from the MulRan dataset with Ouster OS1-64, it

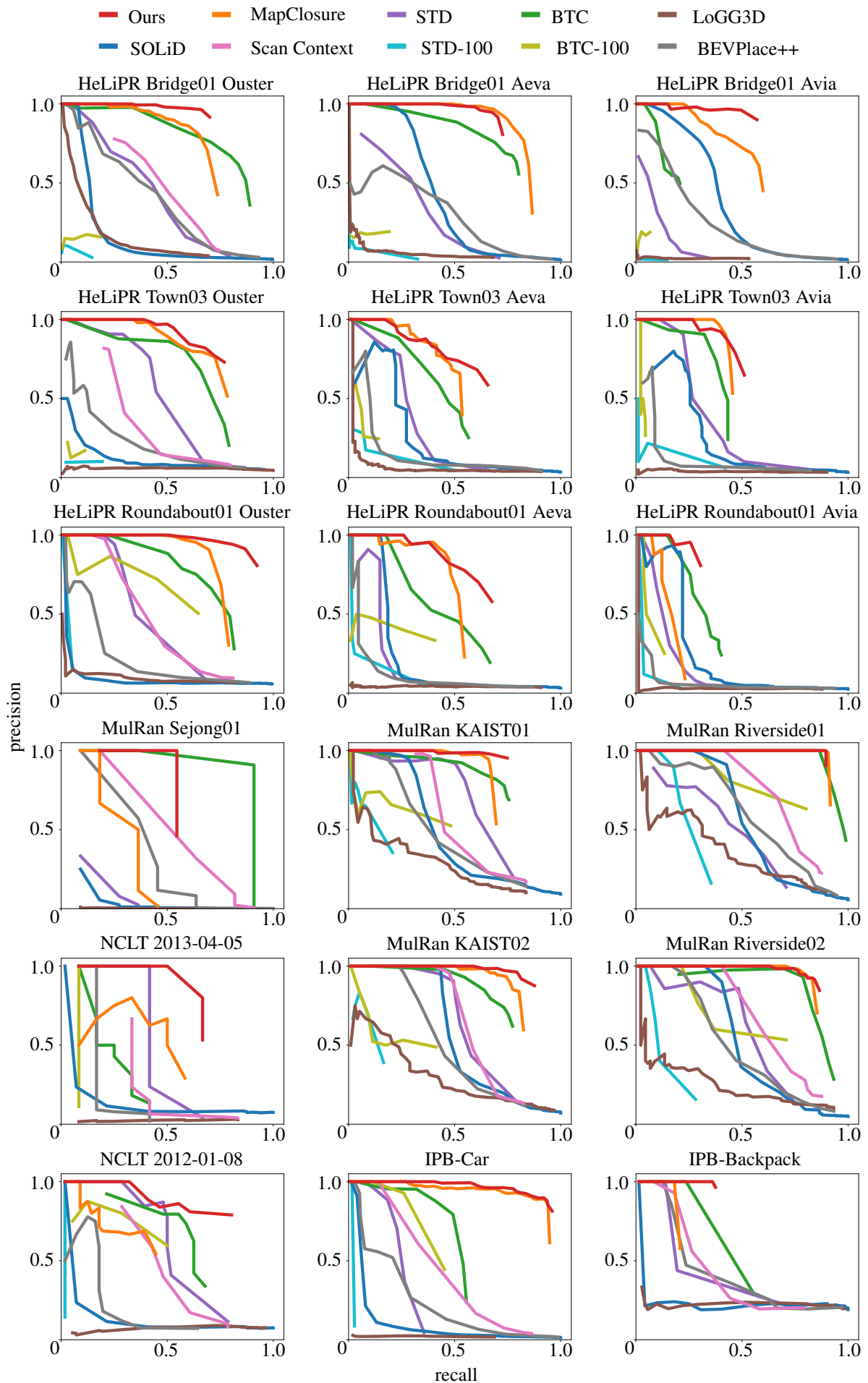


Figure 10. The precision-recall curves of state-of-the-art baselines and our approach for intra-session loop closure detection.

Table 1. Average precision (AP), maximum recall at 100% precision (R@1), and maximum F1 scores (F1_m) of state-of-the-art baselines and our approach for intra-session loop closure detection. Larger values indicate better performance. The best values are in bold, and the second-best values are underlined. The R@1 field is marked with “-” when the baseline cannot achieve a 100% precision upon varying its corresponding thresholding parameter.

(a) HeLiPR dataset → Ouster OS2-128															
Datasets	Bridge01			Bridge02			Roundabout01			Town02			Town03		
Metrics	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m
STD	0.344	0.049	0.460	0.300	-	0.469	0.316	0.200	0.420	0.293	0.059	0.421	0.373	0.037	0.538
STD-100	0.006	-	0.048	0.024	-	0.121	0.008	0.005	0.078	0.005	0.008	0.076	0.017	-	0.133
BTC	0.733	0.005	<u>0.730</u>	0.644	0.009	0.719	0.627	0.220	0.689	0.651	0.034	<u>0.703</u>	0.584	0.022	0.678
BTC-100	0.029	-	0.171	0.247	-	0.393	0.446	0.031	0.566	0.098	-	0.286	0.013	-	0.135
SOLiD	0.136	0.079	0.211	0.181	0.009	0.226	0.072	0.015	0.121	0.098	0.008	0.167	0.106	-	0.167
LoGG3D	0.125	0.007	0.199	0.127	0.014	0.188	0.077	-	0.162	0.065	-	0.128	0.059	-	0.112
BEVPlace++	0.347	0.025	0.445	0.273	0.009	0.423	0.167	0.020	0.227	0.211	0.068	0.263	0.168	-	0.262
SC	0.186	-	0.489	0.166	-	0.558	0.217	0.144	0.412	0.151	-	0.382	0.104	-	0.351
MapClosure	0.662	<u>0.219</u>	0.720	0.752	<u>0.470</u>	<u>0.781</u>	<u>0.731</u>	<u>0.472</u>	<u>0.788</u>	0.568	0.119	0.690	0.694	<u>0.343</u>	<u>0.743</u>
Ours	<u>0.688</u>	0.332	0.794	<u>0.685</u>	0.614	0.797	0.892	0.508	0.890	<u>0.607</u>	0.424	0.750	0.711	0.388	0.749

(b) HeLiPR dataset → Aeva Aeries II															
Datasets	Bridge01			Bridge02			Roundabout01			Town02			Town03		
Metrics	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m
STD	0.191	-	0.376	0.170	0.005	0.340	0.087	-	0.252	0.082	0.055	0.177	0.242	0.010	0.369
STD-100	0.012	-	0.080	0.056	0.005	0.168	0.021	0.009	0.082	0.000	-	0.058	0.032	-	0.113
BTC	0.609	0.076	0.722	0.511	0.010	0.679	0.377	<u>0.176</u>	0.483	0.259	-	0.429	0.371	0.020	0.494
BTC-100	0.033	-	0.192	0.201	-	0.392	0.157	-	0.367	0.029	-	0.150	0.033	-	0.180
SOLiD	0.378	0.166	0.451	0.156	-	0.404	0.211	0.148	0.277	0.068	0.027	0.171	0.235	-	0.338
LoGG3D	0.041	0.003	0.103	0.063	0.010	0.162	0.038	-	0.083	0.030	-	0.071	0.058	0.020	0.106
BEVPlace++	0.265	-	0.414	0.240	0.005	0.417	0.082	0.018	0.139	0.035	-	0.098	0.121	-	0.168
MapClosure	0.814	0.488	0.818	0.834	0.493	0.785	<u>0.478</u>	0.139	<u>0.607</u>	0.196	0.192	0.356	0.465	0.202	<u>0.584</u>
Ours	<u>0.711</u>	<u>0.424</u>	<u>0.796</u>	<u>0.683</u>	<u>0.488</u>	<u>0.765</u>	0.584	0.259	0.654	<u>0.206</u>	0.110	0.358	0.543	<u>0.172</u>	0.642

(c) HeLiPR dataset → Livox Avia															
Datasets	Bridge01			Bridge02			Roundabout01			Town02			Town03		
Metrics	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m
STD	0.056	-	0.167	0.073	0.005	0.178	0.104	0.024	0.208	0.062	0.035	0.171	0.290	0.111	0.357
STD-100	0.002	-	0.025	0.013	0.010	0.065	0.015	0.012	0.056	0.000	-	0.028	0.032	-	0.107
BTC	0.111	0.040	0.293	0.211	-	0.476	<u>0.241</u>	<u>0.146</u>	<u>0.394</u>	<u>0.133</u>	0.035	0.330	0.346	0.022	0.500
BTC-100	0.011	-	0.098	0.105	-	0.288	0.050	0.037	0.175	0.000	0.018	0.034	0.010	0.022	0.079
SOLiD	0.365	0.052	0.455	0.400	0.038	0.462	0.223	0.024	0.346	0.064	-	0.098	0.212	-	0.358
LoGG3D	0.013	-	0.045	0.020	-	0.057	0.026	0.012	0.059	0.023	0.018	0.053	0.036	-	0.084
BEVPlace++	0.205	-	0.302	0.200	0.014	0.321	0.052	-	0.112	0.030	-	0.092	0.078	-	0.144
MapClosure	<u>0.512</u>	0.222	<u>0.609</u>	0.562	<u>0.229</u>	<u>0.605</u>	0.133	0.073	0.252	0.102	0.105	0.265	0.431	0.367	<u>0.576</u>
Ours	0.549	<u>0.150</u>	0.698	<u>0.543</u>	0.300	0.688	0.278	0.158	0.442	0.141	<u>0.088</u>	<u>0.269</u>	0.464	<u>0.267</u>	0.587

performs well only on that dataset and generalizes poorly elsewhere. It fails to reach 100% precision even at low recall values in most sequences, underscoring the need to retrain such methods for each sensor setup.

BEVPlace++, also a learning-based method, performs more robustly than LoGG3D does due to its density-preserving BEV projection. However, it struggles on NCLT and IPB-Backpack sequences because of its assumption of planar motion for BEV projection.

Overall, these experiments confirm our core claim: our method delivers state-of-the-art performance in intra-session loop closure detection across datasets and LiDAR platforms. By leveraging local maps as the core representation, our pipeline balances precision and recall, remains robust

under perceptual aliasing, adapts to non-planar motion, and generalizes across different sensor characteristics, all without requiring changes to the pipeline parameters.

5.2 Inter-Session and Inter-LiDAR Loop Closure Detection

In this experiment, we evaluate the performance of our approach for inter-session and inter-LiDAR loop closure detection and compare it against state-of-the-art baselines introduced in Sec. 4.2. We select sequences spanning revisit intervals from a few weeks to several years to test the robustness under diverse temporal conditions.

