IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

DVMM: A Dual-View Combination Descriptor for Multi-Modal LiDARs Online Place Recognition

Xuzhe Duan, Qingwu Hu, Mingyao Ai, Pengcheng Zhao, and Jiayuan Li

***Abstract*—Existing place recognition descriptors developed for single-agent SLAM struggle with multi-modal LiDAR differences in collaborative SLAM. To overcome this, we propose an online place recognition method for multi-modal LiDARs. This method introduces a dual-view combination descriptor, termed DVMM, by separately encoding azimuthal and vertical scene information. The place recognition process consists of two stages: loop closure detection and verification. In the detection stage, point clouds are projected onto an adaptive grid and a 1D azimuthal descriptor is generated via Gaussian-weighted column summation. The azimuthal descriptor is utilized to retrieve loop candidates through vector matching. In the verification stage, point clouds within a fixed height range are encoded as a binary occupancy image, which serves as the cross-section descriptor. Accurate loop closures are determined by performing image matching on the cross-section descriptors. We evaluate the proposed method on both public and real-world datasets encompassing a total of seven LiDAR sensors. The results demonstrate that DVMM significantly outperforms state-of-the-art descriptors in handling multi-modal LiDAR data and is compatible with collaborative SLAM systems.**

***Index Terms*—SLAM, multi-robot SLAM, localization.**

I. Introduction

LIGHT Detection and Ranging (LiDAR)-based place recognition is a critical step in Simultaneous Localization and Mapping (SLAM) [1]. In real-world environments, the perceptual capabilities of a single agent are limited, making it challenging to work across large-scale scenarios. Collaborative SLAM (CSLAM) systems leverage the advantages of multiple agents to enhance the overall robustness, accuracy, and efficiency [2]. Agents in CSLAM system perform inter-loop closure detection to determine whether they have visited the same place. Similar to intra-loop closure detection in single-agent SLAM, inter-loop closure detection relies on the distance between feature descriptors to retrieve the most relevant loop candidate.

Received 8 May 2025; accepted 8 August 2025. Date of publication 19 August 2025; date of current version 29 August 2025. This letter was recommended for publication by Associate Editor X. Chen and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grant 42371439, in part by the National Key R&D Program of China under Grant 2024YFB3908900, and in part by the Fundamental Research Funds for the Central Universities under Grant 2042025kf0085. *(Corresponding authors: Qingwu Hu; Mingyao Ai.)*

The authors are with the School of Remote Sensing and Information Engineering, Wuhan University, and Hubei Luojia Laboratory, Wuhan 430079, China (e-mail: duanxz@whu.edu.cn; huqw@whu.edu.cn; aimingyao@whu.edu.cn; pengcheng.zhao@whu.edu.cn; ljy_whu_2012@whu.edu.cn).

Our method has been open-sourced and can be accessed at: https://github.com/duanxz0127/dvmm.

Digital Object Identifier 10.1109/LRA.2025.3600141

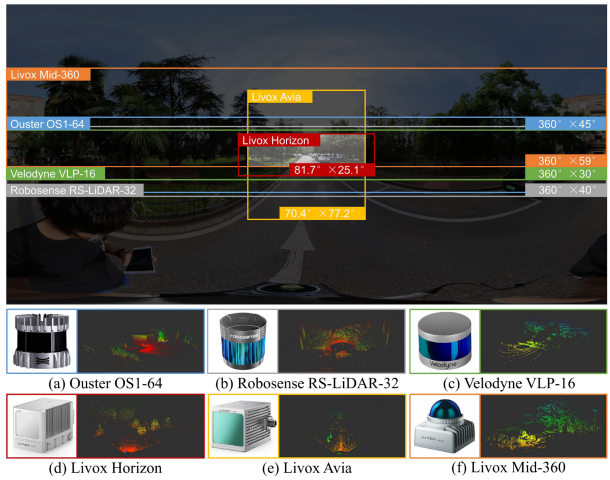


Fig. 1. Multi-modal LiDARs and their FOV differences.

With the rapid advancement of sensor technology, various types of LiDAR have continuously evolved, transitioning from early spinning LiDARs to the currently popular solid-state LiDARs [3], [4], [5], [6]. Multi-modal LiDARs exhibit differences in scanning density, range, mounting height, horizontal field of view (HFOV), vertical field of view (VFOV), and other configurations. Fig. 1 illustrates the differences in field of view (FOV) among six LiDARs observing the same scene. These variations make multi-modal LiDARs suitable for diverse operational scenarios. For instance, indoor applications prioritize complete spatial coverage, whereas outdoor tasks favor LiDARs with long-distance echo returns. Therefore, cross-scene CSLAM across indoor and outdoor environments necessitates the use of different LiDARs to leverage their respective advantages and ensure reliable perception under varying conditions.

Current CSLAM systems typically reuse intra-loop closure descriptors for inter-loop closure detection [7], [8], [9], [10], [11], [12], [13]. Although these descriptors perform well with data from the same sensor, they struggle with the inherent differences across multi-modal LiDARs. To address this, we propose DVMM, a novel dual-view global descriptor for online place recognition. It mitigates the differences in scanning density, range, mounting height, HFOV, and VFOV across multi-modal LiDARs, thus enabling modality-invariant online place recognition.

The main contributions of this work are the following:

- We propose a unified online place recognition method for multi-modal LiDARs. It extracts shared features from

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

TABLE I
COMPARISON OF PLACE RECOGNITION METHODS IN TERMS OF THEIR ROBUSTNESS TO MULTI-MODAL LIDAR DIFFERENCES

Method	Density	Range	Height	HFOV	VFOV
STD [14]	✓	✓	✓	✗	✗
BTC [15]	✗	✓	✓	✗	✗
Segmatch [16]	✓	✓	✓	✓	✗
Segmap [17]	✓	✓	✓	✓	✗
LoGG3D-Net [18]	✓	✓	✓	✗	✗
SeqLPD [19]	✓	✓	✓	✗	✗
Minkloc3D [20]	✓	✓	✓	✗	✗
NDT-Transformer [21]	✓	✓	✓	✗	✗
Scan Context (SC) [22]	✓	✓	✗	✗	✗
LiDAR-Iris [23]	✓	✓	✗	✗	✗
BEVPlace [24]	✗	✓	✓	✗	✗
RING [25] / RING++ [26]	✓	✓	✓	✗	✓
OverlapTransformer (OT) [27]	✓	✗	✗	✗	✗
SeqOT [28]	✓	✗	✗	✗	✗
SOLiD [29]	✗	✓	✗	✗	✗
CVTNet [30]	✓	✗	✗	✗	✗
Ours	✓	✓	✓	✓	✓

✓: robust to the LiDAR difference; ✗: not robust.

diverse LiDAR data and performs retrieval in a coarse-to-fine manner. This work makes one of the first attempts to achieve modality-invariant online place recognition across different LiDARs.

