Effectively Detecting Loop Closures using Point Cloud Density Maps

Saurabh Gupta Tiziano Guadagnino Benedikt Mersch Ignacio Vizzo Cyrill Stachniss

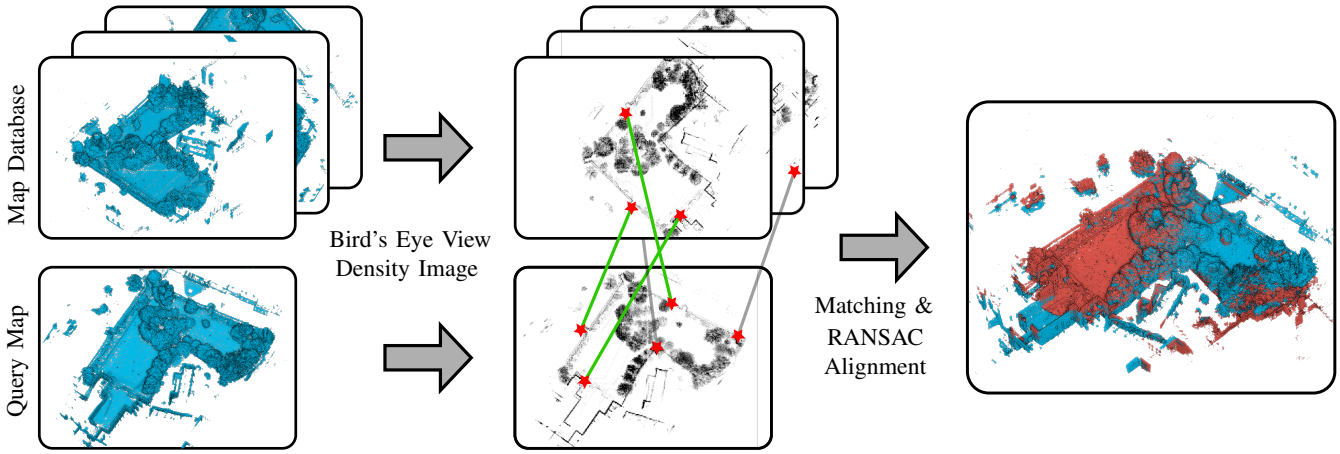


Fig. 1: A database over previously seen local maps is queried for matching a new local map (left). We perform feature matching on features computed on the bird's eye view density images of these maps to detect closures (center). Finally, a RANSAC-based 2D rigid body alignment is used to obtain the 2D pose estimate between loop closed local maps (right).

***Abstract*— The ability to detect loop closures plays an essential role in any SLAM system. Loop closures allow correcting the drifting pose estimates from a sensor odometry pipeline. In this paper, we address the problem of effectively detecting loop closures in LiDAR SLAM systems in various environments with longer lengths of sequences and agnostic of the scanning pattern of the sensor. While many approaches for loop closures using 3D LiDAR sensors rely on individual scans, we propose the usage of local maps generated from locally consistent odometry estimates. Several recent approaches compute the maximum elevation map on a bird's eye view projection of point clouds to compute feature descriptors. In contrast, we use a density image bird's eye view representation, which is robust to viewpoint changes. The utilization of dense local maps allows us to reduce the complexity of features describing these maps, as well as the size of the database required to store these features over a long sequence. This yields a real-time application of our approach for a typical robotic 3D LiDAR sensor. We perform extensive experiments to evaluate our approach against other state-of-the-art approaches and show the benefits of our proposed approach.**

I. Introduction

Autonomous mobile robots must accurately estimate their ego motion while traversing through previously unseen environments. This task forms a core element of a simultaneous localization and mapping (SLAM) system [6], [28]. Sequential odometry estimates often suffer from drift, especially over long traversals. This drift can occur due to the inherent noise in the robot motion and sensor data, dynamics in the environment, and non-trivial data association problems. When a robot revisits a known place, its belief may indicate a completely different location because of such drift. To overcome this problem, detecting revisited locations, also known as loop closures, is paramount for any SLAM pipeline. Robust loop closure detection is a prerequisite for drift correction through the integration of pose constraints often done in the underlying pose-graph [6].

Detecting previously seen places from sensor data [22], [35], [36] requires crafting an as-unique-as-possible description of the sensed environment. At the same time, this description should be invariant to changes in the viewpoint, scan pattern of LiDARs, and dynamics in the sensed environment. Furthermore, to incorporate loop closures in a SLAM system, this description should preferably capture the geometry of the scene to enable relative pose estimation.

Traditional methods either work directly on the maps [7], [31], [37] or rely on computing a description for each LiDAR scan [5], [8], [10], [15], [17]. Subsequently, they use a matching algorithm to compare these descriptions and detect loop closures. The bird's eye view (BEV) projection of point clouds is one popular approach [15], [17] towards detecting loop closures, providing a compact 2D representation for faster feature detection and matching.

