A New Horizon: Employing Map Clustering Similarity for LiDAR-based Place Recognition

Pengcheng Shi, Yilin Xiao, Wenqing Chen, Jiayuan Li, Yongjun Zhang

Abstract*—Lidar-based Place Recognition (LPR) is crucial for intelligent vehicle navigation. Existing methods generally create LiDAR descriptors for pairwise comparisons or employ prior maps for metric localization but face challenges in computational complexity, limited robustness, and excessive memory overhead. Thus, this paper offers a fresh perspective called Map Clustering Similarity (MCS), improving robustness while reducing memory and remarkably boosting efficiency. We start by treating the ground as potential vehicle locations, i.e., virtual points, and introduce a compact LiDAR descriptor called Occupancy Scan Context (OcSC) to capture environmental occupancy from a bird's-eye view. We then employ the point cloud map, virtual points, and k-means clustering to condense the map data into 4Kb cluster centers. Eventually, we devise a two-phase online search algorithm. In the first phase, we extract the OcSC's ring key from online single-frame data, gauge its resemblance to map cluster centers to derive a cluster descriptor, and search loop candidates using the Spearman loss. In the second phase, we propose an occupancy loss to compare all candidates' OcSC descriptors to find the optimal candidate. Our method introduces a novel framework and merges advantages from existing solutions. Experiments on the KITTI dataset and two self-collected indoor sequences showcase MCS-BF's superior performance over mainstream methods in place recognition recall, F_1 score, and memory consumption. Additionally, MCS successfully balances runtime with accuracy. The source code will be available in https://github.com/ShiPC-AI/MCS.

***Index Terms*—LiDAR-based place recognition, occupation scan context, cluster descriptor, ground segmentation, point cloud map.**

I. Introduction

ROBOTS and autonomous vehicles generally depend on identifying historical locations to reduce cumulative errors of the Simultaneous Localization and Mapping (SLAM) system [1], [2]. Traditional Global Positioning System (GPS) navigation falls short in elevated, alley, and indoor environments [3]–[5]. While camera-based methods excel in indoor loop closure detection, they may degrade in urban and highway scenarios due to sensitivity to light and rapid motion [6], [7]. LiDAR sensors [8]–[11], providing detailed 3D and distance information and performing well in low-light conditions and adverse weather, are increasingly preferred for place recognition.

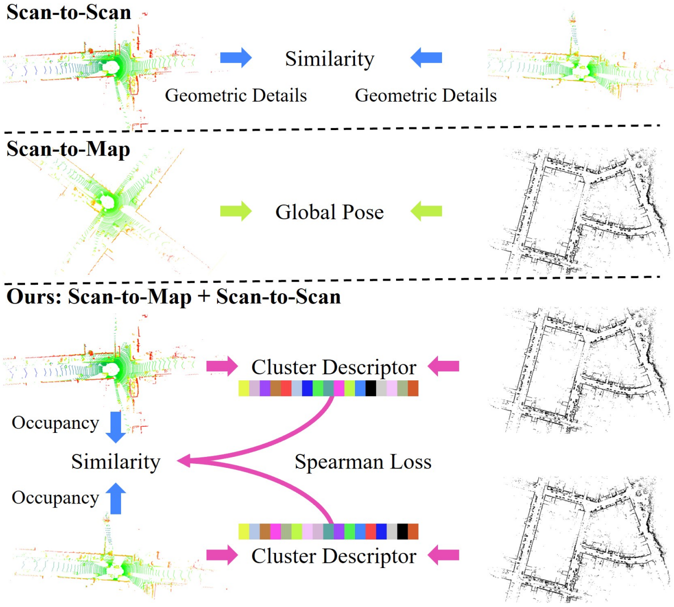


Fig. 1. Differences between our method and current mainstream solutions.

Generally, mainstream LiDAR-based place recognition (LPR) methods fall into two categories. The first integrates real-time sensor measurements with prior maps for scan-to-map matching [12]–[15], which addresses challenges like robot kidnapping and vehicle re-localization. However, the substantial disparities between map and real-time data make matching challenging. The second method conducts scan-to-scan comparisons between current and historical data, which employs matching [16] or similarity evaluation [17] for optimal place retrieval. Enhancing efficiency and stability remains a significant challenge despite considerable research efforts.