- We introduce a dual-view combination descriptor, comprising an azimuthal descriptor for coarse loop closure detection and a cross-section descriptor for fine verification. Both descriptors are designed to be compatible with multi-modal LiDARs.
- Experiments on public and real-world datasets show that the proposed method outperforms state-of-the-art (SOTA) approaches in multi-modal LiDARs place recognition. We further integrate the descriptor into a CSLAM framework to demonstrate its practical value.

II. RELATED WORK

LiDAR-based place recognition methods can be broadly classified into 3D feature representation and 2D feature representation approaches. Both categories are susceptible to inherent differences among multi-modal LiDARs, as summarized in Table I. We now review these methods and analyze their sensitivity to these sensor-induced differences.

A. 3D Feature Representation Methods

Some existing works utilize raw point clouds or their voxelized and segmented forms as inputs for 3D feature extraction. These methods focus on local features and are generally insensitive to LiDAR scan range and height. For instance, STD [14] and its improved variant, BTC [15], rely on local triangles formed by geometric keypoints. However, variations in point cloud density and FOV can significantly affect the distribution of keypoints, introducing ambiguities in triangle-based matching. Additionally, BTC encodes triangle vertices via vertical projection, making it particularly sensitive to VFOVs. SegMatch [16] and SegMap [17] are segment-based methods that leverage richer information than individual points, enhancing robustness to HFOV variations. However, differences in VFOV can influence the characteristics of each segment, introducing uncertainty in place recognition. Deep learning-based methods use various network architectures to extract features from raw point clouds [18], [19], voxels [20], or NDT cells [21], and aggregate local features into global descriptors. Feature aggregation methods, such as NetVLAD [31], are affected by variations in both HFOV and VFOV, resulting in inconsistencies in the global descriptors generated from multi-modal LiDARs. Despite their potential for improved robustness, learning-based methods are constrained by the availability of adequate training data.

B. 2D Feature Representation Methods

Although 3D feature representation methods yield reasonable results, most of them cannot operate online. As an alternative, 2D representation methods reduce data dimensionality by extracting features from point cloud projections. Some methods adopt the bird's-eye-view (BEV) representation by projecting raw point clouds onto polar grids [22], [23] or Cartesian grids [24], [25], [26]. Due to variations in HFOV, these methods yield descriptors with inconsistent information content across different LiDARs, resulting in reduced robustness to HFOV changes. Additionally, the features selected for bin encoding largely determine the method's sensitivity to LiDAR differences. For instance, BEVPlace [24] uses point density as the feature, making it sensitive to density variations, whereas RING [25] and RING++ [26] employ binary occupancy to construct descriptors, rendering them more robust to VFOV differences. Another category of methods, such as OT [27] and SeqOT [28], projects point clouds along the laser beam emission direction. These methods are also susceptible to range and FOV variations, resulting in inconsistent descriptor representations. In addition, the 2D representation is highly sensitive to viewpoint changes, making it vulnerable to variations in LiDAR mounting height.

The methods most similar to ours are SOLiD [29] and CVTNet [30], both of which build descriptors from more than one view. However, SOLiD encodes bins using point density and is thus sensitive to density variations, while CVTNet lacks range filtering and is susceptible to range differences. Moreover, neither method considers the height and FOV variations, consequently limiting their capability for cross-modal LiDAR place recognition.

III. METHOD

This section details our multi-modal LiDAR online place recognition approach, termed DVMM, shown in Fig. 2. The proposed pipeline comprises three stages: data pre-processing (Section III-A), descriptor generation (Section III-B), and place recognition (Section III-C). The outputs include the detected loop closures and a four-degree-of-freedom (4-DoF) relative pose estimation.

A. Pre-Processing

We adopt submaps as the fundamental units for place recognition. Each submap is constructed by accumulating certain recent LiDAR scans (e.g., consecutive n_f frames). Following this, range filtering and downsampling (with voxel size, ΔL) are applied to maintain consistency in range and density across

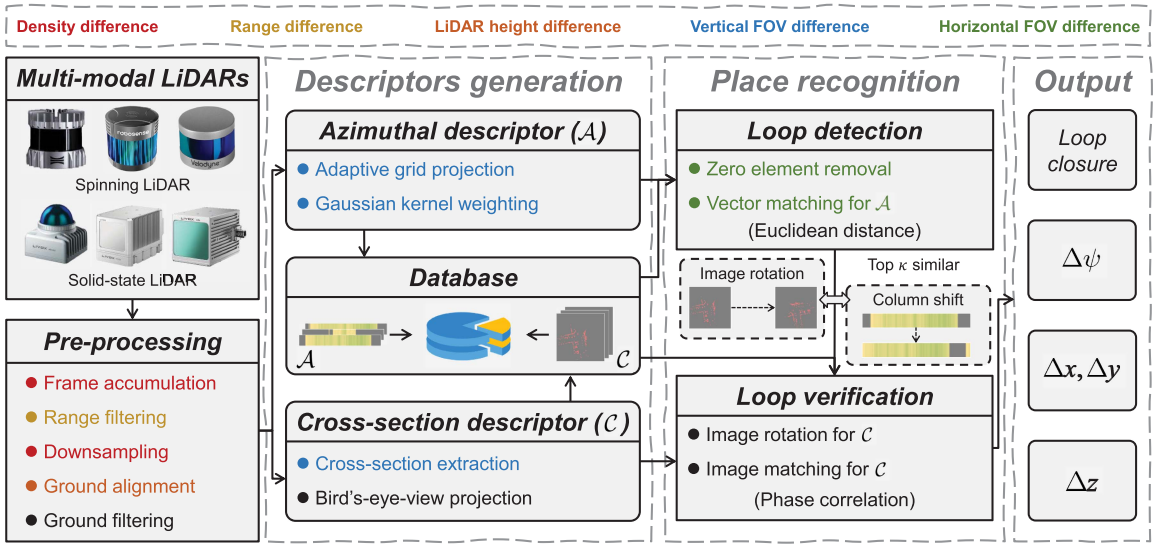


Fig. 2. Overall pipeline of DVMM. Different font colors highlight the solutions addressing various types of differences across multi-modal LiDARs.