The main contribution of this paper is a simple yet

All authors are with the Center for Robotics, University of Bonn, Germany. Ignacio Vizzo is with Dexory, UK; during the work on the paper, he was with the University of Bonn. Cyrill Stachniss is additionallywith the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

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 European Union's Horizon 2020 research and innovation program under grant agreement No 101017008 (Harmony) and the Horizon Europe research and innovation program under grant agreement No 101070405 (DigiForest).

effective approach, see Fig. 1, for detecting loop closures in online LiDAR SLAM systems. Local maps generated using locally consistent sensor odometry estimates form the core element of our approach. We show that a simple feature detection and matching pipeline exploiting density images from the BEV projections of the local maps can effectively detect loop closures. We also provide a validation step to perform a geometric verification of detected loop closures and provide a complete 2D rigid body transform between the loop-closed local maps. Our approach (i) can effectively detect loop closures between two temporally separated local maps in a variety of environments, (ii) provides a 2D rigid body transform between the detected loop closures, and (iii) is agnostic of the sensor scan pattern by exploiting local maps. The open-source implementation of our approach is available at: `https://github.com/PRBonn/MapClosures.git`.

II. Related Work

The task of loop closing in the domain of 3D ranging sensors has been discussed in many recent works, as laid out thoroughly by Zhang et al. [40]. Many approaches [1], [10], [25], [26], [38] propose 3D point features inspired by image-based localization. They discretize the neighborhood around a point into a geometrical grid and compute the local descriptor based on the height, density, distance, or angle of the points within. These methods are partially affected by the sparsity of typical LiDAR point clouds and are not viewpoint-invariant.

Learning-based approaches such as SegMap [4] segment point clouds using a region-growing algorithm and compute data-driven descriptors of these segments to detect loop closures. OverlapNet [3] computes overlap between the range-image representation of two scans using a deep neural network. Such approaches, though, require a training step along with GPU acceleration.

Several approaches project the 3D point cloud into a 2D spherical-view range image [29], [30] or a BEV image [17], [18], [19], and then use established techniques in image processing to compute descriptors. Some methods [8], [13], [15], [21] use BEV projections to define a global descriptor capturing holistic information from each scan.

The popular ScanContext [15] proposes a polar elevation map centered at the local reference frame of the LiDAR scan. This allows the computation of a rotation-invariant global descriptor but loses translational invariance. They try to overcome this by manually shifting the reference frame of scans by the expected translation during revisits. ScanContext++ [13] computes Cartesian elevation maps along with polar elevation maps. However, most of these global descriptors assume a spherical scan pattern of conventional LiDARs, and, as such, they are not well suited for other scan patterns, such as the Livox LiDAR.

ContourContext [9] generates BEV projections at predefined heights and clusters (contours) the projected points within these elevation maps. They use the statistics computed from these contours to define a global descriptor for matching. Another recent work by Yuan et al. [39] proposes stable triangle descriptors (STD) computed using keypoints within a local neighborhood. These keypoints too, are based on a local BEV projection of the points within a voxel at the boundary of a planar region. They accumulate a fixed number of consecutive scans for computing these descriptors to have a sufficient point cloud density. However, these methods use the elevation map as the 2D representation, which is sensitive to the sensor viewpoint. BVMatch [18] proposes using a density map representation but requires training a bag-of-words model on features detected from individual scans to detect loop closures. Thus, it is not ideal for online SLAM in novel environments.

Among the 2D projection based approaches, BEV projections [13], [15], [17], [18], [19] seem to be preferred over the range image based approaches [3], [20]. A key reason is that the range images lose depth information while projecting spherically. On the other hand, BEV projections preserve 2D geometry along the local XY plane, which is crucial for most autonomous ground robots/vehicles.

Our proposed method uses local maps as the primary representation, where we accumulate scans based on travel distance criteria rather than naively accumulating a fixed number of consecutive scans. Such maps allow us the benefit of having a dense collection of points representing the environment, which show a stronger invariance towards viewpoint changes and sensor scan patterns. We use 2D density images [18] of these local maps instead of elevation images to increase robustness against viewpoint changes and noise in the sensing process. Utilizing existing feature descriptors from the computer vision community [24], we compute local features on these density maps and store them in an efficient binary search tree [27] for matching with query maps. This allows us to provide a complete 2D relative pose estimate between detected map-level closures, which aids the final 3D registration step in any geometric SLAM pipeline.

III. Our approach to Loop Closure Detection

Our proposed approach detects loop closures at a local map level and provides a complete 2D initial guess on the relative pose between matched maps. We generate local maps using odometry estimates and represent them as a density image as explained in Sec. III-A and Sec. III-B. Then, we compute local features on these images and create a database for matching with query maps, see Sec. III-C. Finally, in Sec. III-D, we show the geometrical verification and alignment of the loop closures. Fig. 1 shows the overall pipeline.

