Narrowing Your FOV With SOLiD: Spatially Organized and Lightweight Global Descriptor for FOV-Constrained LiDAR Place Recognition

Hogyun Kim , *Graduate Student Member, IEEE*, Jiwon Choi , *Graduate Student Member, IEEE*, Taehu Sim , *Student Member, IEEE*, Giseop Kim , and Younggun Cho , *Member, IEEE*

Abstract—We often encounter limited FOV situations due to various factors such as sensor fusion or sensor mount in real-world robot navigation. However, the limited FOV interrupts the generation of descriptions and impacts place recognition adversely. Therefore, we suffer from correcting accumulated drift errors in a consistent map using LiDAR-based place recognition with limited FOV. Thus, in this letter, we propose a robust LiDAR-based place recognition method for handling narrow FOV scenarios. The proposed method establishes spatial organization based on the range-elevation bin and azimuth-elevation bin to represent places. In addition, we achieve a robust place description through reweighting based on vertical direction information. Based on these representations, our method enables addressing rotational changes and determining the initial heading. Additionally, we designed a lightweight and fast approach for the robot's onboard autonomy. For rigorous validation, the proposed method was tested across various LiDAR place recognition scenarios (i.e., single-session, multi-session, and multi-robot scenarios). To the best of our knowledge, we report the first method to cope with the restricted FOV.

Index Terms—LiDAR, limited FOV, place recognition, lightweight, onboard computing.

I. INTRODUCTION

LIGHT detection and ranging (LiDAR) based place recognition (PR) is critical for the robot navigation, multi-robot mapping [1], [2], and simultaneous localization and mapping (SLAM) problems. Unlike vision-based PR methods [3], [4], [5] that have problems with illumination variations, visual changes,

Hogyun Kim, Jiwon Choi, and Younggun Cho are with the Department of Electrical and Computer Engineering, Inha University, Incheon 22212, South Korea (e-mail: hg.kim@inha.edu; jiwon2@inha.edu; yg.cho@inha.ac.kr).

Taehu Sim is with the Department of Electrical Engineering, Inha University, Incheon 22212, South Korea (e-mail: xogn2359@gmail.com).

Giseop Kim is with the Vision Group of NAVER LABS, Seongnam 13561, South Korea (e-mail: giseop.kim@naverlabs.com).

Our place description and SLAM codes will be released. Also, the supplementary materials of our descriptor are available at https://sites.google.com/view/lidar-solid.

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3440089, provided by the authors.
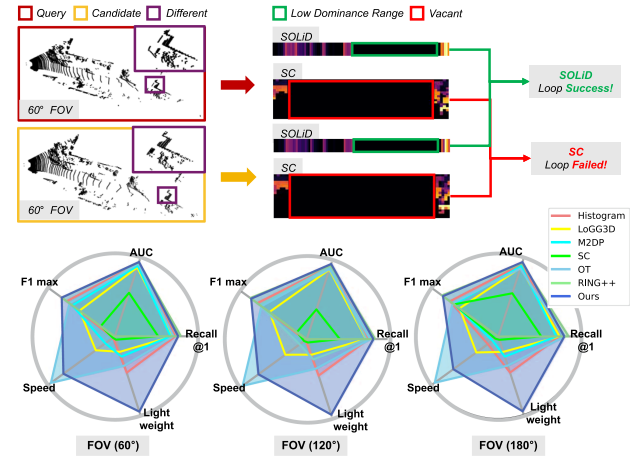
Fig. 1. At the top is our loop matching between the 1631st (red) and 192nd (yellow) frame in the KITTI 00 datasets clipped with a 60° FOV. The loop matching exhibits slight differences in rotation. Our methodology (green) implies a low dominance range, however, Scan Context (SC) (red) [9] signifies vacant information in the description. As the FOV narrows, it becomes more sensitive to subtle changes in small (purple) areas, directly correlating with success or failure. At the bottom are 3 phi charts that consist of recall@1, auc score, f1 max score, time, and descriptor size in KITTI 00 datasets with 60°, 120°, and 180° FOV. Recall@1, auc score, and f1 max score are higher values, the better performance. The higher values of speed and lightweight represent the low computational cost and the lightweight descriptor. We compare our method with well-known methods in LiDAR PR [6], [7], [9], [10], [11], [12] where discussed in Section II-A.

and time consumption in estimating 3-D information, LiDAR PR approaches [6], [7], [8], [9], [10], [11], [12] have become more prominent.

Although recent studies have shown remarkable results, reliable localization remains challenging in various situations. First, we often encounter situations where the field of view (FOV) is restricted through fusion with other sensors [13] or occluded by mechanical components [14] or the robot/sensor operator [15]. Also, existing methods are challenging to deal with various vertical resolutions via LiDAR's channel option (e.g. 6–128) or different scanning patterns such as solid-state LiDAR [16], [17]. Moreover, we need to consider what information we can convey in the description (e.g. semi-metric) and processing time (i.e. descriptor generation and loop searching) as well as summarizing the place compactly. Otherwise, traditional descriptors are challenging to be utilized on onboard computers.

Unlike other approaches, we propose a spatially organized and lightweight global descriptor called *SOLiD* for LiDAR PR to overcome these constraints. The main contributions of this work are the following:

- **Flexibility in Narrow FOV:** Our method applies to various localization scenarios with narrow FOV LiDAR as shown in Fig. 1. Additionally, it ensures performance regardless of vertical resolution and can be applied to solid-state LiDAR with narrow FOV and different scan patterns.
- **Lightweight and Fast Description:** The proposed spatial encapsulation ensures lightweight description and fast searching. As shown in Fig. 1, our descriptor's performance is superior overall, making it well-suited for deployment on onboard computers with limited memory. Also, our method runs in real-time up to 80 Hz without using GPU.
- **Rotational Robustness and Semi-metric Localization:** The proposed method achieves rotational robustness that can deal with rotation changes in narrow FOV. Furthermore, our approach provides a loop detection index and directly derives the 1-DoF angle (initial heading) to contribute to the accuracy of SLAM.
- **Evaluation and Open-source:** We validate the proposed method in various challenging LiDAR localization scenarios (single session, multi-session, and multi-robots) with extensive quantitative metrics on localization performance and efficiency. Also, the source code will be available and easy to utilize for existing SLAM frameworks (https://github.com/sparolab/solid.git).