submaps, yielding a submap $\mathcal{M}$ with a fixed range $[R_{\min}, R_{\max}]$. The ground alignment is performed to compensate for subtle roll and pitch variations, as well as to correct for height differences among LiDARs mounted on different agents. Specifically, the ground parameters $\boldsymbol{f} = [\mathbf{n}^\top, d]^\top$ is estimated using a simple random sample consensus (RANSAC) [32] method, where $\mathbf{n}$ is the normal vector and d is the distance from the origin to the ground plane. The estimated ground $\boldsymbol{f}$ is transformed into a reference ground parameter $\boldsymbol{f}_s = [\mathbf{n}_s^\top, d_s]^\top$ via a transformation $\mathbf{T} \in SE(3)$, where $\mathbf{n}_s = [0, 0, 1]$ and $d_s = 0$. The submap $\mathcal{M}$ is also subjected to the same transformation to achieve ground alignment. After the transformation, the ground plane is aligned to the reference plane $z = 0$, which serves as the common height baseline across all submaps. Given the ground distances d^Q and d^R of the query and reference point clouds, the height difference is $\Delta z = d^R - d^Q$. Finally, ground points are filtered out to retain only structurally significant non-ground points for descriptor generation.

B. Descriptors Generation

1) Azimuthal Descriptor: The VFOV overlap of multi-modal LiDARs mainly occurs around an elevation angle of $0°$, with almost no valid scans near $\pm 90°$. Consequently, we prioritize the central region in the vertical direction. Drawing inspiration from the Mercator projection in cartography [33], we propose a grid projection method that adaptively adjusts the vertical resolution based on the elevation angle, thereby increasing the coverage of the overlapping region. Each point $\mathbf{p}_j \in \mathcal{M}$ is parameterized in spherical coordinates (r_j, θ_j, ϕ_j), where r_j, $\theta_j \in [0, 2\pi)$, and $\phi_j \in (-\pi/2, \pi/2]$ denote the range, azimuth, and elevation angle, respectively.

To generate the azimuthal descriptor, we proceed in three steps. First, the adaptive grid projection partitions the unit sphere into $2M^s \times N^s$ blocks along elevation and azimuth angles, respectively. Each block represents an azimuth angle interval

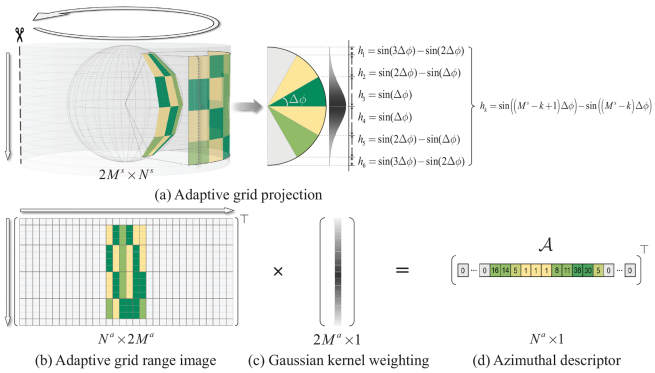


Fig. 3. Schematic of azimuthal descriptor generation. In this case, $M^s = 3$, $M^a = 8$, $N^s = N^a = 32$, the block count is 192, and the bin count is 512.

$\Delta\theta = 2\pi/N^s$ and an elevation angle interval $\Delta\phi = \pi/2M^s$. As shown in Fig. 3(a), a virtual cylinder is placed outside the unit sphere, and the blocks on the sphere's surface are horizontally projected onto the lateral side of the cylinder. The cylinder is then unfolded into an image with $2M^s \times N^s$ blocks of unequal sizes. Following this, the image is partitioned into $2M^a \times N^a$ bins, where N^a is equal to N^s and M^a is determined by $\Delta\phi$. To compute M^a, the height of each block is first calculated as:

$$h_k = \sin\left((M^s - k + 1)\,\Delta\phi\right) - \sin\left((M^s - k)\,\Delta\phi\right), \quad (1)$$

where $k \in \{1, 2, \ldots, 2M^s\}$ is the row index of blocks.

Second, all block heights are normalized by h_1 and summed to determine the updated row count as:

$$2M^a = \sum_{k=1}^{2M^s} \mathrm{Int}\left[\frac{h_k}{h_1}\right], \quad (2)$$

where $\mathrm{Int}[\cdot]$ denotes the floor function. Given the updated row count, we partition the VFOV into $2M^a$ vertical bins, with each bin's elevation angle boundaries defined as:

$$\left(\sin^{-1}\left(1 - \frac{m}{M^a}\right), \sin^{-1}\left(1 - \frac{m-1}{M^a}\right)\right], \quad (3)$$

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

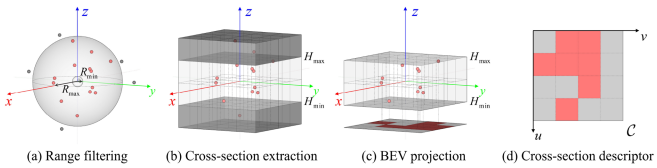


Fig. 4. Schematic of cross-section descriptor generation.

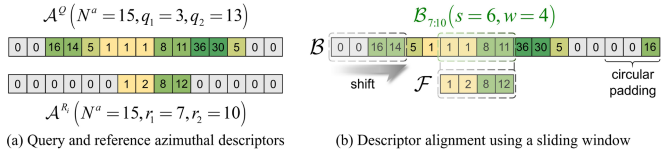


Fig. 5. An example of azimuthal descriptor matching.

where $m \in \{1, 2, \ldots, 2M^a\}$ is the row index of bins. Based on the boundaries defined in (3), we map each 3D point to a specific bin according to its elevation and azimuth angles. This results in the row and column indices of the adaptive grid range image as:

$$\begin{pmatrix} m_j^a \\ n_j^a \end{pmatrix} = \begin{pmatrix} \text{Int}\left[(1 - \sin\phi_j) \times M^a\right] + 1 \\ \text{Int}\left[\frac{\theta_j}{2\pi} \times N^a\right] + 1 \end{pmatrix}. \tag{4}$$

The adaptive grid projection partitions the space into uniform horizontal intervals and non-uniform vertical intervals. The resolution is highest near the 0° elevation angle and progressively decreases toward $\pm 90^\circ$. This ensures adequate preservation of features within the overlapping VFOV region. For each bin, the mean range of the contained points is computed and stored, thereby generating the adaptive grid range image as illustrated in Fig. 3(b).