Our pipeline architecture remains fundamentally the same in this scenario, with only minor modifications. We

Table 2. Average precision (AP), maximum recall at 100% precision (R@1), and maximum F1 scores (F1_m) of state-of-the-art baselines and our approach for intra-session loop closure detection. Larger values indicate better performance. The best values are in bold, and the second-best values are underlined. The R@1 field is marked with “-” when the baseline cannot achieve a 100% precision upon varying its corresponding thresholding parameter.

(a) MulRan dataset → Ouster OS1-64

Datasets	KAIST01			KAIST02			Riverside01			Riverside02			Sejong01		
	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m
STD	0.545	0.030	0.655	0.462	0.258	0.640	0.329	-	0.488	0.443	0.067	0.619	0.016	-	0.143
STD-100	0.097	0.007	0.262	0.080	-	0.232	0.203	0.104	0.318	0.103	0.045	0.202	0.000	0.000	0.000
BTC	<u>0.680</u>	0.007	0.756	0.605	0.103	0.736	0.916	0.865	0.927	0.645	-	0.854	<u>0.289</u>	<u>0.364</u>	0.909
BTC-100	0.293	0.007	0.502	0.224	0.010	0.447	0.615	0.292	0.706	0.456	0.214	0.609	0.000	0.000	0.000
SOLiD	0.356	0.104	0.462	0.457	0.433	0.604	0.505	0.271	0.582	0.471	0.326	0.558	0.010	-	0.133
LoGG3D	0.246	0.022	0.362	0.290	-	0.353	0.337	0.021	0.412	0.225	0.022	0.319	0.001	-	0.018
BEVPlace++	0.387	0.111	0.421	0.418	0.237	0.464	0.490	0.062	0.556	0.428	0.168	0.451	0.195	0.091	0.444
SC	0.176	-	0.550	0.169	0.402	0.636	0.240	0.417	0.684	0.190	0.404	0.587	0.162	0.182	0.424
MapClosure	0.671	0.437	<u>0.777</u>	0.790	<u>0.505</u>	<u>0.828</u>	<u>0.902</u>	<u>0.885</u>	0.946	<u>0.831</u>	0.708	<u>0.876</u>	0.083	0.182	0.421
Ours	0.733	<u>0.400</u>	0.838	0.852	0.588	0.876	0.885	0.896	<u>0.945</u>	0.843	<u>0.629</u>	0.877	0.364	0.546	0.706

(b) NCLT dataset → Velodyne HDL-32E; IPB-Car dataset → Ouster OS1-128; IPB-Backpack dataset → Hesai Pandar-128

Datasets	NCLT 2012-01-08			NCLT 2013-04-05			IPB-Car			IPB-Backpack		
	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m	AP	R@1	F1 _m
STD	<u>0.429</u>	<u>0.286</u>	0.621	0.219	0.417	<u>0.588</u>	0.219	0.068	0.344	0.261	0.139	0.323
STD-100	0.000	0.018	0.035	0.000	0.000	0.000	0.013	0.016	0.042	0.000	-	0.053
BTC	0.333	-	<u>0.654</u>	0.104	0.083	0.333	0.470	0.036	0.607	0.302	<u>0.236</u>	0.383
BTC-100	0.319	-	0.544	0.000	0.083	0.154	0.331	0.112	0.448	0.000	0.083	0.154
SOLiD	0.092	0.018	0.153	0.092	0.018	0.153	0.078	0.012	0.120	0.216	0.014	0.365
LoGG3D	0.070	-	0.161	0.019	-	0.060	0.014	-	0.045	0.215	-	0.355
BEVPlace++	0.154	-	0.274	0.101	0.167	0.286	0.196	0.028	0.297	0.370	0.125	<u>0.398</u>
SC	0.144	-	0.495	0.031	-	0.444	0.168	-	0.426	0.253	0.083	0.358
MapClosure	0.321	0.089	0.500	<u>0.307</u>	-	0.571	<u>0.899</u>	<u>0.272</u>	<u>0.906</u>	0.169	0.181	0.306
Ours	0.703	0.321	0.796	0.506	0.500	0.727	0.929	0.404	0.909	<u>0.361</u>	0.361	0.540

store the ground-alignment transformations for each local map together with the HBST database from the reference sessions. For a given query session, we match the ORB features against the reference database without adding any features from the query session to the database. Then, using the ground-alignment transforms of the corresponding local map from the reference session and the query local map, we compute a complete $\mathbb{SE}(3)$ pose estimate to align the loop closure between the two sessions.

We first evaluate inter-session loop closures using the same LiDAR sensor. The precision-recall curves in Fig. 11 and the metrics reported in Tab. 3 reveal trends consistent with the intra-session results. Our method consistently achieves the highest recall at 100% precision across different LiDARs and revisit periods, confirming its reliability in challenging conditions. The precision-recall curves highlight this robustness, as they terminate earlier but maintain high precision. Our method also ranks among the top two in average precision and maximum F1 scores, with BTC being the only baseline matching our overall performance. Other baselines often fail to reach 100% precision or achieve it only at very low recall, except in the MulRan KAIST01–KAIST03 sequences.

The Sejong02–Sejong03 sequences from the MulRan dataset were recorded as opposite-direction traversals on a long highway, which produces sparse structural geometry throughout the sequences. The restricted FoV of the dataset’s

LiDAR further increases the difficulty of inter-session loop closure detection. As shown in Tab. 3a, all baselines perform substantially worse on this pair of sequences, especially when we compare their results to those on the KAIST01–KAIST03 sequences from the same dataset, recorded with the same setup but in a dense urban environment.

The Bridge sequences from the HeLiPR dataset underscore the benefit of our self-similarity pruning strategy. Strong perceptual aliasing causes most baselines to either miss 100% precision entirely or reach it only at negligible recall. Even BTC, which performs comparably to our method in most other scenarios, fails to maintain its high precision on the Bridge sequences because it lacks an explicit mechanism to handle perceptual aliasing. In contrast, our approach detects accurate loop closures at comparatively higher recall, as reflected in the clear separation of our precision-recall curves from the baselines.

Additionally, we achieve better performance across different LiDAR sensors on the same sequences without changing any parameters in our pipeline. As shown in Tab. 3b, the Aeva Aeries II LiDAR experiences a sharp drop in maximum recall at 100% precision on the Bridge02–Bridge03 sequences. This drop arises from the overlap threshold used to define ground-truth loop closures: a few valid closures fall below the threshold and are therefore marked as false positives in ground-truth local maps, which lowers the maximum recall at 100% precision. Nevertheless,

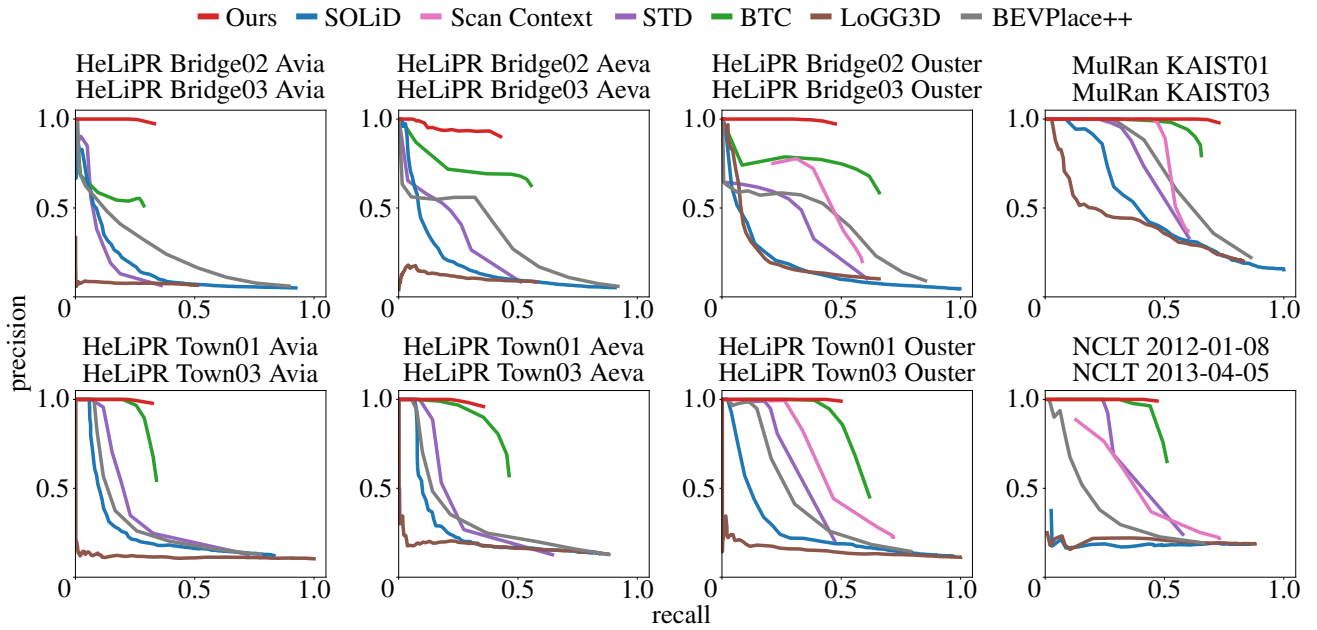


Figure 11. The precision-recall curves of state-of-the-art baselines and our approach for inter-session loop closure detection.

Table 3. Average precision (AP), maximum recall at 100% precision (R@1), and maximum F1 scores ($F1_m$) of state-of-the-art baselines and our approach for inter-session loop closure detection between sequences recorded with the same LiDAR sensor. Larger values indicate better performance. Best values are in bold, and the second-best values are underlined. The R@1 field is marked with “-” when the baseline cannot achieve a 100% precision upon varying its corresponding thresholding parameter.