A. Local Maps

Our approach for detecting loop closures in SLAM uses local maps. We build a local map by accumulating consecutive scans with their relative pose estimates obtained from an ICP odometry pipeline. In particular, we use the KISS-ICP [34] odometry pipeline, a state-of-the-art approach for LiDAR odometry with strong open-source community support.

To generate these local maps, we consider a set of n consecutive 3D point clouds $\{\mathsf{P}_i, \ldots, \mathsf{P}_{i+n-1}\}$ in their local reference frame, and transform them into the reference frame of the i^{th} scan $\{{}^i\mathsf{P}_i, \ldots, {}^i\mathsf{P}_{i+n-1}\}$ as follows:

$$ {}^i\mathsf{P}_{i+n-1} = \mathsf{T}_i^{-1}\mathsf{T}_{i+n-1}\mathsf{P}_{i+n-1}, \tag{1} $$

where $\mathsf{T}_i \in \mathbb{SE}(3)$ is the odometry estimate of the pose connected to the i^{th} scan. We aggregate all these scans into a local map $\mathcal{M}_i$ using a voxel grid with a resolution of ν_{map} meters per voxel. The local map $\mathcal{M}_i$ is centered at the frame T_i, the first scan of the map. To ensure a uniform density of points in the voxel grid cells, we retain a maximum of 20 points per cell. The size of this map $\mathcal{M}_i$ is decided based on the distance traveled by the sensing platform. We accumulate n consecutive scans until the travel distance $\|\boldsymbol{t}_{i+n-1} - \boldsymbol{t}_i\|_2$ exceeds a threshold of τ_m meters, where $\boldsymbol{t}_i \in \mathbb{R}^3$ is the translational component of the odometry pose estimate T_i.

B. Density Images

The BEV projection of point clouds is a widely used approach in place recognition using LiDARs [13], [15], [17], [18], [19]. Such a projection preserves local 2D geometry and, at the same time, also reduces the computational complexity of algorithms due to a reduced dimension to process. However, many traditional BEV projection approaches [13], [15], [17], [19] store the maximum height (elevation map) of the points in each bin, not point densities. The elevation map is sensitive to the orientation of the sensor as the maximum height recorded varies with the distance between the scanner and the object. The density of points scanned on a surface, on the other hand, is less sensitive to viewpoint changes.

For a local map $\mathcal{M}_i$, we project all the points to the local ground plane through an orthographic BEV projection. When the reference frame T_i of the local map $\mathcal{M}_i$ has its xy-plane parallel to the ground, as is the case for ground robots/vehicles, this would correspond to simply dropping the z-coordinate of all the individual points. For other robots, such as UAVs, we require a gravity vector that one can directly obtain from an IMU. The BEV projection gives us a set of points B_i in $\mathbb{R}^2$ bounded by a rectangular window from $[x_i^l, y_i^l]^\mathsf{T}$ to $[x_i^u, y_i^u]^\mathsf{T}$ meters where,

$$ \begin{bmatrix} x_i^u \\ y_i^u \end{bmatrix} = \max_{x,y} \mathsf{B}_i \; ; \; \begin{bmatrix} x_i^l \\ y_i^l \end{bmatrix} = \min_{x,y} \mathsf{B}_i. \tag{2} $$

Furthermore, we discretize B_i into a 2D Cartesian grid $\mathsf{N}_i(u,v) \in \mathbb{N}_0^{W_i \times H_i}$ of resolution ν_{res} meters per cell where,

$$ W_i = \left\lceil \frac{x_i^u - x_i^l}{\nu_{\text{res}}} \right\rceil \; ; \; H_i = \left\lceil \frac{y_i^u - y_i^l}{\nu_{\text{res}}} \right\rceil. \tag{3} $$

Each cell in this grid $N(u,v)$ stores the number of points contained in that cell after discretization. The grayscale density image $\mathsf{I}_i(u,v)$ of the local map $\mathcal{M}_i$ is then defined as,

$$ \mathsf{I}_i(u,v) = \frac{N(u,v) - N_{\text{min}}}{N_{\text{max}} - N_{\text{min}}} \in \mathbb{R}^{W_i \times H_i}, \tag{4} $$

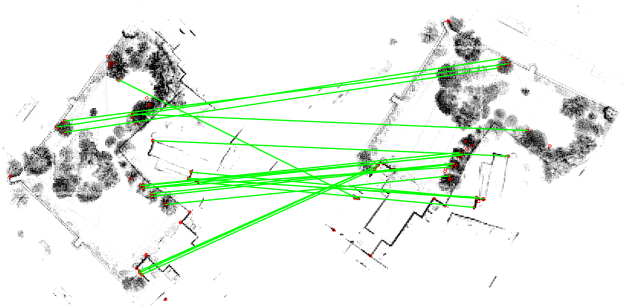


Fig. 2: ORB features (red) and matches (green) between two density images from the same location, revisited from a different viewpoint.

$$ N_{\text{max}} = \max_{u,v} N_i(u,v) \; ; \; N_{\text{min}} = \min_{u,v} N_i(u,v). \tag{5} $$

We set all the image pixels (u,v) with density lower than $5\,\%$ of the maximum value to be zero. This helps to reduce the noise in the local maps and to remove the ground plane, which is typically insignificant for feature detection. These density images capture 2D floorplan-like information from the local maps, like building facades and trees, as seen in the density maps in Fig. 2. These rigid structures provide reliable features to track for place recognition.