Third, the adaptive grid range image is used as input for the Gaussian kernel weighting shown in Fig. 3(c). This operation performs elevation-wise compression on the range image via matrix multiplication and emphasizes the overlapping VFOV region by assigning it greater weight. As a result, a compact azimuthal descriptor $\mathcal{A} = [a_1, a_2, \ldots, a_{N^a}]^\top \in \mathbb{R}^{N^a}$ is obtained, as illustrated in Fig. 3(d). Each element a_n represents the weighted sum of bin values in its corresponding column:

$$a_n = \sum_{m=1}^{2M^a} \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(m-\mu)^2}{2\sigma^2}\right) \times \mathcal{I}_{m,n}^a, \tag{5}$$

where $\mathcal{I}_{m,n}^a, m \in \{1, 2, \ldots, 2\,M^a\}, n \in \{1, 2, \ldots, N^a\}$ denotes the bin value at the m-th row and n-th column of the range image, μ represents the Gaussian kernel mean, set to M^a to emphasize the central region, and the kernel width σ is set to αM^a, where α is a scaling factor.

2) Cross-Section Descriptor: The cross-section descriptor is a binary representation based on BEV occupancy. Unlike RING [25] and RING++ [26], which utilize all scan points, our method selectively projects points within a height range to generate the descriptor. Specifically, each point $\mathbf{p}_j \in \mathcal{M}$ is represented using Cartesian coordinates (x_j, y_j, z_j). For the points within the scan range $[R_{\min}, R_{\max}]$ in Fig. 4(a), the cross-section description pipeline begins with extracting points within the height range $[H_{\min}, H_{\max}]$ of the 3D voxelized space, as illustrated in Fig. 4(b). We set $H_{\min}$ to 2.0 m and $H_{\max}$ to 5.0 m to exclude most moving objects on the ground (e.g., pedestrians and cars), and mitigate VFOV-induced occupancy ambiguity. These points are then projected vertically onto the $x-y$ plane, as shown in Fig. 4(c). An $M^c \times N^c$ binary image is generated based on whether each bin is occupied, which serves as the cross-section descriptor $\mathcal{C}$ in Fig. 4(d). Generally, M^c and N^c are set to equal powers of two to ensure compatibility with the subsequent Fast Fourier Transform (FFT) model. The row and column indices of the cross-section descriptor image are:

$$\begin{pmatrix} m_j^c \\ n_j^c \end{pmatrix} = \begin{pmatrix} \text{Int}\left[\frac{x_j + R_{\max}}{2R_{\max}} \times (M^c - 1)\right] + 1 \\ \text{Int}\left[\frac{y_j + R_{\max}}{2R_{\max}} \times (N^c - 1)\right] + 1 \end{pmatrix}. \tag{6}$$

C. Place Recognition

1) Loop Closure Detection: Given a query azimuthal descriptor $\mathcal{A}^Q$ and a set of reference azimuthal descriptors $\{\mathcal{A}^{R_i}\}$, the coarse loop closure detection process aims to determine a revisit based on the distance between each $\mathcal{A}^Q$ and $\mathcal{A}^{R_i}$. For multi-modal LiDARs, this problem is particularly challenging since their HFOV differences lead to variations in the number of zero-value bins in the azimuthal descriptors, rendering traditional similarity metrics such as cosine distance or Euclidean distance not directly applicable. To address this, we incorporate only non-zero bins into the distance calculation. For a pair of azimuthal descriptors shown in Fig. 5(a), we first compute the indices q_1 and r_1 of the first non-zero bins, as well as q_2 and r_2 of the last non-zero bins in $\mathcal{A}^Q$ and $\mathcal{A}^{R_i}$, respectively. Two vectors, including a background $\mathcal{B}$ and a foreground $\mathcal{F}$, are then constructed for distance calculation:

$$\begin{cases} \mathcal{B} = \left[\mathcal{A}^Q, \mathcal{A}^Q_{1:r_2-r_1}\right] & \mathcal{F} = \left[\mathcal{A}^{R_i}_{r_1:r_2}\right], \quad q_1 < r_1, \\ \mathcal{B} = \left[\mathcal{A}^{R_i}, \mathcal{A}^{R_i}_{1:q_2-q_1}\right] & \mathcal{F} = \left[\mathcal{A}^Q_{q_1:q_2}\right], \quad q_1 \geq r_1. \end{cases} \tag{7}$$

where $[A, B]$ denotes the concatenation of vectors A and B. Equation (7) extracts non-zero bins from the narrower HFOV descriptor as the foreground $\mathcal{F}$ and applies circular padding to the wider HFOV descriptor to form the background $\mathcal{B}$ with a full 360° HFOV. This guarantees that the length of $\mathcal{B}$ (either $N^a + r_2 - r_1$ or $N^a + q_2 - q_1$) is always greater than that of $\mathcal{F}$ (either $r_2 - r_1 + 1$ or $q_2 - q_1 + 1$), regardless of the query and reference descriptors. Here, we denote the length of $\mathcal{F}$ as w. To achieve alignment, a sliding window of length w is shifted over the background $\mathcal{B}$. As the window moves by s units, the cropped background segment is denoted as $\mathcal{B}_{s+1:s+w}$. As illustrated in Fig. 5(b), the distance between $\mathcal{A}^{R_i}$ and $\mathcal{A}^Q$, and the best shift s^* are calculated as:

$$D^a\left(\mathcal{A}^{R_i}, \mathcal{A}^Q\right) = \min_{s \in \{0,1,\ldots,N^a-1\}} \left\|\mathcal{B}_{s+1:s+w} - \mathcal{F}\right\|_2, \tag{8}$$

$$s^* = \underset{s \in \{0,1,\ldots,N^a-1\}}{\operatorname{argmin}} \left\|\mathcal{B}_{s+1:s+w} - \mathcal{F}\right\|_2. \tag{9}$$

Once s^* is determined, a coarse yaw alignment is applied to the two point clouds. The rotation angle $\Delta\psi$ from the query to

the reference is:

$$\Delta\psi = \begin{cases} (r_1 - (s^* + 1))\,\dfrac{2\pi}{N^a}, & q_1 < r_1, \\ ((s^* + 1) - q_1)\,\dfrac{2\pi}{N^a}, & q_1 \geq r_1. \end{cases} \tag{10}$$

For each $\mathcal{A}^Q$, the top-κ nearest reference descriptors are selected as candidates for verification.