II. RELATED WORKS

LiDAR PR methods can be broadly categorized into those that directly utilize geometric information (handcrafted methods) and those that leverage embedding vectors (deep learning methods). To cope with various situations, LiDAR PR systems must exhibit flexibility to handle limited FOV effectively. Thus, we discuss traditional LiDAR PR methodologies and their applicability to limited FOV situations.

A. Place Recognition for LiDAR

Histogram [6] was crafted as a vector that gauges the intervals between points and constructed a cumulative distribution function based on the counts of these intervals. M2DP [7] analyzed the principal components of points in the current scene and projected the points onto these principal components to reduce dimensionality. Although these methods are lightweight 1-D descriptors, they require prolonged construction times. While pre-processing, such as downsampling, is possible, there is a trade-off with performance degradation, necessitating further consideration. DeLight [8] generated 16 non-overlapping 3-D bins in spherical coordinates and incorporated intensity information into each bin. Because the intensity is a relative value, intensity calibration is necessary. LoGG3D-Net (LoGG3D) [10] employed local consistency loss and local features to improve the performance of global descriptors. OverlapTransformer (OT) [11] based on range images utilized the transformer model to enhance the LiDAR PR performance. As with most deep learning networks, there is a generalization issue. In addition, none of these methods proceed with estimating the initial pose. SC [9] created 2-D bins in polar coordinates and designated the representative value of each bin as the height of the highest point. Estimated rotation via column shifting could assist in pose

correction using the iterated closest point (ICP). RING [18] was generated by the Radon Transform (RT) of a binary bird's eye view (BEV) image. Then, a translational invariant descriptor called TI-RING was created using the Fast Fourier Transform (FFT) of RING. However, as they are 2-D descriptors, memory efficiency diminishes over long-term autonomy. RING++ [12], the SOTA method, generated 6 BEV images through 6 features similar to RING. However, RING++ is a 3-D descriptor, and the loop searching speed decreases more than the 2-D method.

B. Place Recognition in Limited FOV

We often encounter limited FOV in robot navigation. When the FOV is limited, the reduced number of bins in the constructed Histogram makes it challenging to describe the current scene effectively. Moreover, M2DP exhibits performance limitations due to the constrained range of principal components, and LoGG3D's performance is also degraded due to the loss of the number of local features. Additionally, when the orientation exceeds 180°, 50% of the range image becomes vacant space, causing a loss of functionality in OT. Furthermore, DeLight's 8 bins are transformed into empty space. Similarly, SC becomes disabled as the description itself contains more than 50% blank space. RING and RING++ benefit from the construction of rays from the RT, providing a relatively unrestricted capability in dealing with vacant spaces. However, beneath this lies an inherent limitation for real-time performance, as the focus has been primarily on optimizing overall performance.

When considering various methodologies, it is observed that traditional approaches including [9], [12], [18], [19], [20] often rely on the BEV representation, which projects the z-axis. However, in scenarios where the FOV is restricted, LiDAR's horizontal and vertical angles are similar, and in some cases, the vertical angle can even be significant such as Livox AVIA (70.4° (Horizontal) × 77.2° (Vertical)).

III. METHOD

In this section, we define the spatial organization and describe *SOLiD* generation from a 3-D scan and calculation of the distance between two *SOLiD* as represented in Fig. 2. Spatial organization refers to organizing a 3-D space into 2-D bins along the range-elevation and azimuth-elevation directions using polar coordinates and determining representative values for these bins. Unlike the BEV representation, the spatial organization includes elevation information, allowing for multi-directional analysis of 3-D space. Therefore, to achieve place recognition within a limited FOV, we utilize spatial organization and create *SOLiD* by reweighting vertical information.

A. Spatially Organized Place Description

1) REC ($\mathcal{R}$) & AEC ($\mathcal{A}$): The key idea of *SOLiD* is inspired by the counting algorithm [6] and spherical representation [8]. First, we define the 3-D space in spherical coordinates. Second, we organize the 3-D space into 3-D bins in the radial, azimuthal, and elevational directions. Finally, we downsample the points to a voxel size of 0.5 m and count the number of points within each 3-D bin. The points x, y, and z in the current 3D scan are converted to spherical coordinates as below:

$$\begin{cases} r = \sqrt{x^2 + y^2} \\ \theta = \text{atan2}(y, x) \\ \phi = \arctan(\frac{z}{r}). \end{cases} \tag{1}$$
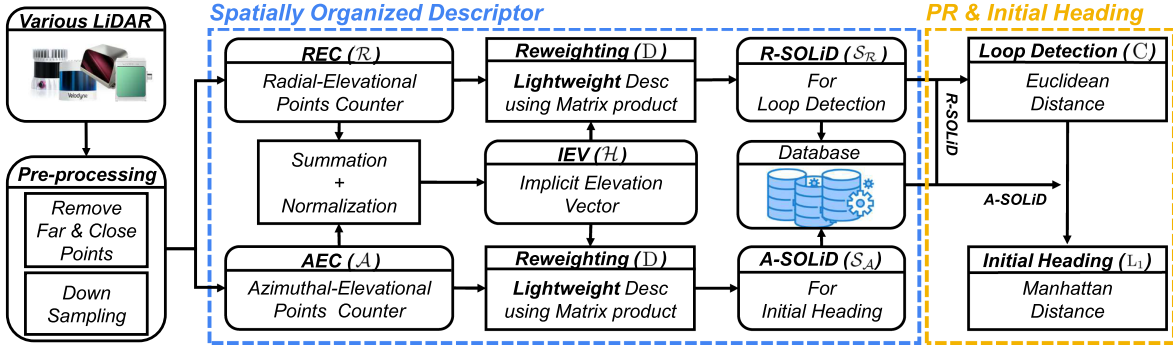
Fig. 2. Overall pipeline of our algorithm. Angled and rounded rectangles represent data and algorithms.