C. Feature Detection and Database Creation

Facilitated by accumulating information in the form of the local map density images I_j, we utilize the well-known ORB [24] feature descriptors to capture corner-like features from them. To speed up computation, we compute ORB features without scale-invariance. We can do this since the density image is an orthographic projection of the 3D world and has no scale ambiguity as regular camera images. Fig. 2 shows the ORB features computed on sample density images in red.

Furthermore, the binary domain of ORB descriptors allows for efficient feature storage and matching by leveraging the Hamming distance metric. In particular, we employ the Hamming distance embedding binary search tree (HBST) [27] to store the set of feature descriptors D_i obtained from each density image $\mathsf{I}_i(u,v)$ along with the corresponding map index i. The depth of the HBST is bounded by the number of bits in the binary descriptor (256). It implies that a query descriptor would have a maximum of 256 bitwise comparisons with the tree's nodes before it terminates in one of the leaf nodes. Furthermore, the capacity of each leaf node is limited to a maximum of 100 descriptors per leaf node. Fixing these two design parameters imposes an upper bound on the computational time of the feature-matching process while not introducing any restriction in the practical use of our approach. Although the binary tree is not a complete data structure for a nearest-neighbor computation, it provides a good initial guess on the feature matches for subsequent geometric verification and refinement.

After obtaining a new set of descriptors D_q from the query local map's density image I_q, we find the nearest match for each descriptor in D_q from the binary tree database. We use a threshold of 50 bits on the Hamming distance between two

ORB descriptors to call them a match. Since the binary tree stores descriptors along with an index i of the corresponding map they were obtained from, we cast a vote over the map indices for each such match. After processing all the query descriptors, we select the reference maps corresponding to top-N votes from this voting scheme. We set N to equal half the number of local maps in the database at any time. This dynamic factor allows us to find multiple potential loop closures at the same physical location, the chances of which increase with the number of local maps in the database. As a result, we obtain a list of feature matches between the query map and the reference maps in the database, on which we perform a geometrical verification. Fig. 2 visualizes the corresponding set of feature matches between two density images.

D. Loop Detection and Map Alignment

The geometrical verification step involves a 2D alignment of the matched features. It implies computing a 2D rigid body transformation (3 degrees of freedom) that best aligns the matched features from the binary tree based on a distance metric. This is similar to the image-alignment problem but limited to an $\mathbb{SE}(2)$ transform instead of a homography. We use a Random Sampling Consensus (RANSAC)-based alignment strategy to reject outlier associations due to the incompleteness of the binary tree-based matching. This verification step is performed for matches between the query map and each of the top-N reference maps separately.

We only need two sets of matching keypoints to compute a rigid body alignment in 2D. Using this fact, we design a RANSAC scheme, randomly drawing 2 feature matches from the entire set of feature matches between a query (I_q) and a reference (I_p) density image. We compute the relative 2D alignment between them using the Kabsch-Umeyama [12], [32] algorithm. It provides us with a rotation matrix $\boldsymbol{R} \in \mathbb{SO}(2)$ and a translation vector $\boldsymbol{t} \in \mathbb{R}^2$, which can be composed into a homogenous transformation ${}^{q}\mathsf{T}_{p} \in \mathbb{SE}(2)$. Using this 2D alignment, we transform all the features from I_p to the reference frame of I_q.

We compute the point-wise error between the matching features for verification within RANSAC as the Euclidean distance between the matched point sets. Matches with distances larger than $1.5\,\mathrm{m}$ ($3\,\mathrm{pixels}$ for $\nu_{\mathrm{res}} = 0.5\,\mathrm{m}$) are considered as outliers. The RANSAC scheme terminates after a fixed number of iterations or if we find a high-quality solution, i.e., in case more than 30 inliers are obtained.

We define a threshold γ for the minimum number of inliers required from the RANSAC alignment to conclude whether two local maps belong to the same location. The associated 2D transform between the two density images is a reasonable initial estimate for the complete 3D transform between local maps for ground robots/vehicles. Note that we scale the translation vector by the voxel size ($\nu_{\mathrm{res}}\boldsymbol{t}$) to undo the effect of the discretization while generating the density images. We can later use this initial alignment as an initial guess for a fine point cloud registration in 3D.