(a) Conventional spinning LiDAR sensors

Datasets	MulRan			NCLT			HeLiPR								
Revisit Period	≈ 2 months			>1 year			2 weeks			4 weeks					
Reference Seq.	KAIST01			Sejong02			2012-01-08			Bridge02					
Query Seq.	KAIST03			Sejong03			2013-04-05			Bridge03					
LiDAR	Ouster OS1-64			Ouster OS1-64			Velodyne HDL-32E			Ouster OS2-128			Ouster OS2-128		
Metrics	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$
STD	0.451	0.223	0.529	0.095	-	0.224	0.297	0.241	0.408	0.220	-	0.396	0.266	0.084	0.360
BTC	<u>0.635</u>	0.309	<u>0.740</u>	0.378	<u>0.005</u>	0.526	0.465	<u>0.311</u>	0.605	0.490	0.001	0.647	<u>0.461</u>	<u>0.259</u>	<u>0.636</u>
SOLiD	<u>0.472</u>	0.085	0.435	0.022	-	0.055	0.153	-	0.307	0.157	0.004	0.223	<u>0.239</u>	0.024	0.276
LoGG3D	0.342	0.027	0.424	0.017	-	0.069	0.179	-	0.312	0.152	<u>0.005</u>	0.205	0.149	0.002	0.222
BEVPlace++	0.547	0.276	0.575	0.053	-	0.149	0.238	0.017	0.308	0.335	0.002	0.468	0.316	0.025	0.350
SC	0.160	<u>0.445</u>	0.643	0.039	-	0.140	0.252	-	0.426	0.205	-	0.500	0.247	-	0.479
Ours	0.725	0.633	0.835	<u>0.203</u>	0.037	<u>0.346</u>	<u>0.464</u>	0.411	0.637	<u>0.472</u>	0.300	<u>0.638</u>	0.497	0.437	0.664

(b) Unconventional hybrid solid-state and solid-state, non-spinning LiDAR sensors

Datasets	HeLiPR											
Revisit Period	2 weeks			4 weeks								
Reference Seq.	Bridge02			Town01								
Query Seq.	Bridge03			Town03								
LiDAR	Livox Avia		Aeva Aeries II		Livox Avia		Aeva Aeries II					
Metrics	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$
STD	0.087	-	0.168	0.171	0.001	0.319	0.238	0.066	0.283	0.226	<u>0.088</u>	0.270
BTC	0.161	0.001	<u>0.368</u>	0.391	0.008	0.593	<u>0.295</u>	0.198	<u>0.442</u>	0.415	0.068	0.549
SOLiD	0.151	0.002	0.209	0.163	<u>0.013</u>	0.212	0.218	0.058	0.248	0.224	0.066	0.254
LoGG3D	0.039	-	0.125	0.065	-	0.158	0.114	0.002	0.194	0.152	0.002	0.249
BEVPlace++	<u>0.207</u>	<u>0.009</u>	0.302	0.279	0.002	0.408	0.224	0.078	0.268	0.252	0.054	0.298
Ours	0.328	0.223	0.494	0.404	0.060	<u>0.580</u>	0.319	0.218	0.485	<u>0.349</u>	0.215	<u>0.519</u>

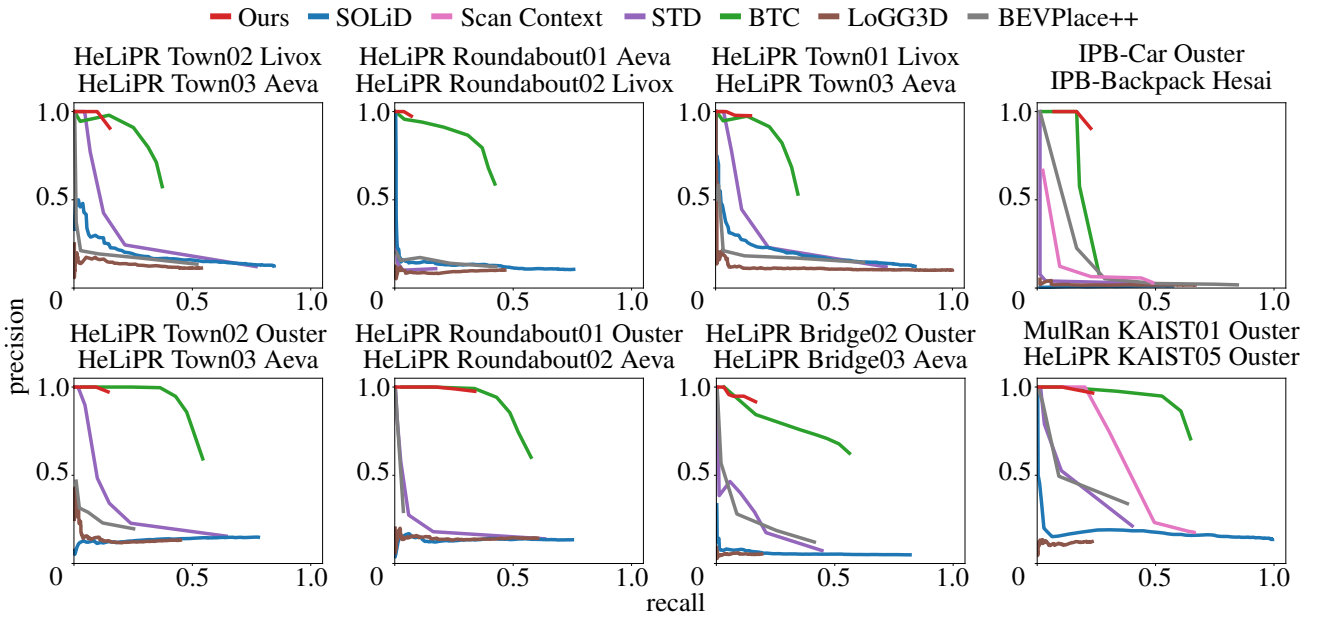


Figure 12. The precision-recall curves of state-of-the-art baselines and our approach for inter-LiDAR loop closure detection.

Table 4. Average precision (AP), maximum recall at 100% precision (R@1), and maximum F1 scores ($F1_m$) of state-of-the-art baselines and our approach for inter-session loop closure detection between sequences recorded with different LiDAR sensors. Larger values indicate better performance. The best values are in bold, and the second-best values are underlined. The R@1 field is marked with “-” when the baseline cannot achieve a 100% precision upon varying its corresponding thresholding parameter.

(a)

Datasets	HeLiPR														
Revisit Interval	2 weeks			4 weeks			\approx 2 weeks			2 weeks					
Reference Seq.	Bridge02			Bridge01			Roundabout01			Town02					
Query Seq.	Bridge03			Bridge03			Roundabout02			Town03					
Reference LiDAR	Ouster OS2-128			Ouster OS2-128			Ouster OS2-128			Aeva Aeries II			Ouster OS2-128		
Query LiDAR	Aeva Aeries II			Aeva Aeries II			Aeva Aeries II			Livox Avia			Aeva Aeries II		
Metrics	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$
STD	0.086	0.003	0.208	0.061	-	0.188	0.109	0.003	0.229	0.019	-	0.134	0.165	0.017	0.250
BTC	0.398	0.033	0.592	0.352	0.004	0.520	0.499	<u>0.152</u>	0.620	0.361	0.002	0.504	0.478	0.243	0.612
SOLiD	0.048	-	0.097	0.028	-	0.072	0.102	-	0.228	<u>0.099</u>	<u>0.004</u>	0.198	0.105	-	0.251
LoGG3D	0.010	-	0.084	0.006	-	0.061	0.086	-	0.232	0.043	-	<u>0.169</u>	0.063	-	0.204
BEVPlace++	0.079	0.003	0.215	0.016	<u>0.002</u>	0.108	0.010	0.001	0.065	0.058	-	0.188	0.055	-	0.221
Ours	<u>0.156</u>	<u>0.032</u>	<u>0.280</u>	<u>0.082</u>	-	0.176	<u>0.332</u>	0.176	<u>0.502</u>	0.067	0.039	0.130	0.141	<u>0.093</u>	<u>0.251</u>

(b)

Datasets	HeLiPR						MulRan \times HeLiPR						Self-recorded		
Revisit Interval	2 weeks			4 weeks			$>$ 4 years						2 weeks		
Reference Seq.	Town02			Town01			MulRan KAIST01						IPB-Car		
Query Seq.	Town03			Town03			HeLiPR KAIST05						IPB-Backpack		
Reference LiDAR	Livox Avia			Livox Avia			Ouster OS1-64			Ouster OS1-64			Ouster OS1-128		
Query LiDAR	Aeva Aeries II			Aeva Aeries II			Ouster OS2-128			Aeva Aeries II			Hesai Pandar-128		
Metrics	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$	AP	R@1	$F1_m$
STD	0.159	<u>0.045</u>	0.229	0.159	<u>0.034</u>	0.226	0.124	0.014	0.278	0.108	-	<u>0.271</u>	0.012	<u>0.012</u>	0.052
BTC	0.325	0.002	0.468	0.297	0.004	0.437	0.585	0.100	0.713	0.410	0.087	0.579	0.156	0.167	<u>0.286</u>
SOLiD	<u>0.163</u>	-	0.242	0.171	0.002	0.250	0.171	0.002	0.266	<u>0.156</u>	-	0.252	0.005	-	0.025
LoGG3D	0.073	-	0.187	0.113	0.002	0.186	0.027	-	0.162	0.051	-	0.194	0.013	-	0.050
BEVPlace++	0.083	0.003	0.215	0.105	-	0.236	0.151	0.002	0.359	0.140	-	<u>0.303</u>	0.054	<u>0.012</u>	0.192
SC	N/A	N/A	N/A	N/A	N/A	N/A	<u>0.251</u>	0.202	<u>0.432</u>	N/A	N/A	N/A	0.031	-	0.107
Ours	0.145	0.098	<u>0.259</u>	0.143	0.044	<u>0.256</u>	0.229	<u>0.106</u>	0.377	0.091	<u>0.054</u>	0.200	<u>0.149</u>	0.167	0.362

the average precision and the corresponding precision-recall curve in Fig. 11 demonstrate that our method still performs strongly, maintaining near-100% precision across a substantially larger recall range.

On the NCLT dataset, which includes a revisit after about one year with a low-resolution Velodyne HDL-32E LiDAR, our approach again achieves the best recall at 100% precision and maximum F1 score. Only STD and BTC are the baselines that perform comparably in this scenario. This emphasizes the importance of having explicit knowledge about the ground plane under non-planar LiDAR motion.

These results also highlight the challenges that scan-based global descriptor methods such as Scan Context, SOLiD, and LoGG3D face in long-term revisit scenarios. In such scenarios, significant lateral shifts, viewpoint changes during revisits, and scene changes can all alter the global descriptor substantially. In contrast, local map-based local descriptor methods like ours, STD, and BTC avoid these issues by aggregating multiple scans to reduce viewpoint dependence and by focusing on local regions of the scene that remain similar across revisits, rather than compressing the local context into a single global descriptor.

The results on the HeLiPR sequences further highlight the impact of the LiDAR type on long-term place recognition. Even among the stronger baselines, we observe a noticeable drop in performance when switching from a 360°-horizontal-FoV Ouster OS2-128 LiDAR to the limited-horizontal-FoV LiDAR sensors such as Aeva Aeries II and Livox Avia, even when evaluating the same pair of sequences.

We next evaluate inter-LiDAR loop closure detection across multiple sessions. As shown in Tab. 4, our approach ranks second overall to BTC but achieves the best recall at 100% precision on five of ten sequences. The precision-recall curves in Fig. 12 highlight the same trend: our method often detects fewer closures but preserves high precision, ensuring reliable results without extensive parameter tuning.

The performance of scan-based methods like Scan Context, SOLiD, LoGG3D, and BEVPlace++ is adversely affected by the difference in the scanning patterns, FoVs, and resolutions across LiDAR sensors. The local map-based methods achieve better results in such cross-LiDAR scenarios.