The number of points in each bin signifies the importance of the 3-D bin itself. Therefore, we assign each bin the number of points as a representative. However, we bear the burden of high computational costs owing to the heavyweight size if we utilize the 3-D bins as a place descriptor. Also, these bins often suffer from the issue of being empty, and the sparsity increases with distance. Unlike existing methods using elevational projection (i.e. BEV) [9], [12], [18], [19], [20], we utilize the spatial organization in both the radial and azimuthal directions to generate dense representations, namely, *AEC* ($\mathcal{A}$) and *REC* ($\mathcal{R}$), as follows:

$$\mathcal{R} = \bigcup_{i \in [N_r],\ k \in [N_e]} \mathcal{R}_{ik}, \qquad \mathcal{A} = \bigcup_{j \in [N_a],\ k \in [N_e]} \mathcal{A}_{jk}, \quad (2)$$

where N_r, N_a, and N_e denote the number of radial, azimuthal, and elevational bins, respectively. The symbols $[N_r]$, $[N_a]$, and $[N_e]$ are equal to $\{1, 2, \ldots, N_{r-1}, N_r\}$, $\{1, 2, \ldots, N_{a-1}, N_a\}$, and $\{1, 2, \ldots, N_{e-1}, N_e\}$. Each $\mathcal{R}$ and $\mathcal{A}$ is composed of indices i, k and j, k, respectively, and indices i, j, k ($i, j, k \in \mathbb{Z}$) can be determined by range r, azimuth θ, elevation ϕ as follows:

$$i \in \left\{ \mathrm{R}\left(N_r \times \frac{r}{L_{\max}} + 1\right) \ \middle| \ 0 \le r < L_{\max},\ r \in \mathbb{R} \right\}, \quad (3)$$

$$j \in \left\{ \mathrm{R}\left(N_a \times \frac{\theta}{360} + 1\right) \ \middle| \ 0 \le \theta < 360,\ \theta \in \mathbb{R} \right\}, \quad (4)$$

$$k \in \left\{ \mathrm{R}\left(N_e \times \frac{\phi - F_d}{F_u - F_d} + 1\right) \ \middle| \ F_d \le \phi < F_u,\ \phi \in \mathbb{R} \right\}, \quad (5)$$

where function $\mathrm{R}(\cdot, \cdot)$ means round function, $L_{\max}$ is the maximum observable distance and, F_u and F_d represent the down and up vertical FOV of the LiDAR sensor, respectively. As shown in Fig. 3, the number of points in the index i, j, k bin become the values that make up the *REC* and *AEC*. As a result, we do not overlook vertical information by describing a place using spatial organization within a limited FOV where the amount of information in the vertical direction is comparable to that in the horizontal direction.

2) IEV ($\mathcal{H}$): Using only *REC* or *AEC* for place recognition problems can cause system performance degradation. For instance, on a solid-state LiDAR, the number of points belonging to a 3-D bin can be different due to non-repetitive scan patterns, even at the same place. Also, in limited FOV situations, *REC* information becomes uncertain, and *AEC*, similar to SC, incorporates vacant spaces as described in Fig. 1. To address this issue, we apply reweighting using *IEV* ($\mathcal{H}$), which effectively
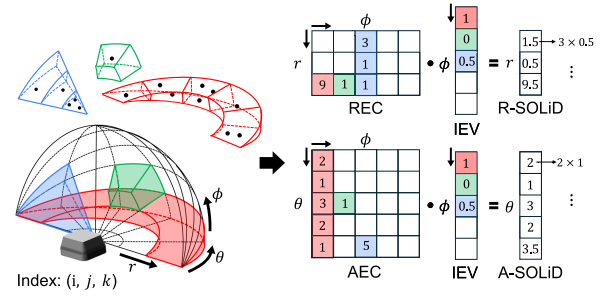

Fig. 3. Simple example schema of our methodology. The indices i, j, k are determined along the r, θ, ϕ axes. Utilizing the spatial organization, 3-D bins are projected onto *REC* and *AEC* according to their respective indices. Subsequently, *R-SOLiD* and *A-SOLiD* are generated by taking the matrix product with *IEV*.
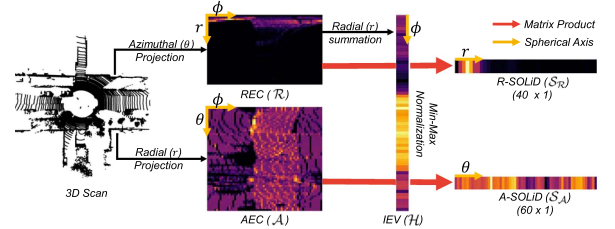

Fig. 4. The process of generating *R-SOLiD* and *A-SOLiD*. First, when a 3-D scan is received, *REC* and *AEC* are generated through spatial organization. Second, summation is performed in the radial direction, followed by the creation of *EC*. Third, min-max normalization is applied to the created *EC*, resulting in the generation of *IEV*. Finally, the matrix product of *IEV* with *REC* and *AEC* respectively yields *R-SOLiD* and *A-SOLiD*.

highlights the spatially dominant components. To obtain *IEV*, we first create *EC* by summing *REC* in the radial direction, as represented in Fig. 4. Summing *AEC* in the azimuthal direction also becomes *EC*, but since N_r is typically smaller than N_a, generating *EC* via *REC* is more time-efficient. Thus, the *EC* is represented as follows:

$$\mathcal{E} = [\psi_1, \psi_2, \ldots, \psi_{N_e}], \qquad \psi_k = \sum_{i=1}^{N_r} \mathcal{R}_{ik}. \quad (6)$$

EC is a measure that examines which vertical angles have a higher number of points (i.e. dominant) in an elevation-wise manner. However, as mentioned above, the number of points is vulnerable to various issues such as non-repetitive scan patterns. Therefore, we generate *IEV* ($\mathcal{H}$) through min-max normalization of *EC* ($\mathcal{E}$) to identify the tendency for vertical structure at the

current place as below:

$$\mathcal{H} = \frac{\mathcal{E} - \mathcal{E}_{\min}}{\mathcal{E}_{\max} - \mathcal{E}_{\min}}, \quad (7)$$

where $\mathcal{E}_{\min}$ and $\mathcal{E}_{\max}$ are min and max value in EC ($\mathcal{E}$). As a result, *IEV* is a robust weight that includes elevation-wise dominance and can cope with different scan patterns.

3) SOLiD ($\mathcal{S}$): Finally, we acquire a *SOLiD* by reweighting, which takes the matrix product between *REC*, *AEC*, and *IEV* ($\mathcal{H}$) are as follows:

$$\mathcal{S} = [\mathcal{S}_\mathcal{R}, \mathcal{S}_\mathcal{A}], \quad \begin{cases} \mathcal{S}_\mathcal{R} = \mathrm{D}(\mathcal{R}, \mathcal{H}) \\ \mathcal{S}_\mathcal{A} = \mathrm{D}(\mathcal{A}, \mathcal{H}), \end{cases} \quad (8)$$

where function $\mathrm{D}(\cdot, \cdot)$ means matrix production. *R-SOLiD* ($\mathcal{S}_\mathcal{R}$) and *A-SOLiD* ($\mathcal{S}_\mathcal{A}$) form vectors that encapsulate radial and azimuthal dominance. This strategy propagates the current place's element-wise dominance to each range and azimuth. In detail, the element-wise multiplication and summation are performed between all elevation elements within each range and azimuth of *REC*, *AEC*, and all elements of *IEV* as represented in Fig. 3. This step is called reweighting, where higher-value elements converge higher-value elements than before and lower-value elements converge lower-value than before. Each *SOLiD* is a vector representing the degree of dominance for the given range and azimuth within the current place. In this work, $\mathcal{S}_\mathcal{R}$ is used for loop detection, and $\mathcal{S}_\mathcal{A}$ is used for initial heading estimation, where each consists of sizes N_r and N_a. In addition, we determine that $N_r = 40$, $N_a = 60$, and N_e are aligned with the vertical resolution of the LiDAR.