IV. Experimental Evaluation

This work provides a pipeline to compute loop closures for SLAM using local maps. We present our experiments to show the capabilities of our method. The results of our experiments also support our key claims, which are: (i) we can effectively detect loop closures between two temporally separated local maps in a variety of environments, (ii) we provide a 2D rigid body transform between detected loop closures, and (iii) our approach is agnostic of the sensor scan pattern by exploiting local maps.

A. Evaluation Criteria

To assess the performance of loop closure detection, we identify scan-wise reference closures based on the volumetric overlap of measured point clouds. The community needs more consensus on what a ground-truth loop closure is, as highlighted by Jiang et al. [9]. Related works [9], [15], [39] usually consider two scans as a true positive closure if the distance between the corresponding reference locations is below a certain threshold. This criterion implies a potentially wrong assumption that a sensor observes the same objects when being at a similar location. It does not hold when other objects occlude the previously seen area, the sensor has a limited field of view, or when the orientation differs strongly.

Instead, we propose to use the volumetric overlap of scans to decide if a loop should be closed. We argue that if the point clouds from two different points in time overlap, we can find their relative pose and integrate it into a pose-graph. The first step of our reference identification is to sample the reference trajectory at equidistant locations ($2\,\mathrm{m}$) to reduce the number of candidates. Next, we accumulate the registered scans between two consecutive key locations for a dense representation of the local environment. We find all possible pairs of key locations within a distance of the sensor's maximum range with a minimum travel distance. Further, we voxelize the accumulated point clouds of both keyframes. We use a voxel size of $0.5\,\mathrm{m}$. Finally, we compute the overlap $o \in [0, 1]$ between two non-empty voxel grids $\mathcal{V}_i$ and $\mathcal{V}_j$ using the overlap coefficient [33] as in Eq. (6), also called the Szymkiewicz-Simpson coefficient:

$$o\left(\mathcal{V}_i, \mathcal{V}_j\right) = \frac{\left|\mathcal{V}_i \cap \mathcal{V}_j\right|}{\min\left(\left|\mathcal{V}_i\right|, \left|\mathcal{V}_j\right|\right)}. \tag{6}$$

We consider two keyframes to be a loop closure if their overlap $o > 0.5$.

To also evaluate and compare predicted closures between local maps, we need to compute scan index closures eventually since other benchmarks provide closures on individual scans. To achieve this, we apply the 2D pose correction ${}^{q}\mathsf{T}_{p}$ between maps to the individual scans comprising these maps, transforming them to a common reference frame. Then, we compare all the pairs of scan-level poses between the two maps, computing the pairwise distance between them. Finally, we use a distance threshold τ_d to classify whether or not two scan indices are predicted to be a loop closure.

TABLE I: Precision (P), Recall (R) and F1 scores of state-of-the-art baselines and our approach for all datasets.

Datasets	MK03			MR02			MS01			NCD			HT01		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
SC [15]	**0.890**	0.450	0.598	0.259	0.071	0.111	0.027	0.126	0.045	0.415	0.007	0.013	N/A	N/A	N/A
CC [9]	0.798	0.424	0.554	0.695	0.080	0.144	0.244	0.076	0.116	0.226	0.030	0.053	N/A	N/A	N/A
STD [39]	0.679	0.435	0.530	0.491	0.042	0.078	–	–	–	–	–	–	–	–	–
Ours	0.730	**0.668**	**0.698**	**0.939**	**0.731**	**0.822**	**0.983**	**0.298**	**0.458**	0.927	**0.068**	**0.127**	**0.473**	**0.669**	**0.554**
Ours (E)	0.759	0.606	0.674	0.639	0.118	0.199	0.978	0.296	0.454	**0.933**	0.043	0.083	–	–	–

B. Experimental Setup

We evaluate our work on the MulRan [14], Newer College [23], and HeLiPR [11] datasets. From the MulRan dataset, we use the KAIST03 (MK03), Riverside02 (MR02), and Sejong01 (MS01) sequences, which contain loop closures from varying viewpoints. They also cover a variety of driving scenarios like city and highway scenes and longer trajectories. The Newer College 01_short_experiment (NCD) sequence provides data recorded from a hand-held scanning platform in contrast to the car-mounted scanner in MulRan. For HeLiPR, we use the Town01 (HT01) sequence with scans from a Livox LiDAR with a small field of view and an irregular scanning pattern.

We compare our approach against three other baselines: ScanContext-10 (SC) [15], ContourContext (CC) [9], and STD [39]. SC is widely used in the context of SLAM and among the current state-of-the-art methods. CC and STD are recently published approaches for loop closures using LiDARs. All these baselines provide a publicly available working code for evaluation. We adapt their source code to provide all possible loop closure candidates for each query scan instead of only the best candidate. It ensures a fair comparison, as our approach also provides multiple closure candidates. The STD pipeline proposes accumulation of 10 consecutive scans into a local map, so we utilize the same method as in our approach, to obtain scan index closures from map index closures, with $\tau_d = 6\,\text{m}$. We set all the other parameters for the baselines to the default values provided in their respective implementations.