On the Roundabout01–Roundabout02, Town01–Town03, and Town02–Town03 sequences, our pipeline achieves the best recall at 100% precision among all baselines, demonstrating robustness to opposite traversals and constrained sensor views. However, on Bridge01–Bridge03 and Bridge02–Bridge03 sequences, the feature-pruning strategy limits recall to very low values, leaving room for improvement.

Cross-dataset evaluations between MulRan and HeLiPR push this further, with revisit intervals of over 4 years and different LiDAR sensors. These scenarios introduce structural changes in addition to sensor differences, making them highly challenging. Even so, our method achieves the second-best recall at 100% precision, while all baselines, including ours, achieve lower recall overall.

Finally, in self-recorded datasets collected two weeks apart with platforms having different motion profiles and LiDARs, our method performs comparably to BTC across

all metrics, confirming its ability to generalize across diverse sensor setups.

Overall, these evaluations demonstrate that our approach effectively detects loop closures across multiple sessions for different LiDAR sensors and revisit intervals, thereby supporting our second key claim. At the same time, they highlight opportunities for further research on the highly challenging problem of inter-LiDAR loop closure detection.

5.3 Analysis and Evaluation of the Ground Alignment Module

In this section, we present both qualitative and quantitative evaluations of the ground alignment stage, which enhances the detection and alignment of loop closures. As we describe in Sec. 3.2, we need this stage in scenarios where the LiDAR experiences non-planar motion, allowing us to maintain a consistent reference plane for the BEV projection across revisits.

When we turn off the ground alignment module, our approach projects local maps onto the xy-plane, assuming planar motion. Consequently, using the RANSAC-based validation, it can align the local maps only along the xy-plane. We illustrate such a 2D alignment in the top-left image of Fig. 13, where two local maps appear well-aligned from a top-down view. However, this alignment can be misleading. In the top-right image of Fig. 13, a different perspective reveals an apparent misalignment in height and tilt between the two local maps, caused by non-planar LiDAR motion.

By applying explicit ground alignment, we ensure that the local maps involved in a loop closure align correctly in 3D. This improvement enables our method to perform complete 3D global alignment rather than relying solely on a 2D assumption. The second row of Fig. 13 shows a more accurate initial alignment between the local map pairs, even before any point cloud registration takes place.

We further validate this improvement through quantitative analysis. For each detected loop closure, we compute the overlap between the corresponding local maps after applying our initial transformation estimate mT_r . We evaluate this on two sequences with non-planar LiDAR motion: the 2013-04-05 sequence from the NCLT dataset (Carlevaris-Bianco et al. 2016) and the IPB-Backpack sequence recorded using our in-house setup. In Tab. 5, we report the number of true positive loop closures with at least five inliers and the root mean square error of the overlap between aligned local maps, compared to the overlap between their reference counterparts generated with ground-truth poses.

As seen in Tab. 5, the ground alignment stage significantly reduces the overlap error between local maps. This improvement occurs without applying any local point cloud registration. This advantage becomes evident in datasets with strong non-planar motion, such as the IPB-Backpack setup, where roll and pitch angles vary approximately between -30° to 30° . In such cases, the proposed ground alignment module enables our system to detect substantially more loop closures. This is because the BEV density images remain consistent across revisits when we project them onto the actual ground plane in the environment rather than onto the xy-plane of the local map frame.

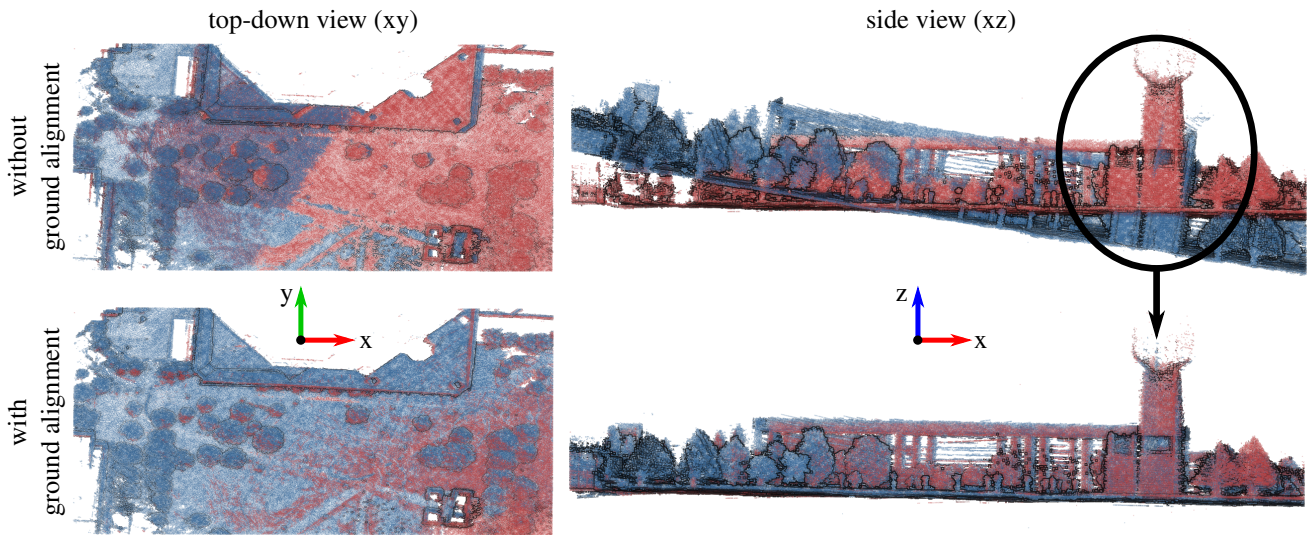


Figure 13. Visual comparison of loop-closed local map alignments with and without ground alignment. The top-down view (xy) shows both local maps to be aligned in either case, as RANSAC operates in 2D. However, the side view (xz) reveals significant misalignment without ground alignment due to uncorrected pitch and roll differences, as highlighted by the black ellipse. Applying ground alignment provides consistent 3D alignment by projecting the local maps onto a common physical ground plane.

Table 5. Evaluation of the ground alignment strategy for loop closure detection and local map alignment. We report the number of correctly identified loop closures and the root mean square (RMS) error in the overlap between aligned local maps using our pose estimate mT_r , and the overlap between the reference local maps. We report results with and without the proposed ground alignment module.

Dataset	NCLT 2013-04-05		IPB-Backpack	
	OFF	ON	OFF	On
Correct Loop Closures \uparrow	6	6	10	26
RMS Error in Overlap \downarrow	0.280	0.110	0.468	0.104
Recall \uparrow	0.500	0.500	0.139	0.361

We next evaluate the ability of our ground alignment module to correct misalignments between the ground plane and the map’s local xy -plane. To test this, we manually apply rotations in the range ($10^\circ - 80^\circ$) about ten random axes lying in the xy -plane of each reference local map. Our method then estimates the transform gT_m that realigns the ground plane with the map’s xy -plane. We perform this evaluation on the MulRan KAIST01 and Sejong01 sequences, as well as the HeLiPR Bridge01 sequence, using the Aeva Aeries II and Livox Avia LiDARs. These sequences feature near-planar LiDAR motion, so their reference local maps already have the ground plane aligned with the xy -plane.

In Fig. 14, we report the mean and standard deviation of the absolute error in the predicted rotation magnitude, averaged across all ten axes for each local map, as a function of the applied misalignment. The ground alignment module reliably corrects misalignments up to 60° , with a mean error below 1° and a standard deviation below 8° . Even at initial misalignments of around 80° , the mean absolute error typically remains below 10° , except in the Livox Avia sequence, where observed failures stem from the Livox Avia sensor’s unusual reflective artifacts. These artifacts produce misleading, consistently oriented below-ground points that disrupt low-lying points sampling and PCA filtering under artificially large rotations. We would like to highlight that

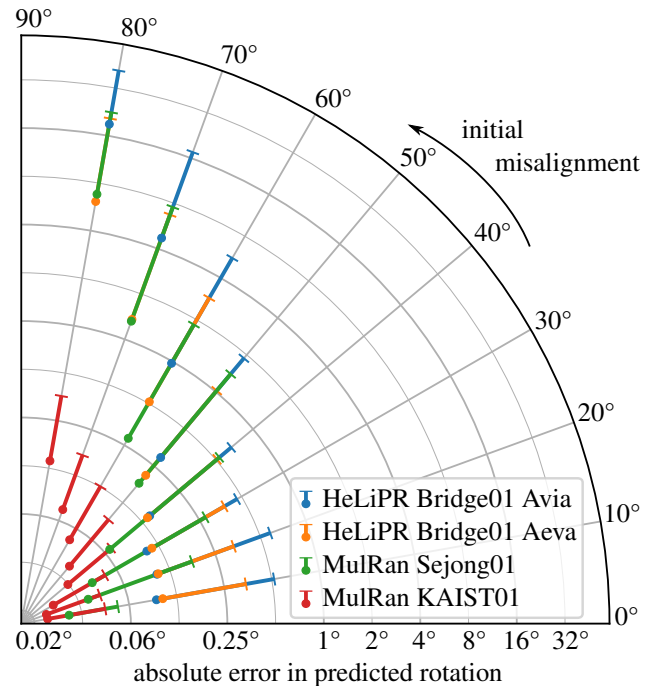


Figure 14. Quantitative evaluation of the ground alignment accuracy. The azimuthal axis indicates the absolute initial misalignment magnitude between the ground plane and xy -plane of the local maps. The radial axis (log scale) shows the absolute error of the predicted ground alignment magnitude, reported as mean values (dots) and standard deviation (bars), across different sequences and LiDARs from the MulRan and HeLiPR datasets.

this is a sensor-specific quirk and not a flaw in the alignment method. However, since real-world misalignments rarely approach such extreme values, this level of performance is sufficient for practical use.

These results support our third claim that the proposed ground alignment module enables robust handling of handheld or non-planar LiDAR motion and also provides a

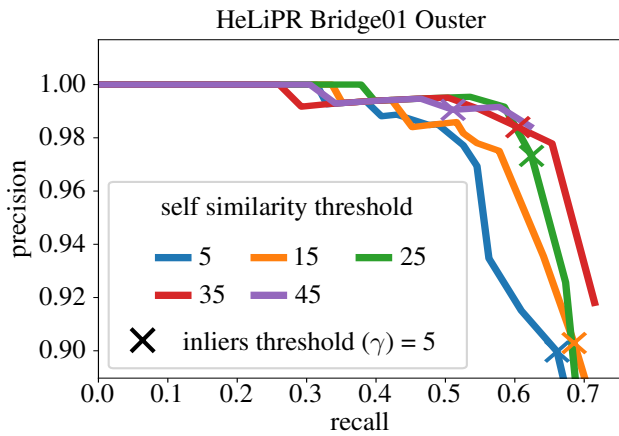


Figure 15. Precision-recall curves for various choices of pruning thresholds in terms of the Hamming distance between ORB descriptors. We visualize a zoomed-in precision-recall curve for better differentiation between different curves.

strong initial pose estimate for the 3D alignment of loop-closed local maps.