B. Place Recognition & Initial Heading Estimation

1) Place Recognition: To recognize the revisited place, we compare the cosine distance between the query *R-SOLiD* ($\mathcal{S}_\mathcal{R}^q$) and the candidate *R-SOLiD* ($\mathcal{S}_\mathcal{R}^c$) in the database as follows:

$$\mathrm{C}(\mathcal{S}_\mathcal{R}^q, \mathcal{S}_\mathcal{R}^c) = \left(1 - \frac{\mathcal{S}_\mathcal{R}^q \cdot \mathcal{S}_\mathcal{R}^c}{||\mathcal{S}_\mathcal{R}^q|| \cdot ||\mathcal{S}_\mathcal{R}^c||}\right). \quad (9)$$

By constructing a kd-tree, we search for candidates with the closest distances in the database and recognize them as revisited places.

2) Initial Heading Estimation: Owing to the semi-metric representation of the $\mathcal{S}_\mathcal{A}$, each unit movement in the vector $\mathcal{S}_\mathcal{A}$ represents a corresponding heading (yaw). Thus, we find the nearest distance index n^*, shifting the candidate *A-SOLiD* ($\mathcal{S}_{\mathcal{A}_n}^c$) by n unit as follows:

$$n^* = \operatorname*{argmin}_{n \in [N_a]}(||\mathcal{S}_\mathcal{A}^q - \mathcal{S}_{\mathcal{A}_n}^c||). \quad (10)$$

IV. RESULTS

All experiments were conducted on desktop Intel i7-12700 KF with RTX 3070Ti and onboard computer Jetson AGX Orin to assess onboard computing capabilities. Furthermore, we differentiated experiment results for easy comprehension: **first rank** in bold blue, the **second rank** in green bold, and the **third rank** in black bold, respectively.

A. Experiments Setup

1) Comparative Methods: We presented Histogram [6], M2DP [7], SC [9], LoGG3D [10], OT [11], and RING++ [12] as comparative methods, generated using Python-based code to
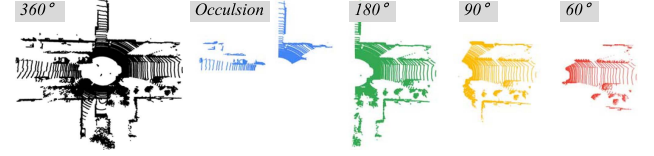

Fig. 5. 500th frame from various angles in KITTI 00. Black represents full view (360°) points. Blue, green, yellow, and red mean occlusion, 180°, 120°, and 60° points, respectively.

meet the same conditions. We also differentiated between methods that can be used only with CPU and those where GPU usage is inevitable. Each method was labeled for easy identification as either (C) for CPU-only or (G) for GPU-dependent. Especially, we utilized the pre-trained models provided for deep-learning methods (LoGG3D and OT).

Each of these methods employs different search strategies. Among them, radius search strongly assumes that the initial pose is known. Therefore, we converted radius search into brute force searching. For a fair comparison, the proposed method also derived results exclusively using brute force searching. The results obtained using the kd-tree method were explained in Section IV-D.

2) Scenarios: For extensive evaluation, we validated across various scenarios (single session, multi-session, and multi-robot system). For each session, we deployed different LiDARs (different vertical angles, horizontal angles, channels, or scan patterns). As shown in Fig. 5, we modified 360° FOV datasets to restricted FOV datasets by clipping the pointcloud to assess performance in dealing with limited FOV.

We validated the performance across various narrow FOV scenarios and LiDAR scan occlusion situations in the well-known KITTI dataset [21] of a single session. KITTI was a sequence of driving cars equipped with Velodyne HDL-64E LiDAR. We evaluated various metrics through loops within 10 meters using sequences 00, 02, and 05.

In a multi-session HeLiPR dataset [17], we evaluated the performance of solid-state LiDAR with heterogeneous scan patterns and limited FOV. HeLiPR was a sequence of driving cars, and we tested the robustness of PR through solid-state LiDARs, Aeva Aeries II and Livox-Avia. In this experiment, we leveraged the sequences KAIST04-05, Town01-02, and RoundAbout01-02. In this dataset, the vehicle remained in the same spot for extended periods due to waiting at traffic lights. Therefore, we sampled the pointcloud at intervals of 2 meters based on the ground truth pose and validated using loops within 5 meters.

In a multi-robot scenario Park dataset [1], we measured the performance across various constrained FOV setups and captured the communication time between robots. Park was a sequence where three robots equipped with Velodyne VLP-16 LiDAR map a designated area. Since each robot needs to be aware of its relationship with the other two robots, we evaluated PR based on several metrics through six experiments. Since robot driving was inevitably slower than driving cars, we sampled at intervals of 2 meters and considered loops within 5 meters as correct loops.

B. Evaluation Metrics

1) PR Curve: For evaluating the performance of loop detection, the Precision-Recall (PR) curve was utilized. Precision and

TABLE I
COMPARISON OF DESCRIPTOR SIZES

Descriptor	Hist. (C)	LoGG3D (G)	M2DP (C)	SC (C)	OT (G)	RING++ (G)	Ours (C)
Shape	100×1	256×1	192×1	20×60	256×1	$6 \times 120 \times 120$	40×1
Size [B]	**928**	2176	1664	10016	**1152**	345728	**448**

recall [22] are defined as below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (11)$$

where TP, FP, and FN are true positive, false positive, and false negative, respectively.

2) ROC Curve and AUC Score: We also plotted the Receiver operating characteristic (ROC) curve to assess the ability to distinguish between true and false loops. True Positive Rate (TPR) and False Positive Rate (FPR) are defined as below:

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}, \quad \text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad (12)$$

where TP, FP, and TN, FN are true positive, false positive, and true negative, false negative, respectively. We also evaluated the AUC score, which represents the area under the ROC curve. TPR should be greater than FPR for a model to diagnose true and false loops. Thus, the method has no discriminatory ability if the AUC score is less than 0.5.