2) Loop Closure Verification: The azimuthal descriptor encodes the 3D scene into a 1D vector. This compression may cause perceptual aliasing. To fix this, the cross-section descriptor is used to verify the top-ranked references.

Let $L = \{l_1, l_2, \ldots, l_\kappa\}$ denote the index set of the top-κ references, and $\{\mathcal{C}^{R_l}\}_{l\in L}$ their corresponding cross-section descriptors. We firstly rotate the query cross-section descriptor $\mathcal{C}^Q$ by the yaw angle $\Delta\psi$ to achieve a rough alignment with $\mathcal{C}^{R_l}$. The FFT is then applied to convert spatial-domain translations into frequency-domain phase shifts. The cross-power spectrum is computed to extract phase differences, and the translation peak location $(\Delta u, \Delta v)$ is identified via inverse FFT. The aligned query descriptor $\mathcal{C}^{Q_a}$ is obtained after rotation and translation. Its distance to $\mathcal{C}^{R_l}$ is:

$$D^c\left(\mathcal{C}^{R_l}, \mathcal{C}^{Q_a}\right) = 1 - \frac{\left\|\mathcal{C}^{R_l} \wedge \mathcal{C}^{Q_a}\right\|_0}{\left\|\mathcal{C}^{R_l}\right\|_0}, \tag{11}$$

where $\wedge$ denotes the logical AND operation and $\|\cdot\|_0$ is the L_0-norm of the matrix. The nearest candidate satisfying an acceptance threshold τ is selected as the final loop closure:

$$l^* = \arg\min_{l\in L} D^c\left(\mathcal{C}^{R_l}, \mathcal{C}^{Q_a}\right), \quad \text{s.t. } D^c < \tau. \tag{12}$$

The horizontal translation of the query point cloud is:

$$\Delta x = \Delta u_{l^*} \times \frac{2R_{\max}}{M^c}, \quad \Delta y = \Delta v_{l^*} \times \frac{2R_{\max}}{N^c}. \tag{13}$$

where $(\Delta u_{l^*}, \Delta v_{l^*})$ represents the image offset between the query and the l^*-th reference.

With the optimal candidate detected, the 4-DoF relative pose estimation $(\Delta x, \Delta y, \Delta z, \Delta\psi)$ is simultaneously derived. Additional verification techniques, such as iterative closest point (ICP) [34] or pairwise consistency maximization (PCM) [35], can be followed to reinforce the accuracy.

IV. Experiments and Results

In this section, we evaluate the performance of DVMM through experiments on both public and real-world datasets. All experiments are conducted on a desktop computer running Ubuntu 20.04 LTS. The setup includes an Intel Core i7-13700F CPU and 32 GB of RAM.

A. Experiments Setup

DVMM is evaluated on the kth_night_01, kth_day_06, tuhh_day_02, and tuhh_night_09 sequences from the MCD dataset [3], the Road03 and Forest02 sequences from the TIERS dataset [4], and three real-world sequences collected using the setup shown in Fig. 6. The configurations of the multi-modal LiDARs are detailed in Table II. In the following, we use an en dash to indicate the query and reference pair. For example, "H–O" means using Livox Horizon as the query and Ouster OS1-64 as the reference.

(a) Livox Mid-360 (b) Robosense RS-LiDAR-32 (c) Livox Horizon

Fig. 6. Multi-modal LiDAR data acquisition platforms.

TABLE II
Multi-Modal LiDARs Used in the Experiment

LiDAR (Abbreviation)	Type	Rings	Range	HFOV	VFOV	Used in Datasets
Velodyne VLP-16 (V)	spinning	16	100m	360°	30°	TIERS
Robosense RS-LiDAR-32 (R)	spinning	32	200m	360°	40°	Self-made
Ouster OS1-64 (O)	spinning	64	120m	360°	45°	TIERS, MCD
Livox Avia (A)	solid-state	N/A	450m	70.4°	77.2°	TIERS
Livox Horizon (H)	solid-state	N/A	260m	81.7°	25.1°	TIERS, Self-made
Livox Mid-70 (M70)	solid-state	N/A	260m	70.4°	70.4°	MCD
Livox Mid-360 (M360)	solid-state	N/A	70m	360°	59°	Self-made

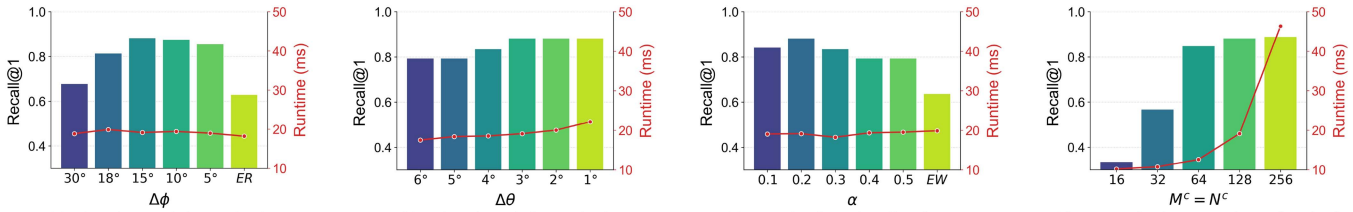
We employ seven comparative descriptors, including five handcrafted methods: STD [14], BTC [15], SOLiD [29], SC [22], and LiDAR-Iris [23], as well as two learning-based methods: OT [27] and CVTNet [30]. The learning-based methods are tested using the author-released pre-trained models. All methods adopt submaps with $n_f = 10$ as input. When applicable, we apply the same range limits ($R_{\min} = 2.0$ m and $R_{\max} = 50.0$ m) and downsampling resolution ($\Delta L = 0.1$ m), while keeping other parameters at default. A detection is considered a true positive (TP) if the distance between the ground truth pose of the query and the optimal candidate is below a preset threshold (5.0 m in this study). Otherwise, it is counted as a false positive (FP).