5.4 Analysis and Evaluation of the Self-Similarity Pruning Strategy

In this section, we evaluate the effectiveness of the self-similarity pruning strategy in detecting loop closures within environments characterized by highly repetitive structures. Specifically, we focus on the Bridge sequences from the HeLiPR dataset, which feature long bridges composed of repeating mechanical elements. These conditions present significant challenges due to perceptual aliasing, which can lead to false loop closures.

Fig. 15 shows a zoomed-in precision-recall curve for different self-similarity pruning thresholds based on the Hamming distance between ORB descriptors. The mark X on the curve represents the performance at an inlier threshold of 5, the default setting in our pipeline. Lower threshold values prune only strictly similar features, leading to higher recall values but a drop in precision, similar to the performance without pruning. Conversely, higher thresholds prune more features, leading to a drop in recall but maintaining high precision. The threshold value of 35 provides a desirable trade-off, maintaining a recall of around 0.6 with minimal loss in precision.

Without the pruning strategy, our method detects false loop closures in these Bridge sequences. We show one example at the bottom-left of Fig. 17, where a false match occurs with a near-perfect alignment between local maps. This alignment is supported by a sufficient number of inlier correspondences after a 2D RANSAC alignment of their BEV features (right-hand side of Fig. 17), primarily caused by repetitive structures on the bridge. We successfully eliminate these matches between similar structures by applying the feature pruning strategy described in Sec. 3.4, thereby preventing the false loop closure.

We further compare the performance of our pipeline with and without feature pruning in Fig. 16. The precision-recall curves for both configurations, across the two Bridge sequences and three different LiDAR sensors, show that our

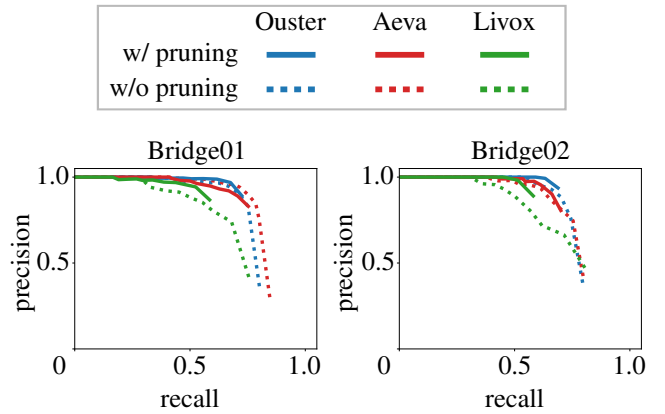


Figure 16. Precision-recall curves evaluating the impact of the pruning strategy for loop closure detection on the Bridge sequences from the HeLiPR dataset. These sequences are recorded in an environment with strong perceptual aliasing.

Table 6. The effect of feature pruning on loop closure accuracy, measured by RMS APE w.r.t. translation after pose-graph optimization. We report the results for three LiDAR sensors across two sequences from the HeLiPR dataset.

Metric	RMS APE w.r.t. Translation [m] ↓					
	HeLiPR Bridge01			HeLiPR Bridge02		
Sequence	Ouster	Aeva	Livox	Ouster	Aeva	Livox
LiDAR						
Odometry	178.8	124.4	223.0	52.4	149.3	114.9
w/o Pruning	331.1	282.7	695.9	572.6	314.1	457.7
w/ Pruning	23.9	28.7	51.0	16.6	23.5	56.4

approach with pruning maintains high precision throughout rather than compromising on precision to achieve a better recall. In contrast, our approach without pruning achieves slightly higher recall but significantly drops precision.

To underscore the importance of precision, we conduct a pose-graph optimization experiment using the loop closures detected under both settings. As shown in Tab. 6, loop closures obtained with pruning lead to markedly better absolute trajectory estimates despite detecting fewer closures. In contrast, disabling pruning results in more false positives, which degrade the final trajectory estimate, even surpassing the error of odometry-only results.

Overall, this study supports our fourth claim that the self-similarity feature pruning strategy introduced in Sec. 3.4 is essential for maintaining robustness in loop closure detection under high structural repetition. By prioritizing precision, our method avoids catastrophic failures in pose estimation, ultimately leading to more reliable mapping and localization.

5.5 Evaluation of the 3D Alignment Estimates

In this section, we evaluate the accuracy of the initial alignment (mT_r) for the loop closures detected by our approach across the HeLiPR, NCLT, and IPB-Car datasets. To obtain a reliable reference for evaluation, we refine the local map alignments using Open3D’s (Zhou et al. 2018) point-to-point ICP registration, initialized with the dataset ground-truth pose. We use these refined results strictly as a *proxy ground-truth alignment*, since directly comparing against the dataset poses would conflate alignment error with

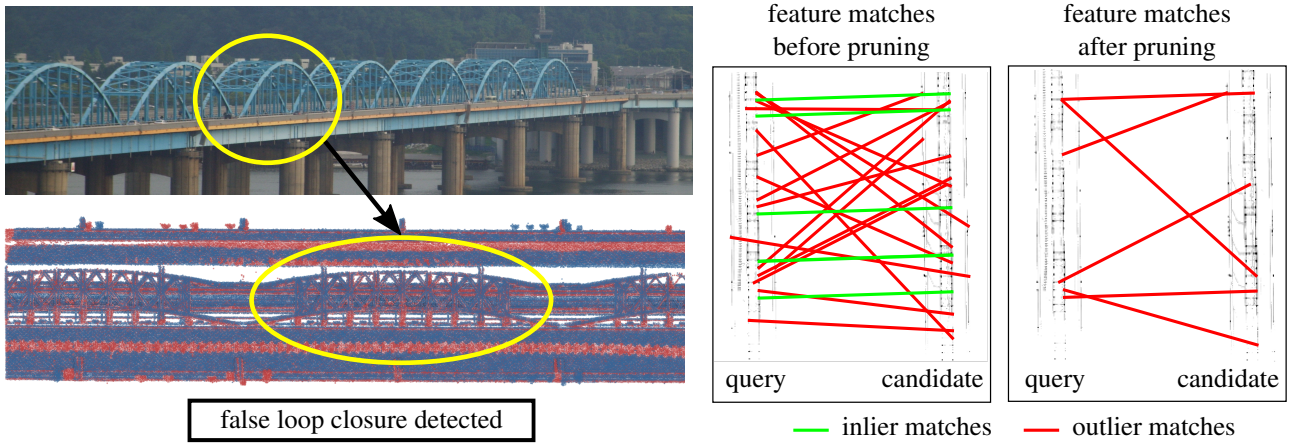


Figure 17. Visualization of perceptual aliasing in the HeLiPR Bridge sequence and the impact of feature pruning on loop closure detection. *Left:* The repetitive semi-circular arches of the bridge (top) lead to structurally similar local maps (below) as highlighted by the yellow ellipses, causing a false loop closure. *Right:* Feature correspondence plots before and after pruning. Without pruning, RANSAC mistakenly accepts many inlier correspondences (green lines), resulting in a false positive. After pruning, only outlier matches (red lines) remain, correctly preventing erroneous loop closure. Feature pruning effectively removes ambiguous, repetitive features that lead to perceptual aliasing.

Table 7. Evaluation of the loop closure alignment accuracy. We report accuracy in terms of RMS APE in translation (m) and rotation ($^{\circ}$) compared to the proxy ground-truth alignment obtained via Open3D point-to-point ICP. Best results are in **bold**.

(a) We report the number of true loop closures and the accuracy of the initial pose estimates (mT_r) produced by our method. Results from KISS-Matcher (KM) are also shown to contextualize the quality of our estimates; KISS-Matcher’s values correspond to only successful alignments.

Sequence	LiDAR	# Closures		RMS APE (translation) [m]		RMS APE (rotation) [$^{\circ}$]	
		KM	Ours	KM	Ours	KM	Ours
HeLiPR Bridge01	Ouster OS2-128	215	216	5.38 ± 5.34	5.57 ± 5.40	0.87 ± 0.83	1.80 ± 1.47
	Livox Avia	122	122	15.68 ± 14.95	2.07 ± 1.24	2.09 ± 1.87	2.74 ± 1.91
	Aeva Aeries II	213	214	1.65 ± 1.43	2.17 ± 1.50	1.06 ± 0.87	1.90 ± 1.44
HeLiPR Town02	Ouster OS2-128	42	42	0.30 ± 0.20	0.89 ± 0.44	0.31 ± 0.17	1.11 ± 0.84
	Livox Avia	5	5	1.07 ± 0.62	1.46 ± 0.73	0.65 ± 0.26	0.82 ± 0.44
	Aeva Aeries II	9	9	0.55 ± 0.23	1.31 ± 0.60	1.32 ± 0.96	1.62 ± 0.90
HeLiPR Roundabout01	Ouster OS2-128	125	139	0.33 ± 0.22	0.72 ± 0.33	0.33 ± 0.23	0.56 ± 0.33
	Livox Avia	12	13	0.44 ± 0.27	2.52 ± 1.88	0.56 ± 0.40	2.04 ± 1.45
	Aeva Aeries II	44	51	1.02 ± 0.74	1.63 ± 0.90	1.14 ± 0.83	1.14 ± 0.61
IPB-Car	Ouster OS1-128	216	230	0.49 ± 0.40	0.84 ± 0.49	0.50 ± 0.36	1.15 ± 0.80
NCLT 2012-01-08	Velodyne HDL-32E	14	26	1.06 ± 0.87	1.23 ± 0.85	1.06 ± 0.86	1.28 ± 0.97

(b) We refine the pose estimates from both methods using Open3D point-to-point ICP, and report the RMS APE in translation and rotation relative to the reference alignment. Both methods converge to similarly accurate solutions, demonstrating that our initial pose estimates are sufficiently precise to serve as reliable initialization for ICP-style refinement within a SLAM pipeline.