3) Recall@1 and F1 Score: We derived Recall@1 as a metric for performance in finding loops, rather than the PR curve or ROC curve. Additionally, we derived the F1 score as a metric for performance, which is the harmonic mean between Precision and Recall. Each Recall@1 and F1 score [22] metric is defined as follows:

$$\text{Recall@1} = \frac{\text{TP}}{\text{GT}}, \quad \text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (13)$$

where TP, GT, Precision, and Recall are true positive, the number of ground truth, precision, and recall are mentioned above, respectively.

4) Rotation Error: We evaluated a Rotation Error (RE) [22], which is the error between the estimated rotation and the ground truth rotation calculated by

$$\text{RE} = \arccos((\text{trace}(\hat{\text{R}}^T \cdot \text{R}) - 1)/2), \quad (14)$$

where $\hat{\text{R}}^T$ is the estimated rotation matrix and R is the ground truth rotation matrix.

5) Processing Time: In PR experiments, much time is consumed in both the process of generating descriptions and finding loops. The process of generating descriptions remains consistent even as the database size increases. However, loop searching time inevitably increases as the database grows. Therefore, we designated and measured the processing time by summing the average time of the entire loop searching process with the time taken for description generation. We verified the processing time on the desktop and onboard computer, labeling them respectively as (D) and (O). Especially in the case of onboard computers (O), including laptops installed on real robots, GPU-based methods were not considered in ranking processing time due to the lack of GPU.

C. Lightness

As represented in Table I, we measured the size and shape of each descriptor for comparison. The proposed method is at least approximately $2\times$ lighter, up to $1000\times$ lighter compared with traditional methods. Thus, our method is more effective in large-scale mapping as it can be stored for longer periods in a finite number of storages. Especially, our method is more effective on onboard computers with limited memory capacity. In the following sections, we can see that our method performs comparatively or better than other methods. This suggests that the proposed method is lightweight and achieves robust PR using spatial organization and reweighting in limited FOV. In particular, RING++ employs a 3-D descriptor of size $6 \times 120 \times 120$, as it minimizes compression in the 3-D space. While this lack of dimension reduction allows for a more diverse representation of the scene, it is natively a heavyweight descriptor. Furthermore, the processing time naturally increases because heavyweight descriptors have more space to compare.

D. Single Session Place Recognition

1) Narrow FOV: As described in Table II, our approach and RING++ outperform across various metrics than other approaches. The proposed method finds fewer loops overall compared to RING++. However, our method has an outstanding ability to distinguish between correct loops and incorrect loops, which affects SLAM performance. Especially in terms of time, it is overwhelmingly superior. Therefore, our method maintains a consistent map efficiently and quickly even in limited FOV. We also compared the proposed method using the kd-tree and brute force searching. While the kd-tree is about 10–20 Hz faster, its performance is slightly lower or similar. Therefore, it can be effectively used according to the trade-off. Unlike Table II, the Table III reports processing time from the onboard computer in KITTI 00 sequence. On the desktop, LoGG3D, M2DP, OT, and our method can operate in real-time, whereas on an onboard computer, only OT and the proposed method can achieve real-time. However, OT falls short in terms of performance. Furthermore, as shown in Tables II and III, narrowing the FOV results in lower processing time consumption. This means that if we can robustly handle limited FOV, we can utilize descriptors more efficiently on the onboard computer.

2) Rotational Robustness: To evaluate the robustness of the rotation changes, we checked the top 1 recall by applying explicit heading changes on KITTI 05 with 180 FOV. Except when the surroundings are completely symmetric, our method shows higher recall values compared to other methods.

When the environment is completely symmetrical, it is difficult to recognize a revisited place with little information because the overlap is either non-existent or vertical planes. On the other hand, when a rotational change occurs within a limited FOV, there is overlap and we can utilize this to counteract the rotational change as shown in Fig. 6.

3) Initial Heading Estimation: As represented in Fig. 1, slight errors on narrow FOV can have a relatively larger impact compared to a 360° FOV. Thus, we evaluated the RE between an estimated initial heading from *A-SOLiD* and a ground truth heading. Table IV indicates that *A-SOLiD* estimates the initial heading to be around 1° of the ground truth heading, making it similar. This means that our method can be given as the initial rotation for ICP, which leads to accurate map building.

4) Occlusion: In robot navigation, occlusion can often be encountered when there are nearby objects. Assuming such a situation, we restricted the FOV as shown in Fig. 5.