For our approach, we set the maximum range of the scanner r as well as the local map truncation distance τ_m to $100\,\text{m}$. For the KISS-ICP odometry, we use the default parameters provided. We use a voxel size of $\nu_{\text{map}} = 1.0\,\text{m}$ for the voxel grid used to generate the local maps and a resolution of $\nu_{\text{res}} = 0.5\,\text{m}$ for the density images. We use the default parameters provided for ORB feature descriptors, only disabling the parameters controlling scale invariance. We require a minimum of 25 feature matches from the HBST within a Hamming distance of $50\,\text{bits}$. All other parameters of the binary tree are set to the defaults. Finally, we predict two local maps to be loop closures based on the number of inliers obtained from the RANSAC-based alignment, the threshold for which is set to be $\gamma = 10$. Finally, to compute scan index loop closures for quantitative evaluation, we use a distance threshold of $\tau_d = 6\,\text{m}$. We maintain these parameters across all datasets except for the HeLiPR Town01 dataset, for which we use a maximum range of $r = 50\,\text{m}$, $\nu_{\text{map}} = 0.5\,\text{m}$, and a minimum of 10 feature matches from the HBST. This is to account for the small field-of-view and the scan pattern of the Livox LiDAR.

C. Performance Evaluation

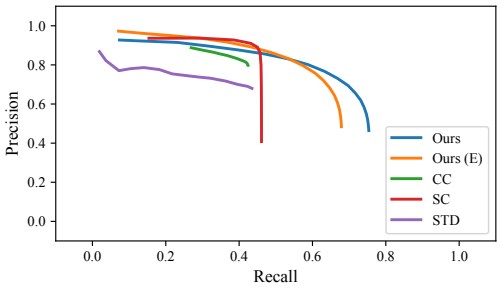
The performance of our approach is shown in Tab. I as a comparison of precision, recall, and F1 scores between each of the baselines and our approach for all the datasets mentioned above. We report the precision and recall scores corresponding to the best achievable F1 scores for each approach, including ours. The F1 score, being the harmonic mean of precision and recall, is a suitable statistic for comparison. It is evident from Tab. I that our approach has the best F1 score among all baselines for all datasets. Furthermore, we achieve these F1 scores at a significantly high precision, which is important for incorporating loop closures into a SLAM pipeline.

One can observe comparable performance across all approaches for the MulRan KAIST03 (MK03) dataset since it is a regular city-like environment about $6.1\,\text{km}$ long. However, we see a significant performance gap between our approach and other baselines for the MulRan Sejong01 (MS01) sequence, which is a $23.4\,\text{km}$ long highway sequence with sparsely distributed features as well as dynamic obstacles. The STD [39] baseline even fails to run over such long sequences having more than 15000 scans.

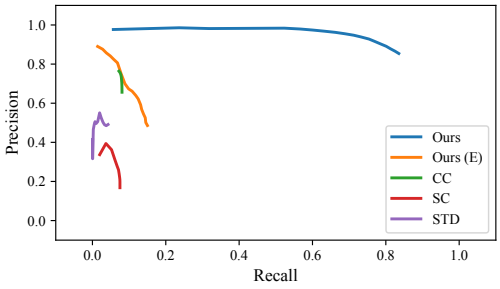
In Fig. 3, we also provide the precision-recall curves for the MulRan KAIST03 and Riverside02 sequences, highlighting the range of performance each approach offers. Our approach not only provides the best F1 score for these datasets as highlighted in Tab. I but also can be adapted to provide either high precision or high recall based on the distance threshold parameter (τ_d).

We also ablate our choice of density images over elevation maps by modifying our pipeline to use elevation images of the local maps. We present the results for the same in the last row (Ours (E)) of Tab. I. We observe that the elevation image-based approach performs at par or worse than our proposed density image-based approach. Also, it fails on the HeLiPR Town01 (HT01) dataset due to the limited field-of-view of the Livox LiDAR.

Overall, this evaluation highlights our approach's ability to detect loop closures in a variety of environments. Furthermore, it is an indirect evaluation of our 2D pose estimates between detected loops, as we use these estimates when computing the scan index level closures.



(a) MulRan KAIST03



(b) MulRan Riverside02

Fig. 3: The precision-recall curves for our approach and other baselines.

D. Livox LiDAR Scan Pattern

The HeLiPR dataset provides scans recorded from a Livox Horizon LiDAR, which has an irregular scanning pattern compared to traditional LiDARs, with a higher density but small field-of-view. This poses a significant challenge in detecting loop closures. However, our approach can overcome these challenges, primarily due to the use of local maps. As seen from the last column (HT01) in Tab. I, we achieve an F1 score of 0.554 on this dataset, also being the only method to work on this dataset. STD [39], even though capable of processing Livox scans due to aggregation of a fixed number of scans, fails on this dataset due to the large size of the HT01 sequence.