Sequence	LiDAR	RMS APE (translation) [m]		RMS APE (rotation) [$^{\circ}$]	
		KM + ICP	Ours + ICP	KM + ICP	Ours + ICP
HeLiPR Bridge01	Ouster OS2-128	5.05 ± 5.03	5.05 ± 5.05	1.12 ± 1.11	0.73 ± 0.73
	Livox Avia	15.67 ± 14.98	0.47 ± 0.43	2.02 ± 1.90	0.70 ± 0.69
	Aeva Aeries II	1.51 ± 1.41	0.60 ± 0.56	0.72 ± 0.70	0.40 ± 0.39
HeLiPR Town02	Ouster OS2-128	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.03 ± 0.01
	Livox Avia	0.04 ± 0.02	0.04 ± 0.02	0.02 ± 0.01	0.02 ± 0.01
	Aeva Aeries II	0.02 ± 0.01	0.02 ± 0.01	0.03 ± 0.01	0.03 ± 0.01
HeLiPR Roundabout01	Ouster OS2-128	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01	0.02 ± 0.01
	Livox Avia	0.23 ± 0.22	0.17 ± 0.16	0.45 ± 0.43	0.23 ± 0.22
	Aeva Aeries II	0.48 ± 0.47	0.26 ± 0.25	0.49 ± 0.48	0.25 ± 0.25
IPB-Car	Ouster OS1-128	0.30 ± 0.29	0.28 ± 0.28	0.17 ± 0.17	0.17 ± 0.16
NCLT 2012-01-08	Velodyne HDL-32E	0.63 ± 0.61	0.94 ± 0.89	0.65 ± 0.62	0.61 ± 0.57

odometry drift accumulated within the local maps that we generate using LiDAR odometry.

In Tab. 7a, we report the number of correctly identified loop closures and the RMS APE in translation and rotation for the pose estimates (mT_r) produced by our method. Our loop closure identification approach consistently yields initial alignments within a couple of meters of translation and a few degrees of rotation relative to the reference alignment. This performance holds across all evaluated LiDARs, despite their differing scanning patterns, resolutions, and FoVs. It can thus be used with ICP for fine alignment.

We also compare our method to KISS-Matcher by Lim et al. (2025) (KM), a state-of-the-art global point cloud registration method based on 3D features. For KISS-Matcher, we report the number of loop closures it successfully aligns (i.e., more than five inliers) and compute the corresponding RMS error metrics with respect to the reference alignment. We include this comparison not as a competing baseline, but to contextualize the quality of our *initial alignment*, and demonstrate how close our pose estimates are to those produced by a dedicated global registration system.

As shown in Tab. 7, KISS-Matcher, a technique designed to globally register point clouds, achieves better alignment accuracy across most sequences. This is expected as KISS-Matcher computes 3D features from the 3D point cloud and estimates the pose using a maximally consistent correspondence set. In contrast, our method combines 2D feature-based pose estimates with the approximate ground-alignment estimates to get a complete 3D pose. Despite this fundamental difference, our RMS APE values remain within a few decimeters in translation and a couple of degrees in rotation compared to those of KISS-Matcher. Notably, KISS-Matcher performs worse than our method on the HeLiPR Bridge01 sequence because it lacks an explicit mechanism to handle perceptual aliasing during alignment.

To further demonstrate that our pose estimates are well-suited as initialization for a fine ICP-style alignment stage within a SLAM system, we explicitly run a point-to-point ICP refinement using Open3D, initialized with the pose estimates from our method as well as KISS-Matcher. The results in Tab. 7b show that both methods converge to solutions that are similarly accurate, and close to the reference alignment.

Our refined results also outperform KISS-Matcher on several sequences. The apparent performance inversion between KISS-Matcher and our pose estimation accuracy before and after ICP refinement may seem counterintuitive at first. A plausible explanation lies in the non-linear least-squares optimization underlying the ICP algorithm, which is sensitive to initialization and can converge to different local minima depending on the initial pose estimate.

Furthermore, two local maps generated from the same location using LiDAR odometry will generally not align perfectly, as they accumulate different amounts of drift and may contain artifacts due to imperfect scan deskewing. This could also lead to a different convergence for different initializations. It is also important to note that the discrepancy in performance between the two methods in Tab. 7b is, in most cases, on the order of only a few centimeters. The overall trend is that both methods

converge to similarly accurate solutions after ICP refinement; therefore, we make no claims about the superiority of one method over the other in terms of final alignment accuracy.

Overall, these results support our fifth claim that the proposed approach yields accurate and complete 3D rigid-body transforms that align detected loop closures and are sufficiently precise to serve as initial guesses for fine-grained ICP-style registration within a SLAM back-end.

5.6 Multi-Map Alignment

This final experiment evaluates our pipeline’s ability to detect loop closures between the Riverside03 sequence and three KAIST sequences from the MulRan dataset. Although these trajectories share only minimal spatial overlap, it is sufficient to align the two scenes. These sequences also vary in temporal separation, with revisits ranging from the same day to a few months apart.

We first compare our method against baseline approaches using the precision-recall curves shown in Fig. 18. Our approach consistently outperforms the baselines, achieving the highest recall while maintaining 100% precision across all three cases. Most other methods, except Scan Context, struggle in this challenging setting, with precision often falling below 50%.

Next, we compute the optimized trajectory for each sequence using in-session loop closure constraints detected by our method, followed by an additional pose-graph optimization step to align these optimized sequences across sessions using the inter-session closures identified by our pipeline.

We show in Fig. 19 the final aligned trajectories and corresponding local maps, illustrating the successful alignment in the overlapping region. We report the precision, recall, and F1 scores for the inter-session loop closures in Tab. 8, using the default parameters detailed in Sec. 4.2. Despite the limited overlap, our method achieves perfect precision in all cases.

To assess the quality of the predicted alignment transformation mT_r , we compute the RMS error of the overlap between aligned local maps, compared to the overlap between their reference counterparts generated with ground-truth poses. The last row of Tab. 8 demonstrates that our alignment closely matches ground-truth performance.

Additionally, we perform a similar cross-session multi-map alignment between the two self-recorded datasets, IPB-Car and IPB-Backpack, captured with different LiDAR sensors mounted on separate mobile platforms. These sequences exhibit varying motion dynamics and sensor configurations but still contain overlapping regions. We show the resulting aligned trajectories and detected loop closures in Fig. 1.

This experiment supports our final claim that our approach can successfully detect and align loop closures in challenging scenarios with little overlap. This capability is critical for tasks such as multi-robot map alignment, collaborative mapping, and long-term change detection.

5.7 Runtime Evaluation of Our Approach

In this additional experiment, we evaluate the runtime of our approach. We run all the experiments on an Intel

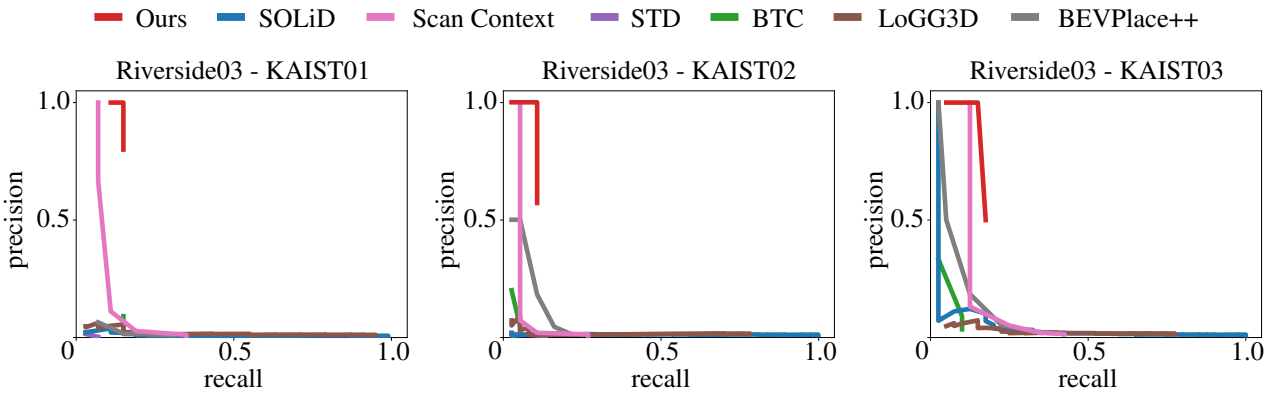


Figure 18. Precision-recall curves of our approach and state-of-the-art baselines for loop closure detection across sequences from the MulRan dataset with limited overlap.

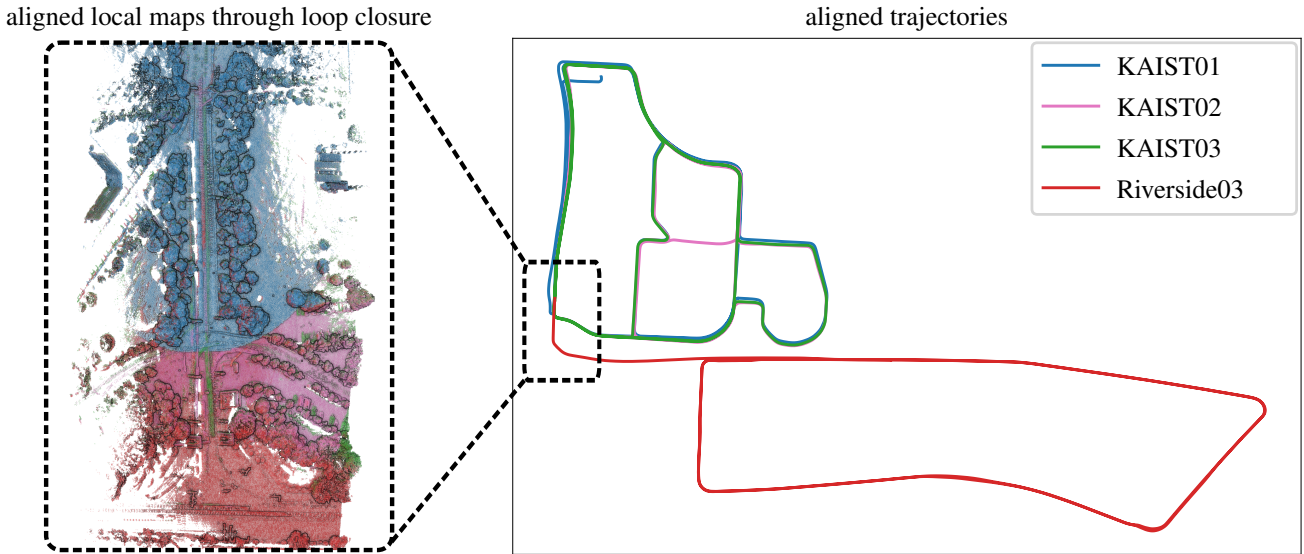


Figure 19. We align three KAIST sequences to the Riverside03 sequence from the MulRan dataset using our loop closure pipeline. Despite the minimal spatial overlap, highlighted by the dashed rectangle, our approach accurately detects loop closures and aligns the corresponding local maps.

Table 8. Precision, recall, and F1 score for multi-session loop closures between the Riverside03 and KAIST sequences from the MulRan dataset. We report the RMS error in the overlap between aligned local maps using our pose estimate mT_r and the overlap between the reference local maps.