TABLE II
EVALUATION IN KITTI DATASETS

FOV	Method	Sequence											
		00 (4541 frames)				02 (4661 frames)				05 (2761 frames)			
		Recall@1	AUC	F1 max	Time (D) [Hz]	Recall@1	AUC	F1 max	Time (D) [Hz]	Recall@1	AUC	F1 max	Time (D) [Hz]
60	Hist. (C)	0.764	0.965	0.754	3.06	0.483	0.960	0.452	3.05	0.611	0.949	0.606	3.21
	Logg3D (G)	0.726	0.914	0.552	30.76	0.674	0.954	0.627	30.59	0.593	0.936	0.531	33.55
	M2DP (C)	0.703	0.935	0.664	15.40	0.471	0.972	0.469	15.36	0.468	0.969	0.517	15.96
	SC (C)	0.547	0.577	0.220	3.24	0.177	0.116	0.031	3.24	0.373	0.399	0.159	3.29
	OT (G)	0.740	0.964	0.708	135.14	0.651	0.946	0.574	133.51	0.540	0.958	0.553	164.74
	RING++ (G)	0.834	0.982	0.831	0.45	0.701	0.993	0.729	0.44	0.688	0.974	0.753	0.76
	Ours-BF (C)	0.800	0.987	0.830	57.94	0.706	0.990	0.746	57.34	0.639	0.992	0.732	68.40
	Ours-KD (C)	0.800	0.986	0.831	82.58	0.706	0.990	0.745	82.01	0.639	0.992	0.732	87.66
120	Hist. (C)	0.780	0.961	0.771	1.57	0.483	0.965	0.427	1.57	0.596	0.958	0.613	1.61
	Logg3D (G)	0.746	0.920	0.581	26.15	0.674	0.983	0.664	26.03	0.625	0.910	0.511	28.14
	M2DP (C)	0.751	0.959	0.728	8.67	0.404	0.987	0.441	8.66	0.460	0.978	0.525	8.85
	SC (C)	0.417	0.393	0.152	3.21	0.151	0.155	0.045	3.20	0.332	0.379	0.126	3.25
	OT (G)	0.799	0.927	0.705	97.18	0.691	0.969	0.616	96.33	0.614	0.891	0.516	111.61
	RING++ (G)	0.862	0.977	0.880	0.45	0.750	0.978	0.815	0.44	0.732	0.974	0.793	0.75
	Ours-BF (C)	0.822	0.994	0.868	45.19	0.735	0.992	0.790	44.82	0.656	0.989	0.737	51.31
	Ours-KD (C)	0.840	0.994	0.867	58.89	0.735	0.992	0.794	58.61	0.659	0.974	0.710	41.75
180	Hist. (C)	0.780	0.966	0.761	1.09	0.462	0.965	0.410	1.08	0.542	0.952	0.570	1.10
	Logg3D (G)	0.780	0.918	0.609	22.87	0.701	0.966	0.647	22.77	0.611	0.903	0.492	24.37
	M2DP (C)	0.739	0.940	0.694	6.03	0.523	0.981	0.555	6.02	0.497	0.921	0.526	6.11
	SC (C)	0.547	0.577	0.694	3.16	0.177	0.116	0.031	3.16	0.384	0.353	0.146	3.21
	OT (G)	0.842	0.907	0.719	76.34	0.698	0.962	0.646	75.82	0.645	0.864	0.466	84.96
	RING++ (G)	0.895	0.976	0.887	0.45	0.752	0.983	0.813	0.44	0.744	0.988	0.795	0.74
	Ours-BF (C)	0.840	0.994	0.868	37.61	0.735	0.990	0.799	37.36	0.659	0.974	0.710	41.75
	Ours-KD (C)	0.840	0.994	0.871	46.74	0.735	0.990	0.800	46.42	0.659	0.974	0.710	48.20

TABLE III
PROCESSING TIME ON ONBOARD COMPUTER

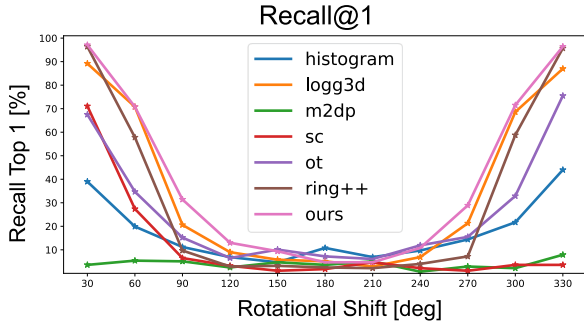| FOV | Processing Time (O) [s] | | | | | | |
	CPU				GPU		
	Hist. (C)	M2DP (C)	SC (C)	Ours (C)	OT (G)	LoGG3D (G)	RING++ (G)
60	1.289	0.297	1.359	0.073	0.036	0.160	10.85
120	2.442	0.531	1.371	0.093	0.047	0.188	10.92
180	3.550	0.710	1.382	0.111	0.056	0.215	10.98

Fig. 6. Experiment on KITTI 05 with a 180° FOV. The clipping angle of the query database is gradually shifted from front-facing to rotating, while the clipping angle of the candidate database remains constant front-facing.

TABLE IV
ROTATION ERROR

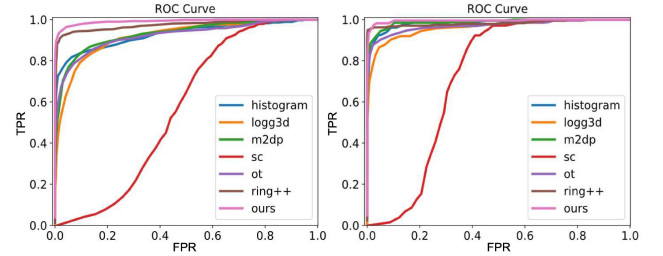| Sequence | KITTI 00 | | | KITTI 02 | | | KITTI 05 | | |
	60	90	180	60	90	180	60	90	180
FOV									
RE [°]	0.24	0.76	0.77	0.59	0.69	0.99	0.72	0.88	1.07

Fig. 7. The ROC curves when LiDAR scan is occluded. Left is KITTI 00's evaluation, and right is KITTI 02's evaluation.

TABLE V
NUMBER OF MATCHING PAIR IN KAIST04-05

LiDAR Type	Metric	Hist. (C)	LoGG3D (G)	M2DP (C)	SC (C)	OT (G)	RING++ (G)	Ours (C)
Aeva	TP ↑	1526	200	1191	1155	874	2116	2291
	FP ↓	848	884	740	1314	1888	171	196
	F1 max ↑	0.603	0.106	0.516	0.448	0.321	0.851	0.886
Avia	TP ↑	884	1233	732	1887	379	2208	2119
	FP ↓	607	1271	1508	717	2424	109	501
	F1 max ↑	0.424	0.477	0.298	0.716	0.138	0.901	0.801

As represented in Fig. 7, our method's graph is positioned at the highest point. In other words, our method can reduce drift error in occlusion situations through effective loop distinction.