B. Parameter Settings

In this section, parameter studies are performed on the Road03 sequence with Livox Horizon as the query and Ouster OS1-64 as the reference, as shown in Fig. 7.

The parameters $\Delta\phi$ and $\Delta\theta$ control the size of the range image. We evaluate $\Delta\phi$ values of $[30°, 18°, 15°, 10°, 5°]$, corresponding to $2M^a = [12, 36, 54, 124, 504]$. Fig. 7(a) shows that the runtime remains almost unchanged, thus $\Delta\phi = 15°$ is selected because it yields the highest Recall@1. An ablation study using 60 equally divided rows (denoted as *ER*) consistently underperforms the adaptive grid range image, highlighting the superiority of the proposed projection method. Similarly, we evaluate $\Delta\theta$ with values of $[6°, 5°, 4°, 3°, 2°, 1°]$, corresponding to $N^a = [60, 72, 90, 120, 180, 360]$. As in Fig. 7(b), increasing the number of columns improves the descriptor's discriminative capability, with a marginal increase in runtime. Considering the trade-off between performance and efficiency, $\Delta\theta = 3°$ is recommended in this letter.

The factor α determines the Gaussian kernel width and is evaluated with values of $[0.1, 0.2, 0.3, 0.4, 0.5]$. It is set to 0.2 to achieve the highest Recall@1in Fig. 7(c). An ablation study uses equal-weight averaging within each column (denoted as



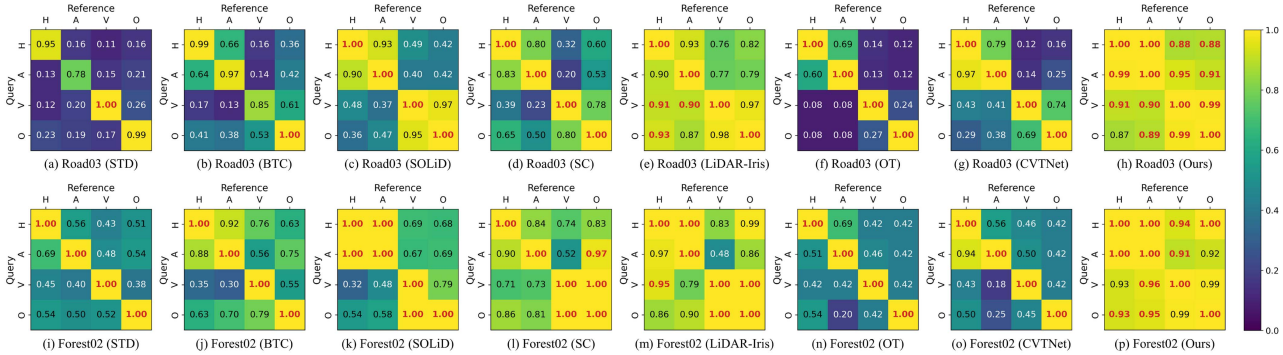
(a) Elevation angle interval $\Delta\phi$ (b) Azimuth angle interval $\Delta\theta$ (c) Gaussian kernel scaling factor α (d) Row and column number M^c and N^c

Fig. 7. Parameter Settings. In (a), "*ER*" denotes equal elevation angle resolution; and in (c), "*EW*" denotes equal weighting.

TABLE III
COMPARISON OF RECALL@1 ACROSS DIFFERENT LIDAR MODALITIES IN THE MCD DATASET

Sequence	LiDAR pair	STD	BTC	SOLiD	SC	LiDAR-Iris	OT	CVTNet	Ours
kth_night_01	O–M70	0.02	0.17	0.04	0.01	0.14	0.01	0.05	**0.62**
	M70–O	0.04	0.15	0.04	0.01	0.12	0.02	0.04	**0.60**
	O–O	0.99	0.72	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	M70–M70	0.99	0.70	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
kth_day_06	O–M70	0.02	0.15	0.04	0.01	0.29	0.01	0.04	**0.66**
	M70–O	0.03	0.10	0.06	0.01	0.09	0.01	0.04	**0.65**
	O–O	0.99	0.66	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	M70–M70	0.99	0.71	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
tuhh_day_02	O–M70	0.04	0.11	0.03	0.02	0.41	0.03	0.05	**0.62**
	M70–O	0.04	0.15	0.08	0.03	0.25	0.01	0.02	**0.62**
	O–O	**1.00**	0.84	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	M70–M70	0.97	0.63	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
tuhh_night_09	O–M70	0.02	0.15	0.10	0.05	0.55	0.15	0.15	**0.72**
	M70–O	0.07	0.16	0.22	0.04	0.60	0.03	0.10	**0.70**
	O–O	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	M70–M70	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**

The highest Recall@1 values are highlighted in bold.



(a) Road03 (STD) (b) Road03 (BTC) (c) Road03 (SOLiD) (d) Road03 (SC) (e) Road03 (LiDAR-Iris) (f) Road03 (OT) (g) Road03 (CVTNet) (h) Road03 (Ours)

(i) Forest02 (STD) (j) Forest02 (BTC) (k) Forest02 (SOLiD) (l) Forest02 (SC) (m) Forest02 (LiDAR-Iris) (n) Forest02 (OT) (o) Forest02 (CVTNet) (p) Forest02 (Ours)

Fig. 8. Comparison of Recall@1 across different LiDAR modalities in the TIERS dataset. The highest Recall@1 values are highlighted in red.

EW), yet yields significantly worse results than Gaussian kernel weighting.

The parameters M^c and N^c define the resolution of the cross-section descriptor. Larger values enable better loop verification, but also increase computational cost. Based on Fig. 7(d), we select $M^c = N^c = 128$ as a balanced choice.

In all subsequent experiments, DVMM follows the parameter settings described above. The only dataset-specific setting is the number of candidates passed to verification: we set $\kappa = 50$ for TIERS and Self-made, and $\kappa = 100$ for MCD due to its larger scale.

C. Place Recognition Evaluation

1) Public Datasets: In each public sequence, LiDARs are mounted on the same agent with synchronized data acquisition, ensuring one-to-one correspondence between queries and references. We use Recall@1 to evaluate the loop closure retrieval capability, as shown in Table III and Fig. 8.