Reference Session	Riverside03		
	KAIST01	KAIST02	KAIST03
Query Session			
Revisit Interval	2 Months	Same Day	1 Week
Precision \uparrow	1.0	1.0	1.0
Recall \uparrow	0.120	0.108	0.150
F1 \uparrow	0.214	0.195	0.261
RMS Error in Overlap \downarrow	0.043	0.055	0.085

i9-10980XE @ 3.00 GHz CPU with 64 GB RAM. We implement our approach in C++ with a single-thread design. We report the runtime of different components of our pipeline in Tab. 9, including the mean and standard deviations for performing ground alignment for a local map, detecting loop closures using BEV density images, and validating detected loop closures through RANSAC. We also provide the average number of scans in each local map. The time required to generate each local map is not reported,

as it directly corresponds to the LiDAR odometry algorithm used, i.e., KISS-ICP.

Additionally, in Tab. 10, we compare the average runtimes of all baselines in terms of frames processed per second (FPS) across three sequences with dense, 360° FoV LiDARs. Scan Context achieves the highest processing speed on all sequences, followed by SOLiD and our method. Notably, the FPS values for STD, BTC, and our method include the runtime of KISS-ICP for scan registration, which generally makes them slower than scan-based methods such as SOLiD and Scan Context, which do not require registration. The learning-based methods, LoGG3D and BEVPlace++, are the slowest, often running below the typical LiDAR frame rate of 10 FPS.

Overall, our approach achieves a favorable balance between speed and effectiveness, outperforming most conventional baselines while remaining significantly faster than other local map-based and learning-based methods, consistently maintaining processing speeds above the sensor frame rate.

In summary, the experiments presented above support all our claims and showcase the applicability of our approach across various scenarios. The first experiment shows the

Table 9. Runtime evaluation of our approach. We report the mean and standard deviation of the execution times for each key component of our pipeline.

Sequence	Avg. Scans per Local Map	Execution Time (mean \pm standard deviation) [ms]		
		Ground Alignment	Loop Closure Detection	Loop Closure Validation
MulRan Sejong01	119	4.27 \pm 1.59	17.74 \pm 10.85	0.31 \pm 0.11
HeLiPR Bridge01	96	7.44 \pm 4.02	32.11 \pm 21.21	0.34 \pm 0.20
NCLT 2012-01-08	664	14.94 \pm 7.21	63.79 \pm 50.49	0.38 \pm 0.31
IPB-Car	137	10.38 \pm 6.04	50.96 \pm 28.40	0.35 \pm 0.33

Table 10. Average runtime evaluation of all the baselines. We report the average number of frames processed per second (FPS) for each baseline on three sequences with dense, 360° FoV LiDARs.

Sequence	MulRan KAIST01	HeLiPR Bridge01	NCLT 2012-01-08
LiDAR	Ouster OS1-64	Ouster OS2-128	Velodyne HDL-32E
STD	24	9	13
BTC	13	8	17
Scan Context	144	101	111
SOLiD	35	24	40
LoGG3D	12	7	15
BEVPlace++	7	5	5
Ours	27	15	29

robustness of our approach to various scanning patterns, FoVs, and resolutions of the LiDAR sensor. The second experiment illustrates the applicability of our approach to detect loop closures between multiple sessions with short-term and long-term revisits through the same environment. The two subsequent experimental evaluations support the newly proposed modules in our loop closure pipeline. We show that our approach can be used on LiDAR platforms with non-planar motion and is also robust towards perceptual aliasing. The fifth experiment confirms that our pipeline produces accurate and complete 3D pose estimates for aligning loop closures. We further demonstrate that our approach can align multiple trajectories with minimal physical scene overlap. Finally, we show that our pipeline can efficiently operate above sensor frame rates, even on dense 3D LiDAR sensors.

6 Limitations

Our approach demonstrates robustness across multiple datasets, but it has certain limitations. The pipeline relies on locally consistent LiDAR odometry to generate local maps, which constitutes its most critical limitation. When the scan matching algorithm produces degenerate maps, loop closure performance also degrades. Incorporating more robust odometry that fuses inertial or visual data with LiDAR scans can help alleviate this limitation.

The method also depends on consistent ground plane detection for BEV projection, which may not hold in cluttered indoor, forested, or off-road scenarios. In such environments, approaches tailored to specific settings, such as ForestLPR (Shen et al. 2025), perform better. For aerial data, the density-preserving BEV projection can undersample vertical structures; a maximum-height BEV may provide a stronger alternative.

Although our method consistently performs better than existing baselines in intra-session and inter-session scenarios, inter-LiDAR scenarios remain challenging. Opposite-direction revisits with low-FoV LiDAR sensors, such as

Livox Avia and Aeva Aeries II, are particularly difficult due to limited scene overlap, even after aggregation into local maps.

In summary, our method performs robustly across diverse urban datasets but is constrained by odometry quality, assumptions of a detectable ground plane, and reduced recall in the inter-LiDAR loop closure detection. Addressing these limitations will broaden its applicability to more diverse environments.

7 Conclusion

In this paper, we present a novel and robust approach to detect loop closures for LiDAR-based SLAM and provide 3D alignments between the detected closures. Our method leverages a density-preserving bird’s-eye-view projection of local maps generated from local odometry estimates. We use the local ground plane as a common reference plane across revisits and align the local maps such that the ground plane coincides with the xy-plane of the local map’s reference frame. This ensures a consistent BEV representation across diverse mobile platforms with varying LiDAR motion profiles. We extract ORB feature descriptors on these BEV projections and apply a self-similarity pruning strategy to reduce incorrect closures in repetitive environments.

We implement and extensively evaluate our approach on several public and self-recorded datasets featuring different LiDAR sensors mounted on various mobile platforms. We compare our method with state-of-the-art baselines and demonstrate its effectiveness in detecting loop closures in diverse urban settings across LiDAR sensors with different scanning patterns and fields of view. Furthermore, we illustrate the capability of our approach to detect long-term loop closures across multiple sessions. Finally, our runtime evaluations indicate the real-time feasibility of our approach.

To facilitate further research on place recognition and loop closure detection in LiDAR-SLAM, we release our software as open-source.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

This work has partially been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy, EXC-2070 – 390732324 – PhenoRob, by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under STA 1051/5-1 within the FOR 5351 (AID4Crops), by the European Union's Horizon Europe research and innovation programme under grant agreement No 101070405 (DigiForest), and by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG).

References

- Arandjelovic R, Gronat P, Torii A, Pajdla T and Sivic J (2016) NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Bailey T and Durrant-Whyte H (2006) Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine* 13(3): 108–117.
- Blanco JL, Jiménez JG and Fernández-Madrigal JA (2013) A Robust, Multi-hypothesis Approach to Matching Occupancy Grid Maps. *Robotica* 31: 687–701.
- Bosse M, Newman P, Leonard J and Teller S (2004) Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework. *Intl. Journal of Robotics Research (IJRR)* 23(12): 1113–1139.
- Bosse M and Zlot R (2013) Place Recognition Using Keypoint Voting in Large 3D Lidar Datasets. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Carlevaris-Bianco N, Ushani A and Eustice R (2016) University of Michigan North Campus Long-term Vision and LiDAR Dataset. *Intl. Journal of Robotics Research (IJRR)* 35(9): 1023–1035.
- Chen X, Läbe T, Milioto A, Röhling T, Vysotska O, Haag A, Behley J and Stachniss C (2020) OverlapNet: Loop Closing for LiDAR-based SLAM. In: *Proc. of Robotics: Science and Systems (RSS)*.
- Cummins M and Newman P (2008) FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *Intl. Journal of Robotics Research (IJRR)* 27(6): 647–665.
- Dellenbach P, Deschaud J, Jacquet B and Goulette F (2022) CT-ICP Real-Time Elastic LiDAR Odometry with Loop Closure. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Dubé R, Cramariuc A, Dugas D, Nieto J, Siegwart R and Cadena C (2018) SegMap: 3D Segment Mapping Using Data-Driven Descriptors. In: *Proc. of Robotics: Science and Systems (RSS)*.
- Ferrari S, Giammarino LD, Brizi L and Grisetti G (2024) MAD-ICP: It Is All About Matching Data—Robust and Informed LiDAR Odometry. *IEEE Robotics and Automation Letters (RA-L)* 9(11): 9175–9182.
- Fischler M and Bolles R (1981) Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM* 24(6): 381–395.
- Fontana S, Agamennoni G, Siegwart R and Sorrenti D (2016) Point Clouds Registration with Probabilistic Data Association. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Frome A, Huber D, Kolluri R, Bülow T and Malik J (2004) Recognizing Objects in Range Data Using Regional Point Descriptors. In: *Proc. of the Europ. Conf. on Computer Vision (ECCV)*.
- Galvez-Lopez D and Tardos JD (2012) Bags of Binary Words for Fast Place Recognition in Image Sequences. *IEEE Trans. on Robotics (TRO)* 28(5): 1188–1197.
- Geiger A, Lenz P, Stiller C and Urtasun R (2013) Vision meets Robotics: The KITTI Dataset. *Intl. Journal of Robotics Research (IJRR)* 32(11): 1231–1237.
- Guadagnino T, Chen X, Sodano M, Behley J, Grisetti G and Stachniss C (2022) Fast Sparse LiDAR Odometry Using Self-Supervised Feature Selection on Intensity Images. *IEEE Robotics and Automation Letters (RA-L)* 7(3): 7597–7604.
- Gupta S, Guadagnino T, Mersch B, Vizzo I and Stachniss C (2024) Effectively Detecting Loop Closures using Point Cloud Density Maps. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Hu X, Rodríguez FSA and Gepperth A (2014) A Multi-Modal System for Road Detection and Segmentation. In: *Proc. of the IEEE Intelligent Vehicles Symposium*.
- Jiang B and Shen S (2023) Contour Context Abstract Structural Distribution for 3D LiDAR Loop Detection and Metric Pose Estimation. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Johnson A and Hebert M (1999) Using Spin Images for Efficient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* 21(5): 433–449.
- Jung M, Yang W, Lee D, Gil H, Kim G and Kim A (2024) HeLiPR: Heterogeneous LiDAR Dataset for inter-LiDAR Place Recognition under Spatiotemporal Variations. *Intl. Journal of Robotics Research (IJRR)* 43(12): 1867–1883.
- Kabsch W (1976) A Solution for the Best Rotation to Relate Two Sets of Vectors. *Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography* 32(5): 922–923.
- Kim G, Choi S and Kim A (2021) Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments. *IEEE Trans. on Robotics (TRO)* 38(2): 21–27.
- Kim G and Kim A (2018) Scan Context: Egocentric Spatial Descriptor for Place Recognition within 3D Point Cloud Map. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Kim G, Park B and Kim A (2019) 1-day learning, 1-year localization: Long-term LiDAR Localization Using Scan Context Image. *IEEE Robotics and Automation Letters (RA-L)* 4(2): 1948–1955.
- Kim G, Park Y, Cho Y, Jeong J and Kim A (2020) Mulran: Multimodal Range Dataset for Urban Place Recognition. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Kim H, Choi J, Sim T, Kim G and Cho Y (2024) Narrowing Your FOV With SOLiD: Spatially Organized and Lightweight Global Descriptor for FOV-Constrained LiDAR Place Recognition. *IEEE Robotics and Automation Letters (RA-L)* 9(11):