E. Multi Session Place Recognition

We evaluated multi-session scenarios for long-term autonomy. As shown in Fig. 9, the proposed method, Histogram, and RING++ can maintain a high precision at a consistent recall level. Thus, these three methods possess the reliable ability to distinguish between false loops and true loops. Fig. 8 shows a 3-D loop matching graph in KAIST04-05 of multi-session, acquired with solid-state LiDAR. With Aeva, our method finds the many loops, while in Avia, RING++ shows the highest performance. Table V refers to the number of true positive and false positive loops in the Fig. 8's 3-D graph. In the Aeva sequence, RING++ achieves a precision that is about 0.03 higher. However, our method discovers higher recall and consequently maintains a higher f1 score. Especially when precision is 100%, our method detects more loops as shown in Appendix. In the Avia sequence, RING++ demonstrates superiority. However, it necessitates the use of GPU as indicated in the Table VI, leading to longer processing times. Also, OT has a short processing time but lacks in terms of f1 score or performance. On the other hand, our method offers both a short processing time and superior performance. Also, by utilizing kd-tree, even faster speeds can be achieved.
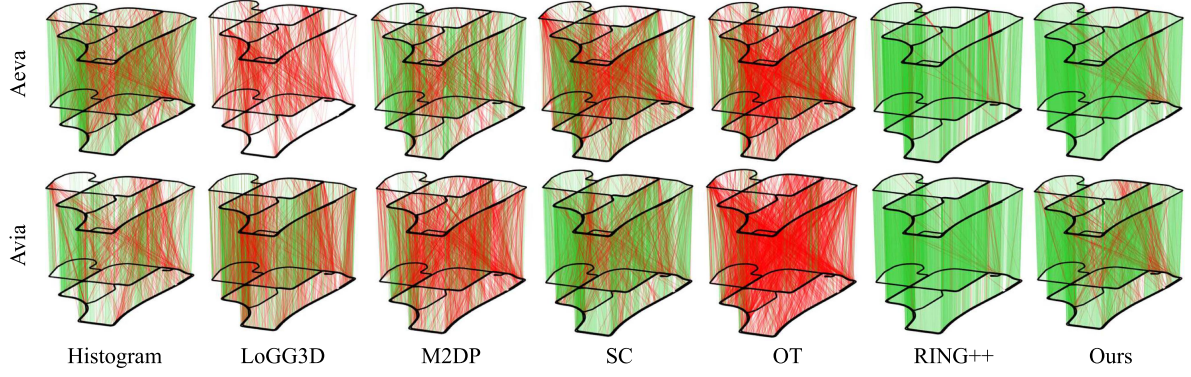
Fig. 8. 3-D matching pair graph when each descriptor's f1 score is highest in HeLiPR dataset (source: KAIST04, target: KAIST05). Green represents true matching pairs, and red represents false matching pairs.
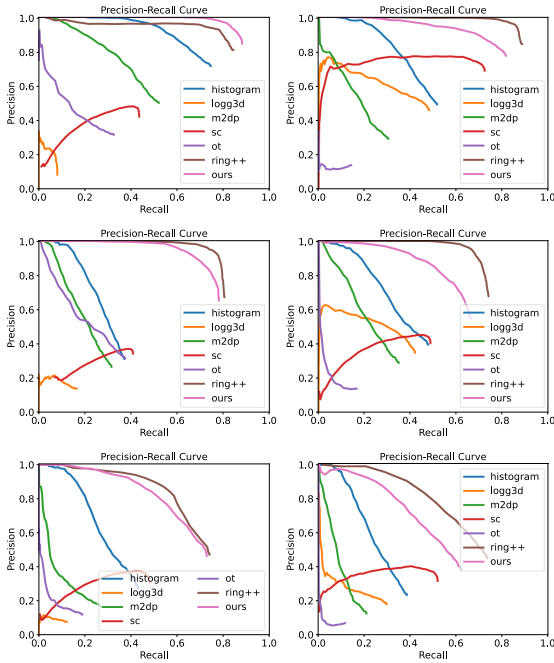

Fig. 9. PR curve on HeLiPR datasets. The left column used Aeva LiDAR and the right used Avia LiDAR. The top represents KAIST04-05, the middle means Town01-02 and the bottom means RoundAbout01-02 session.

TABLE VII
HARDWARE SETUP AND DATA SIZE

Robot ID	Hardware	Data size [MB]						
		Hist. (C)	LoGG3D (G)	M2DP (C)	SC (C)	OT (G)	RING++ (G)	Ours (C)
Robot 1	Jetson AGX Orin	**0.5**	1.3	1.0	6.0	**0.7**	202	**0.3**
Robot 2	Intel NUC 10 Mini PC	**0.6**	1.5	1.4	7.0	**0.8**	242	**0.3**
Robot 3	Intel NUC 12 Pro Mini PC	**0.8**	1.8	1.4	8.0	**1.0**	294	**0.4**

TABLE VIII
EVALUATION FOR MULTI-ROBOT SYSTEM

Method	FOV	Recall@1↑	AUC↑	F1 max↑	Commu. Time ↓
Hist. (C)	60	0.405	0.847	0.349	**0.285s**
	120	0.431	0.862	0.373	
	180	0.465	0.856	0.375	
LoGG3D (G)	60	**0.590**	0.843	**0.387**	0.669s
	120	0.647	0.890	0.471	
	180	0.654	**0.910**	0.516	
M2DP (C)	60	0.379	**0.874**	0.359	0.511s
	120	0.433	**0.919**	0.417	
	180	0.432	0.888	0.383	
SC (C)	60	0.474	0.648	0.272	3.077s
	120	0.468	0.579	0.272	
	180	0.497	0.570	0.278	
OT (G)	60	0.366	0.867	0.348	**0.354s**
	120	0.423	0.834	0.369	
	180	0.482	0.864	0.395	
RING++ (G)	60	**0.670**	**0.940**	**0.585**	106.221s
	120	**0.701**	**0.975**	**0.730**	
	180	**0.707**	**0.986**	**0.767**	
Ours (C)	60	**0.572**	0.881	**0.442**	**0.138s**
	120	**0.665**	0.919	**0.548**	
	180	**0.689**	0.930	**0.571**	

TABLE VI
PROCESSING TIME IN SOLID-STATE LIDAR

Computer	LiDAR Type	Processing Time [s]						
		CPU				GPU		
		Hist. (C)	M2DP (C)	SC (C)	Ours (C)	OT (G)	LoGG3D (G)	RING++ (G)
O	Aeva	5.291	**1.109**	**1.439**	**0.143**	0.083	0.229	15.70
	Avia	**1.260**	**0.258**	1.433	**0.137**	0.047	0.189	15.58
D	Aeva	1.426	0.271	0.145	**0.036**	**0.021**	**0.056**	3.720
	Avia	0.323	0.072	0.142	**0.033**	**0.012**	**0.039**	3.680

F. Multi-Robot Place Recognition

In multi-robot systems, the communication time to exchange descriptors for quickly and accurately knowing the positions of each other's robots is crucial. However, there are various constraints on communication, such as limited bandwidth in real multi-robot mapping. Therefore, within the given time, we need to transmit and receive important information. Longer communication times can lead to bottlenecks and increase the likelihood of data loss.

To measure communication time, we followed three steps. First, we assumed that each robot was installed on an onboard computer as represented in Table VII. Second, we measured each onboard computer's bandwidth under the assumption that there is no limitation on communication distance. Finally, we derived the communication time as follows:

$$\text{Communication Time} = \text{Description} \times \frac{1}{\text{Bandwidth}}, \quad (15)$$

where the communication time, description, and bandwidth have units of [s], [B], and [B/s], respectively.