The MCD dataset yields lower Recall@1 than TIERS due to its longer trajectories. On both datasets, our DVMM outperforms existing methods in most cases. Specifically, BTC and STD rely on triangle-based descriptors constructed from keypoints. However, differences in scanning patterns lead to inconsistent submap coverage, which undermines their reliability in multi-modal scenarios. SOLiD, specifically designed for narrow FOVs, performs well on LiDAR pairs with similar scanning patterns but degrades substantially when the FOVs differ significantly. The BEV projection-based SC and LiDAR-Iris encounter similar challenges. SC achieves yaw invariance via azimuth-wise compression but overlooks HFOV differences across LiDARs. Owing to the use of Fourier transforms for yaw

TABLE IV
AVERAGE TRANSFORMATION ERRORS ON PUBLIC DATASETS

Sequence	DVMM		DVMM + T-ICP	
	RE (°)	TE (m)	RE (°)	TE (m)
Road03	4.90	1.40	0.96	0.21
Forest02	4.77	1.13	0.71	0.14
kth_night_01	3.17	0.95	0.88	0.10
kth_day_06	3.24	1.02	0.76	0.14
tuhh_day_02	2.13	0.72	0.61	0.13
tuhh_night_09	2.04	0.57	0.53	0.09

invariance, LiDAR-Iris ranks second to our DVMM. This frequency-domain method remains robust to HFOVs by identifying spectral peaks, as long as the point clouds share enough feature overlap. For learning-based methods, both achieve higher accuracy on similar LiDAR pairs than on cross-modal ones. CVTNet outperforms OT by leveraging both range images and BEVs, whereas OT relies solely on the former. DVMM employs tailored strategies in data filtering, projection, and encoding, thus enhancing robustness across diverse LiDAR pairs.

For each identified loop closure, we further compute the average transformation error [26], as listed in Table IV. The ground truth pose is derived from the agent's poses and the extrinsic calibration between LiDARs. DVMM achieves average rotation and translation errors below 5° and 1.5 m, indicating the estimated relative pose can provide a reliable initial guess for ICP-based refinement. We apply Trimmed-ICP (T-ICP) [34] for 6-DoF refinement, which further reduces errors to less than 1° and 0.3 m, respectively.

2) Field Test: The field test presents a more challenging loop closure detection task, where each query from different agents needs to determine whether a loop closure exists based on a preset distance threshold. We use the precision–recall curve to evaluate the overall detection performance on the Self-made dataset, as illustrated in Fig. 9.

STD exhibit consistently poor performance across most LiDAR pairs. Although its improved variant, BTC, achieves better results, the improvement remains limited. Similarly, SOLiD struggles to handle such complex tasks and fails to retrieve sufficient TPs. SC and LiDAR-Iris show unstable performance, yielding acceptable accuracy in certain scenarios but significantly deteriorating in others. The performance of learning-based methods is also unsatisfactory, as their effectiveness relies on sufficient training data, which is not available in our limited dataset. Our DVMM consistently achieves SOTA performance across all datasets. By explicitly accounting for variations in FOV and LiDAR mounting height, DVMM exhibits strong robustness in real-world scenarios and shows a reliable capability to distinguish between true and false loop closures.

We also integrate DVMM into a CSLAM system, DCL-SLAM [9], which incorporates PCM [35] for consistency verification of detected loop closures. The point clouds are shown in Fig. 10. Livox Mid-360 offers broad coverage in indoor scenarios, whereas Livox Horizon excels in long-range scanning for urban scenarios. That is to say, LiDARs can effectively exploit their advantages in varying scenarios.

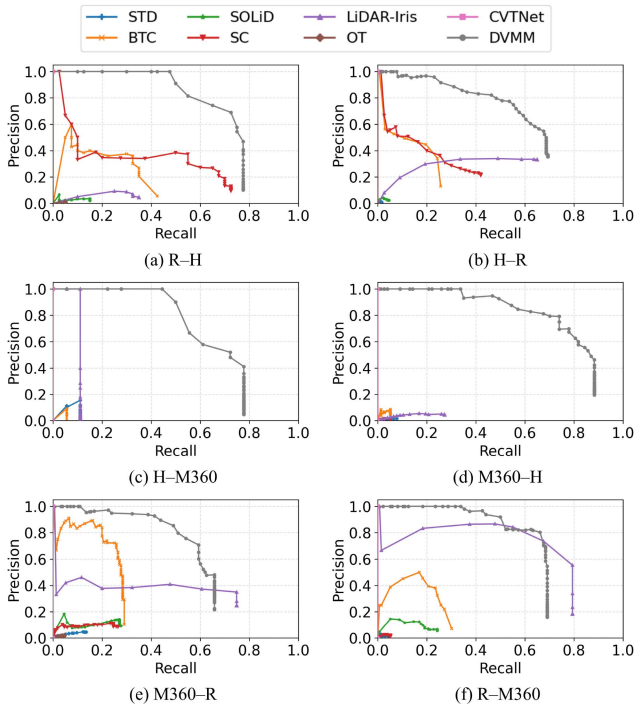


Fig. 9. Precision–recall curve on the Self-made dataset.

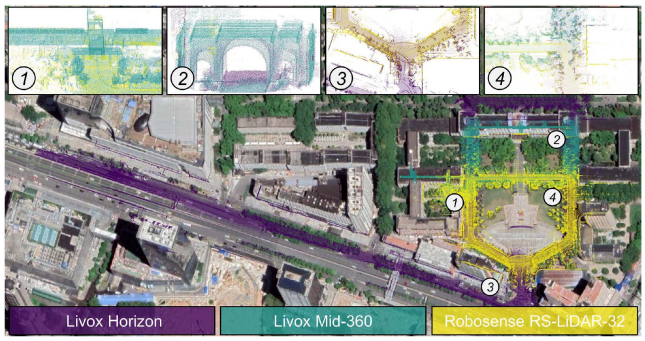


Fig. 10. Mapping results of DVMM deployed in CSLAM.

TABLE V
RUNTIME ANALYSIS ON TIERS ROAD03 SEQUENCE (MS)

Method	Pre-processing	Description	Retrieval	Total
STD	**0.71**	7.81	23.48	32.00
BTC	0.94	12.25	7.85	21.04
SOLiD	13.90	5.09	**0.32**	**19.31**
SC	1.34	6.24	12.82	21.40
LiDAR-Iris	0.75	2.73	272.88	276.36
Ours	20.99	**1.84**	8.87	31.70

The minimum runtime is highlighted in bold.