E. Offline Optimization with Detected Loop Closures

In this final experiment, we showcase the ability of our approach to perform drift correction in a SLAM pipeline. We perform an offline pose-graph optimization, which includes all the detected map-level loops. We perform a fine ICP registration on the detected loops between local maps with the initial guess provided by our pipeline. The refined 3D pose estimate between the local maps is then incorporated as a constraint in the pose-graph along with the odometry constraints. We use g2o [16] and robust kernels [2].

TABLE II: Absolute Pose Error (APE) in translation and rotation, with (w) and without (w/o) loop closures.

		APE tra. (m) (rms)		APE rot. (rms)	
Datasets	# Closures	w	w/o	w	w/o
MK03	73	3.04	14.64	0.05	0.09
MR02	33	12.70	32.58	0.08	0.11
MS01	2	206.6	1248.6	0.48	0.63
NCD	7	0.61	0.62	0.03	0.03
HT01	2	12.92	18.48	0.14	0.15

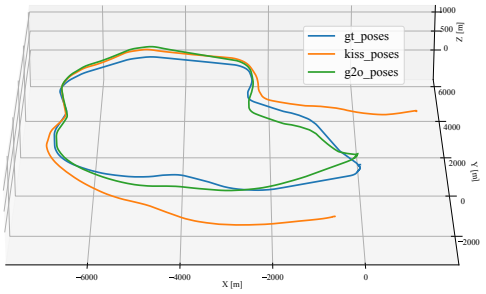


Fig. 4: MulRan Sejong01 sequence poses before and after pose-graph optimization.

In Fig. 4, we show a comparison of pose estimates for the MulRan Sejong01 dataset which is a 23.4 km long highway sequence, with an odometry drift in the order of a few thousand meters. We observe that including our detected loop closures reduces the drift by a significant amount. This is further confirmed by the values of absolute pose error (APE) in translation and rotation with respect to the ground-truth poses, as shown in Tab. II. We provide the APE values with and without the inclusion of loop closures for all the datasets to highlight the significant impact our approach for detecting loop closures can have over the odometry estimation. Furthermore, we achieve this improvement in APE with a significantly lower number of loop closures as seen in Tab. II, allowing for faster optimization of the pose-graph.

In summary, our evaluation suggests that our method can effectively detect loop closures with local maps, with a better or at par performance compared to other approaches on a variety of datasets. The accumulation of scans into local maps makes our method agnostic of the sensor scan pattern and density. Finally, we also demonstrated that our detected loops significantly reduce drift when incorporated into a pose-graph. Thus, we supported all our claims with this experimental evaluation.

V. Conclusion

In this paper, we presented a novel approach to computing loop closures for SLAM. Our approach exploits local maps generated using local odometry estimates and compresses them into a BEV density image representation. This allows us to successfully detect map-level closures that can be effectively incorporated into a pose-graph for optimization. We implemented and evaluated our approach on different datasets to show the generalizability of our approach to different scenarios and provided comparisons to other existing techniques. We provide a thorough experimental evaluation supporting all claims made in this paper. Our conclusion from these experiments is that utilizing local density maps for detecting loop closures is a useful direction for building SLAM systems.

REFERENCES

[1] M. Bosse and R. Zlot. Place Recognition Using Keypoint Voting in Large 3D Lidar Datasets. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2013.

[2] N. Chebrolu, T. Läbe, O. Vysotska, J. Behley, and C. Stachniss. Adaptive Robust Kernels for Non-Linear Least Squares Problems. *IEEE Robotics and Automation Letters (RA-L)*, 6(2):2240–2247, 2021.

[3] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, and C. Stachniss. OverlapNet: Loop Closing for LiDAR-based SLAM. In *Proc. of Robotics: Science and Systems (RSS)*, 2020.

[4] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena. SegMap: 3D Segment Mapping using Data-Driven Descriptors. In *Proc. of Robotics: Science and Systems (RSS)*, 2018.

[5] L. Giammarino, I. Aloise, C. Stachniss, and G. Grisetti. Visual Place Recognition using LiDAR Intensity Information. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2021.

[6] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Trans. on Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[7] G. Grisetti, C. Stachniss, and W. Burgard. Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Trans. on Robotics (TRO)*, 23(1):34–46, 2007.

[8] L. He, X. Wang, and H. Zhang. M2DP: A Novel 3D Point Cloud Descriptor and Its Application in Loop Closure Detection. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[9] B. Jiang and S. Shen. 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)*, 2023.

[10] A. Johnson and M. Hebert. Using Spin Images for Effcient Object Recognition in Cluttered 3D Scenes. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 21(5):433–449, 1999.