- 9645–9652.
- Komorowski J (2021) MinkLoc3D: Point Cloud Based Large-Scale Place Recognition. In: *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*.
- Kümmerle R, Grisetti G, Strasdat H, Konolige K and Burgard W (2011) g2o: A General Framework for Graph Optimization. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Lee S, Lim H and Myung H (2022) Patchwork++: Fast and Robust Ground Segmentation Solving Partial Under-segmentation Using 3D Point Cloud. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Li Y and Li H (2021) LiDAR-Based Initial Global Localization Using Two-Dimensional (2D) Submap Projection Image (SPI). In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Lim H, Kim D, Shin G, Shi J, Vizzo I, Myung H, Park J and Carlone L (2025) KISS-Matcher: Fast and Robust Point Cloud Registration Revisited. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Lim H, Minh O and Myung H (2021) Patchwork: Concentric Zone-based Region-wise Ground Segmentation with Ground Likelihood Estimation Using a 3D LiDAR Sensor. *IEEE Robotics and Automation Letters (RA-L)* 6(4): 6458–6465.
- Lowe D (2004) Distinctive Image Features from Scale-Invariant Keypoints. *Intl. Journal of Computer Vision (IJCV)* 60(2): 91–110.
- Lowry S, Sunderhauf N, Newman P, Leonard J, Cox D, Corke P and Milford M (2016) Visual Place Recognition: A Survey. *IEEE Trans. on Robotics (TRO)* 32(1): 1–19.
- Lu S, Xu X, Yin H, Chen Z, Xiong R and Wang Y (2022) One RING to Rule Them All: Radon Sinogram for Place Recognition, Orientation and Translation Estimation. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Luo L, Cao S, Han B, Shen H and Li J (2021) BVMatch: Lidar-Based Place Recognition Using Bird’s-Eye View Images. *IEEE Robotics and Automation Letters (RA-L)* 6(3): 6076–6083.
- Luo L, Cao S, Sheng Z and Shen H (2022) LiDAR-Based Global Localization Using Histogram of Orientations of Principal Normals. *IEEE Trans. on Intelligent Vehicles (TIV)* 7(3): 771–782.
- Luo L, Cao SY, Li X, Xu J, Ai R, Yu Z and Chen X (2025) BEVPlace++: Fast, Robust, and Lightweight LiDAR Global Localization for Unmanned Ground Vehicles. *IEEE Trans. on Robotics (TRO)* 41: 4479–4498.
- Luo L, Zheng S, Li Y, Fan Y, Yu B, Cao S, Li J and Shen H (2023) BEVPlace: Learning LiDAR-based Place Recognition using Bird’s Eye View Images. In: *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*.
- Ma J, Xiong G, Xu J and Chen X (2024) CVTNet: A Cross-View Transformer Network for LiDAR-Based Place Recognition in Autonomous Driving Environments. *IEEE Trans. on Industrial Informatics (TII)* 20(3): 4039–4048.
- Ma J, Zhang J, Xu J, Ai R, Gu W and Chen X (2022) OverlapTransformer: An Efficient and Yaw-Angle-Invariant Transformer Network for LiDAR-Based Place Recognition. *IEEE Robotics and Automation Letters (RA-L)* 7(3): 6958–6965.
- Magnusson M, Andreasson H, Nuechter A and Lilienthal AJ (2009) Appearance-Based Loop Detection from 3D Laser Data Using the Normal Distributions Transform. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Mendes E, Koch P and Lacroix S (2016) ICP-based Pose-Graph SLAM. In: *Proc. of the IEEE Intl. Symp. on Safety, Security, and Rescue Robotics (SSRR)*.
- Mur-Artal R, Montiel J and Tardos JD (2015) ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. on Robotics (TRO)* 31(5): 1147–1163.
- Paigwar A, Erkent Ö, Sierra-Gonzalez D and Laugier C (2020) GndNet: Fast Ground Plane Estimation and Point Cloud Segmentation for Autonomous Vehicles. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Qi C, Yi K, Su H and Guibas LJ (2017a) PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. In: *Proc. of the Conf. on Neural Information Processing Systems (NeurIPS)*.
- Qi CR, Su H, Mo K and Guibas LJ (2017b) PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Ramezani M, Wang L, Knights J, Li Z, Pounds P and Moghadam P (2024) Pose-Graph Attentional Graph Neural Network for Lidar Place Recognition. *IEEE Robotics and Automation Letters (RA-L)* 9(2): 1182–1189.
- Ramos F, Nieto J and Durrant-Whyte H (2007) Recognising and Modelling Landmarks to Close Loops in Outdoor SLAM. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Röhling T, Mack J and Schulz D (2015) A Fast Histogram-Based Similarity Measure for Detecting Loop Closures in 3-D LIDAR Data. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Rottmann N, Bruder R, Schweikard A and Rueckert E (2019) Loop Closure Detection in Closed Environments. In: *Proc. of the Europ. Conf. on Mobile Robotics (ECMR)*.
- Rublee E, Rabaud V, Konolige K and Bradski G (2011) ORB: An Efficient Alternative to SIFT or SURF. In: *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*.
- Rusu R, Blodow N and Beetz M (2009) Fast Point Feature Histograms (FPFH) for 3D Registration. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Rusu R, Blodow N, Marton Z and Beetz M (2008) Aligning Point Cloud Views Using Persistent Feature Histograms. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Salti S, Tombari F and Stefano L (2014) SHOT: Unique Signatures of Histograms for Surface and Texture Description. *Journal of Computer Vision and Image Understanding (CVIU)* 125: 251–264.
- Schlegel D and Grisetti G (2018) HBST: A Hamming Distance Embedding Binary Search Tree for Visual Place Recognition. *IEEE Robotics and Automation Letters (RA-L)* 3(4): 3741–3748.
- Shan T, Englot B, Duarte F, Ratti C and Rus D (2021) Robust Place Recognition using an Imaging Lidar. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Shan T, Englot B, Meyers D, Wang W, Ratti C and Rus D (2020) LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.

- Shen Y, Tuna T, Hutter M, Cadena C and Zheng N (2025) ForestLPR: LiDAR Place Recognition in Forests Attentioning Multiple BEV Density Images. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Steder B, Grisetti G and Burgard W (2010) Robust Place Recognition for 3D Range Data Based on Point Features. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Steder B, Ruhnke M, Grzonka S and Burgard W (2011) Place Recognition in 3D Scans Using a Combination of Bag of Words and Point Feature Based Relative Pose Estimation. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Tombari F, Salti S and Stefano LD (2010) Unique Shape Context for 3D Data Description. In: *Proc. of the ACM Workshop on 3D Object Retrieval*.
- Umeyama S (1991) Least-squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)* 13(4): 376–380.
- Uy A and Lee G (2018) PointNetVLAD: Deep Point Cloud Based Retrieval for Large-scale Place Recognition. In: *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.
- Vidanapathirana K, Ramezani M, Moghadam P, Sridharan S and Fookes C (2022) LoGG3D-Net: Locally Guided Global Descriptor Learning for 3D Place Recognition. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Vizzo I, Guadagnino T, Mersch B, Wiesmann L, Behley J and Stachniss C (2023) KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *IEEE Robotics and Automation Letters (RA-L)* 8(2): 1029–1036.
- Vysotska O and Stachniss C (2016) Lazy Data Association For Image Sequences Matching Under Substantial Appearance Changes. *IEEE Robotics and Automation Letters (RA-L)* 1(1): 213–220.
- Vysotska O and Stachniss C (2019) Effective Visual Place Recognition Using Multi-Sequence Maps. *IEEE Robotics and Automation Letters (RA-L)* 4(2): 1730–1736.
- Wang Y, Sun Z, Xu C, Sarma S, Yang J and Kong H (2020) LiDAR Iris for Loop-Closure Detection. In: *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*.
- Wiesmann L, Marks E, Gupta S, Guadagnino T, Behley J and Stachniss C (2024) Efficient LiDAR Bundle Adjustment for Multi-Scan Alignment Utilizing Continuous-Time Trajectories. *arXiv preprint arXiv:2412.11760*.
- Xu J, Zhang R, Dou J, Zhu Y, Sun J and Pu S (2021a) RPNNet: A Deep and Efficient Range-Point-Voxel Fusion Network for LiDAR Point Cloud Segmentation. In: *Proc. of the IEEE/CVF Intl. Conf. on Computer Vision (ICCV)*.
- Xu X, Lu S, Wu J, Lu H, Zhu Q, Liao Y, Xiong R and Wang Y (2023) RING++: Roto-Translation Invariant Gram for Global Localization on a Sparse Scan Map. *IEEE Trans. on Robotics (TRO)* 39(6): 4616–4635.
- Xu X, Yin H, Chen Z, Li Y, Wang Y and Xiong R (2021b) DiSCO: Differentiable Scan Context With Orientation. *IEEE Robotics and Automation Letters (RA-L)* 6(2): 2791–2798.
- Yang J, Zhang Q, Xiao Y and Cao Z (2017) TOLDI: An Effective and Robust Approach for 3D Local Shape Description. *Pattern Recognition* 65: 175–187.
- Yin H, Xu X, Lu S, Chen X, Xiong R, Shen S, Stachniss C and Wang Y (2024) A Survey on Global LiDAR Localization: Challenges, Advances and Open Problems. *Intl. Journal of Computer Vision (IJCV)* 132: 1–33.
- Yuan C, Lin J, Liu Z, Wei H, Hong X and Zhang F (2024) BTC: A Binary and Triangle Combined Descriptor for 3-D Place Recognition. *IEEE Trans. on Robotics (TRO)* 40: 1580–1599.
- Yuan C, Lin J, Zou Z, Hong X and Zhang F (2023) STD: Stable Triangle Descriptor for 3D Place Recognition. In: *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*.
- Zhang Y, Shi P and Li J (2024) LiDAR-Based Place Recognition For Autonomous Driving: A Survey. *ACM Computing Surveys* 57(4).
- Zhou B, He Y, Qian K, Ma X and Li X (2021) S4-slam: A real-time 3d lidar slam system for ground/watersurface multi-scene outdoor applications. *Autonomous Robots* 45(1): 77–98.
- Zhou Q, Park J and Koltun V (2018) Open3D: A Modern Library for 3D Data Processing. *arXiv preprint arXiv:1801.09847*.