D. Runtime Analysis

This section compares the runtime of handcrafted descriptors, as listed in Table V. The results show that DVMM achieves efficiency comparable to STD, while being slightly slower than BTC, SOLiD, and SC. DVMM exhibits the highest description efficiency by removing irrelevant points during the preprocessing stage. This stage involves operations such as ground estimation and filtering, thus accounting for the majority of the

overall runtime. Notably, the efficiency of DVMM could be improved in a full SLAM system, where ground information is often available from other modules.

V. Conclusion

This letter introduces DVMM, which is tailored for online place recognition with multi-modal LiDARs. We systematically analyze point cloud differences across LiDAR modalities and categorize them into five types: density, range, LiDAR height, VFOV, and HFOV. DVMM addresses these differences through a structured pipeline comprising pre-processing, adaptive grid projection, Gaussian kernel weighting, and BEV-based cross-section extraction, yielding a modality-invariant representation. Experiments on both public and real-world datasets show that DVMM consistently outperforms SOTA descriptors. Moreover, by integrating DVMM into a CSLAM system, we demonstrate its capability for cross-agent data fusion and real-world applicability.

References

[1] Y. Zhang, P. Shi, and J. Li, "LiDAR-Based place recognition for autonomous driving: A survey," *ACM Comput. Surveys*, vol. 57, no. 4, pp. 1–36, Apr. 2025.

[2] P.-Y. Lajoie, B. Ramtoula, F. Wu, and G. Beltrame, "Towards collaborative simultaneous localization and mapping: A survey of the current research landscape," *Field Robot.*, vol. 2, no. 1, pp. 971–1000, Mar. 2022.

[3] T.-M. Nguyen et al., "MCD: Diverse large-scale multi-campus dataset for robot perception," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, WA, USA, 2024, pp. 22304–22313.

[4] Q. Li, Y. Yu, J. P. Queralta, and T. Westerlund, "Multi-modal lidar dataset for benchmarking general-purpose localization and mapping algorithms," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Kyoto, Japan, 2022, pp. 3837–3844.

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

[6] J. Kim, H. Kim, S. Jeong, Y. Shin, and Y. Cho, "DiTer++: Diverse Terrain and Multi-modal Dataset for Multi-Robot SLAM in Multi-session Environments," 2024, *arXiv:2412.05839*.

[7] G. Kim and A. Kim, "LT-mapper: A modular framework for LiDAR-based lifelong mapping," in *Proc. Int. Conf. Robot. Automat.*, Philadelphia, PA, USA, 2022, pp. 7995–8002.

[8] 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.

[9] S. Zhong, Y. Qi, Z. Chen, J. Wu, H. Chen, and M. Liu, "DCL-SLAM: A distributed collaborative LiDAR SLAM framework for a robotic swarm," *IEEE Sensors J.*, vol. 24, no. 4, pp. 4786–4797, Feb. 2024.

[10] H. Kim et al., "SKiD-SLAM: Robust, lightweight, and distributed multi-Robot LiDAR SLAM in resource-constrained field environments," 2025, *arXiv:2505.08230*.

[11] H. Wei et al., "Large-scale multi-session point-cloud map merging," *IEEE Robot. Automat. Lett.*, vol. 10, no. 1, pp. 88–95, Jan. 2025.

[12] H. Gil, D. Lee, G. Kim, and A. Kim, "Ephemerality meets LiDAR-based lifelong mapping," 2025, *arXiv:2502.13452*.

[13] G. Kang, H. Kim, B. Choi, S. Jeong, Y.-S. Shin, and Y. Cho, "Uni-Mapper: Unified mapping framework for multi-modal LiDARs in complex and dynamic environments," *IEEE Trans. Intell. Veh.*, early access, Jun. 27, 2025, doi: 10.1109/TIV.2025.35835511.

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

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

[16] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena, "SegMatch: Segment based place recognition in 3D point clouds," in *Proc. IEEE Int. Conf. Robot. Automat.*, Singapore, 2017, pp. 5266–5272.

[17] R. Dubé et al., "SegMap: Segment-based mapping and localization using data-driven descriptors," *Int. J. Robot. Res.*, vol. 39, no. 2/3, pp. 339–355, Mar. 2020.

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

[19] Z. Liu et al., "SeqLPD: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1218–1223.

[20] J. Komorowski, "MinkLoc3D: Point cloud based large-scale place recognition," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Waikoloa, HI, USA, 2021, pp. 1789–1798.

[21] Z. Zhou et al., "NDT-Transformer: Large-scale 3D point cloud localisation using the normal distribution transform representation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 5654–5660.

[22] 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.*, Madrid, Spain, 2018, pp. 4802–4809.

[23] Y. Wang, Z. Sun, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "LiDAR Iris for loop-closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Las Vegas, NV, USA, 2020, pp. 5769–5775.

[24] L. Luo et al., "BEVPlace: Learning LiDAR-based place recognition using bird's eye view images," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 8666–8675.

[25] 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.*, Kyoto, Japan, 2022, pp. 2778–2785.

[26] 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.

[27] 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.

[28] J. Ma, X. Chen, J. Xu, and G. Xiong, "SeqOT: A spatial–temporal transformer network for place recognition using sequential LiDAR data," *IEEE Trans. Ind. Electron.*, vol. 70, no. 8, pp. 8225–8234, Aug. 2023.

[29] H. Kim, J. Choi, T. Sim, G. Kim, and Y. Cho, "Narrowing your FOV with SOLiD: Spatially organized and lightweight global descriptor for FOV-Constrained LiDAR place recognition," *IEEE Robot. Automat. Lett.*, vol. 9, no. 11, pp. 9645–9652, Nov. 2024.

[30] 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. Informat.*, vol. 20, no. 3, pp. 4039–4048, Mar. 2024.

[31] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, Jun. 2018.

[32] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[33] J. P. Snyder, *Flattening the Earth: Two Thousand Years of Map Projections*. Chicago, IL, USA: Univ. Chicago Press, Dec. 1997.

[34] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm," *Image Vis. Comput.*, vol. 23, no. 3, pp. 299–309, Mar. 2005.

[35] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, "Pairwise consistent measurement set maximization for robust multi-robot map merging," in *Proc. IEEE Int. Conf. Robot. Automat.*, Brisbane, QLD, Australia, 2018, pp. 2916–2923.