[11] M. Jung, W. Yang, D. Lee, H. Gil, G. Kim, and A. Kim. HeLiPR: Heterogeneous LiDAR Dataset for inter-LiDAR Place Recognition under Spatiotemporal Variations. *arXiv preprint*, arXiv:2309.14590, 2024.

[12] W. Kabsch. 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, 1976.

[13] G. Kim, S. Choi, and A. Kim. Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments. *IEEE Trans. on Robotics (TRO)*, pages 21–27, 2021.

[14] G. Kim, Y. Park, Y. Cho, J. Jeong, and A. Kim. Mulran: Multimodal range dataset for urban place recognition. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[15] G. Kim and A. Kim. 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)*, 2018.

[16] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. g2o: A General Framework for Graph Optimization. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[17] Y. Li and H. Li. LiDAR-Based Initial Global Localization Using Two-Dimensional (2D) Submap Projection Image (SPI). In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[18] L. Luo, S. Cao, B. Han, H. Shen, and J. Li. BVMatch: Lidar-Based Place Recognition Using Bird's-Eye View Images. *IEEE Robotics and Automation Letters (RA-L)*, 6:6076–6083, 2021.

[19] L. Luo, S. Cao, Z. Sheng, and H. Shen. LiDAR-Based Global Localization Using Histogram of Orientations of Principal Normals. *IEEE Trans. on Intelligent Vehicles*, 7(3):771–782, 2022.

[20] J. Ma, J. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen. 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, 2022.

[21] M. Magnusson, H. Andreasson, A. Nuechter, Achim, and J. Lilienthal. Appearance-Based Loop Detection from 3D Laser Data Using the Normal Distributions Transform. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009.

[22] E. Olson. Recognizing Places using Spectrally Clustered Local Matches. *Journal on Robotics and Autonomous Systems (RAS)*, 57(12):1157–1172, 2009.

[23] M. Ramezani, Y. Wang, M. Camurri, D. Wisth, M. Mattamala, and M. Fallon. The Newer College Dataset: Handheld LiDAR, Inertial and Vision with Ground Truth. In *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[24] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *Proc. of the IEEE Intl. Conf. on Computer Vision (ICCV)*, 2011.

[25] R. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009.

[26] S. Salti, F. Tombari, and L. Stefano. SHOT: Unique Signatures of Histograms for Surface and Texture Description. *Journal of Computer Vision and Image Understanding (CVIU)*, 125:251–264, 2014.

[27] D. Schlegel and G. Grisetti. HBST: A Hamming Distance embedding Binary Search Tree for Visual Place Recognition. *IEEE Robotics and Automation Letters (RA-L)*, 3:3741–3748, 2018.

[28] C. Stachniss, J. Leonard, and S. Thrun. *Springer Handbook of Robotics, 2nd edition*, chapter Chapt. 46: Simultaneous Localization and Mapping. Springer Verlag, 2016.

[29] B. Steder, G. Grisetti, and W. Burgard. Robust Place Recognition for 3D Range Data Based on Point Features. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2010.

[30] B. Steder, M. Ruhnke, S. Grzonka, and W. Burgard. 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)*, 2011.

[31] H. Strasdat, C. Stachniss, and W. Burgard. Which Landmark is Useful? Learning Selection Policies for Navigation in Unknown Environments. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009.

[32] S. Umeyama. Least-squares Estimation of Transformation Parameters Between Two Point Patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 13(4):376–380, 1991.

[33] M. Vijaymeena and K. Kavitha. A Survey on Similarity Measures in Text Mining. *Machine Learning and Applications: An International Journal*, 3(1):19–28, 2016.

[34] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss. 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, 2023.

[35] O. Vysotska and C. Stachniss. Lazy Data Association For Image Sequences Matching Under Substantial Appearance Changes. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):213–220, 2016.

[36] O. Vysotska and C. Stachniss. Effective Visual Place Recognition Using Multi-Sequence Maps. *IEEE Robotics and Automation Letters (RA-L)*, 4(2):1730–1736, 2019.

[37] K. Wurm, C. Stachniss, and G. Grisetti. Bridging the Gap Between Feature- and Grid-based SLAM. *Journal on Robotics and Autonomous Systems (RAS)*, 58(2):140 – 148, 2010.

[38] J. Yang, Q. Zhang, Y. Xiao, and Z. Cao. TOLDI: An Effective and Robust Approach for 3D Local Shape Description. *Pattern Recognition*, 65:175–187, 2017.

[39] C. Yuan, J. Lin, Z. Zou, X. Hong, and F. Zhang. STD: Stable Triangle Descriptor for 3D place recognition. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2023.

[40] Y. Zhang, P. Shi, and J. Li. LiDAR-Based Place Recognition For Autonomous Driving: A Survey. *arXiv preprint*, arXiv:2306.10561, 2023.