We assumed that all descriptors generated among the three robots are exchanged after the mapping is completed. Thus, we evaluated the final result based on the sum of the total six communication times. As shown in Table VIII, our method exhibits the shortest communication speed. On the other hand, RING++

TABLE IX
ABLATION STUDY FOR LOOP DETECTION

Variable	Evaluation	Value					
		20	40	60	80	100	120
N_r	Recall@1	**0.871**	**0.874**	0.869	0.869	**0.873**	**0.871**
	AUC	**0.996**	**0.996**	**0.997**	**0.996**	0.995	0.994
	F1 max	**0.887**	**0.894**	**0.891**	0.883	0.877	0.866
N_e	Recall@1	**0.872**	**0.873**	0.871	**0.872**	0.866	0.871
	AUC	**0.996**	**0.996**	**0.996**	**0.996**	**0.997**	**0.996**
	F1 max	0.892	0.892	**0.893**	0.892	0.892	**0.894**

exhibits a relatively longer communication time. Also, we evaluated the overall PR performance in multi-robot sequences with narrow FOV. As described in Table VIII, the performance is superior in the order of RING++, the proposed method, and LoGG3D or M2DP. Although our method has slightly inferior performance compared to RING++, short communication time makes the proposed method more efficient for multi-robot scenarios.

G. Ablation Study

The key components of the loop detection are N_r and N_e. These values determined the *REC* of spatial organization, and we conducted an ablation study to investigate its impact.

Table IX illustrates the LiDAR PR performance of *R-SOLiD* as it varies with N_r and N_e in KITTI 00. While there is some variation based on the values, it can be observed that the changes are negligible, indicating that heuristic efforts through parameter tuning are not crucial.

V. CONCLUSION

We proposed a spatially organized and lightweight global descriptor for FOV-constrained LiDAR PR. To address limited FOV scenarios, we effectively represented the scene through spatial organization and reweighting. Furthermore, we confirmed efficient operation within onboard computing via lightweight description and fast searching. However, most LiDAR place recognition methods, including ours, show weakness in detecting places with reverse directions of limited FOV scenarios. If the robot is traveling in the opposite direction from the revisited place, there may be no overlap in the LiDAR view between the current place and the revisited place, or only a part of the vertical plane may overlap, even if the difference in translation is small. Therefore, to recognize the reverse revisited place with all methods, including our descriptor, it is necessary to leverage 360° information or utilize additional information, such as semantics or sequential queues. The applicability of our descriptor in the 360° view is confirmed as described in the Appendix on the supplementary materials' page.

In future works, we plan to build up sequential information to create a descriptor for recognizing revisited places in the opposite direction with a limited FOV. Also, we will integrate our method with the LiDAR odometry approaches and release a distributed SLAM framework.

REFERENCES

[1] Y. Huang, T. Shan, F. Chen, and B. Englot, "DiSCo-SLAM: Distributed scan context-enabled multi-robot LiDAR SLAM with two-stage global-local graph optimization," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 1150–1157, Apr. 2022.

[2] G. Kang, H. Kim, B. Choi, S. Jeong, Y.-S. Shin, and Y. Cho, "Unified mapping framework for multi-modal lidars in complex and dynamic environments," 2024. [Online]. Available: https://construction-robots.github.io/papers/67.pdf

[3] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5297–5307.

[4] H. Kim, G. Kang, S. Jeong, S. Ma, and Y. Cho, "Robust imaging sonar-based place recognition and localization in underwater environments," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 1083–1089.

[5] B. Choi, H. Kim, and Y. Cho, "ReFeree: Radar-based efficient global descriptor using a feature and free space for place recognition," 2024, Art. no. 02783649241242136.

[6] T. Röhling, J. Mack, and D. Schulz, "A fast histogram-based similarity measure for detecting loop closures in 3-D LIDAR data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 736–741.

[7] L. He, X. Wang, and H. Zhang, "M2DP: A novel 3D point cloud descriptor and its application in loop closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2016, pp. 231–237.

[8] K. P. Cop, P. V. Borges, and R. Dubé, "Delight: An efficient descriptor for global localisation using LiDAR intensities," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 3653–3660.

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

[10] K. Vidanapathirana, M. Ramezani, P. Moghadam, S. Sridharan, and C. Fookes, "LoGG3D-Net: Locally guided global descriptor learning for 3 D place recognition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 2215–2221.

[11] 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 Robot. Automat. Lett.*, vol. 7, no. 3, pp. 6958–6965, Jul. 2022.

[12] X. Xu et al., "RING++: Roto-translation-invariant gram for global localization on a sparse scan map," *IEEE Trans. Robot.*, vol. 39, no. 6, pp. 4616–4635, Dec. 2023.

[13] Y.-S. Shin, Y. S. Park, and A. Kim, "DVL-SLAM: Sparse depth enhanced direct visual-lidar SLAM," *Auton. Robots*, vol. 44, no. 2, pp. 115–130, 2020.

[14] S. Jeong, H. Kim, and Y. Cho, "DiTer: Diverse terrain and multi-modal dataset for field robot navigation in outdoor environments," *IEEE Sensors Lett.*, vol. 8, no. 3, pp. 1–4, Mar. 2024.

[15] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "MulRan: Multimodal range dataset for urban place recognition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6246–6253.

[16] D. Lee, M. Jung, and A. Kim, "ConPR: Ongoing construction site dataset for place recognition," in *Proc. IROS23 Workshop Closing Loop Localization*, 2024, *arXiv:2407.03684*.

[17] M. Jung, W. Yang, D. Lee, H. Gil, G. Kim, and A. Kim, "HeLiPR: Heterogeneous LiDAR dataset for inter-LiDAR place recognition under spatial and temporal variations," *Int. J. Robot. Res.*, 2023, *arXiv:2309.14590*.

[18] S. Lu, X. Xu, H. Yin, Z. Chen, R. Xiong, and Y. Wang, "One ring to rule them all: Radon sinogram for place recognition, orientation and translation estimation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 2778–2785.

[19] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, and R. Xiong, "DiSCO: Differentiable scan context with orientation," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 2791–2798, Apr. 2021.

[20] J. Ma, G. Xiong, J. Xu, and X. Chen, "CVTNet: A cross-view transformer network for LiDAR-based place recognition in autonomous driving environments," *IEEE Trans. Ind. Inform.*, vol. 20, no. 3, pp. 4039–4048, Mar. 2024.

[21] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[22] H. Yin et al., "A survey on global LiDAR localization: Challenges, advances and open problems," *Int. J. Comput. Vis.*, vol .132, pp. 1–33, 2024.