To address these issues, we innovatively combine prior maps and inter-frame comparisons to leverage the strengths of both methods. While some works have utilized scan-to-scan and scan-to-map matching in localization [18] and SLAM [19], [20], our method targets the place recognition task. Unlike them, our approach avoids complex point cloud registration, employs the map solely to construct a cluster descriptor, and assesses pairwise similarity through descriptor distance. We represent the point cloud map with a 50*20 matrix to optimize memory usage. Additionally, constructing cluster descriptors takes only 0.15ms, which greatly improves computational

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 42030102 and Grant 42271444, and the Wuhan university-Huawei Geoinformatics Innovation Laboratory Open Fund under Grant TC20210901025-2023-06. (Corresponding author: Jiayuan Li, Yongjun Zhang)

Pengcheng Shi is with the School of Computer Science, Wuhan University, Wuhan 430072, China. (email: shipc_2021@whu.edu.cn)

Yilin Xiao and Wenqing Chen are with the Riemann lab, Huawei Technologies Co., Ltd, Wuhan 430074, China. (email: xiaoyilin@whu.edu.cn, winkicwq@gmail.com)

Jiayuan Li and Yongjun Zhang are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430072, China. (email:ljy_whu_2012@whu.edu.cn, zhangyj@whu.edu.cn)

efficiency.

Figure 1 illustrates the pipeline distinctions between the proposed method (MCS) and mainstream LPR approaches. Despite employing the prior map, we simplify it into multiple cluster centers for constructing a cluster descriptor, thus avoiding intricate scan-to-map matching. This procedure facilitates ultra-lightweight real-time computation. Moreover, our LiDAR descriptor (OcSC) solely judges whether a region is occupied rather than computing geometric details like height, normals, and density. This design remarkably accelerates descriptor construction speed without sacrificing accuracy.

In this paper, we extract ground points from offline scans and treat them as potential vehicle locations (virtual points). We then take virtual points as centers and the prior map as input, generate numerous OcSC descriptors, and perform k-means clustering to compress the map into several cluster centers. Eventually, we devise a two-phase online search algorithm to identify the optimal candidate for the current single-frame point cloud. The main contributions are as follows:

- We present a novel LPR framework merging both scan-to-scan and scan-to-map comparisons.
- We propose a compact LiDAR descriptor (OcSC) to enhance efficiency and reduce memory requirements without compromising accuracy. By leveraging the OcSC descriptor, virtual points, and k-means clustering, we successfully compress the urban-level point cloud map (1.48Gb at 0.1m resolution) to 4Kb.
- We design a fast two-phase search algorithm, which searches loop candidates using a cluster descriptor and Spearman loss, then identifies the optimal candidate using corresponding OcSC descriptors and an occupancy loss. Our MCS-BF yields superior recall and F_1 scores compared to mainstream solutions, while MCS successfully balances runtime with accuracy.

The remainder of this paper is organized as follows. Section II introduces the related work. Section III details the proposed method's pipeline. Section IV presents experimental results on public and self-collected datasets. Section V provides a conclusion of the paper.

II. Related Work

In this section, we provide a concise overview of recent developments in LiDAR-based place recognition. Readers can refer to [21] for a more comprehensive review.

A. Handcrafted Descriptors

Early methods encode geometric features, including distance [22], point normal [23], angle [24], and density [25], for local description, showcasing success in point cloud registration and shape recognition. Later, researchers encapsulate the entire point cloud into a descriptor. Z-Projection [26] and Fast Histogram [27] encode the angle and height information, then assess pairwise similarities utilizing the Sørensen and Wasserstein metric, respectively. M2DP [28], [29] projects the point cloud onto multiple 2D planes, creating the signature vector by computing point density on each plane. It bypasses point normals calculation via projection, enhancing computation efficiency but incurring information loss. Scan Context [17], [30] divides the horizontal space into distinct rings and sectors, creating a 2D height matrix. It achieves favorable results by utilizing ring keys and pairwise similarity scores for nearest neighbor search. Subsequently, several studies enhance performance by integrating intensities [31], spatial binary patterns [32], and frequency descriptions [33]. LiDAR Iris [34] converts point cloud data into an iris image, employs LoG-Gabor filtering and thresholding operations to create binary signatures, and calculates pairwise similarities via the Hamming distance. Handcrafted descriptors, while mature, may produce incorrect results at large lateral offsets.

B. Learning-based descriptors

PointNetVLAD [9] merges a feature extractor network, PointNet [35], with an aggregator, NetVLAD [36], to acquire a 512-dimension descriptor. SOE-Net [37] and PCAN [38] further bolster the descriptor's robustness by integrating attention mechanisms into the network, thus yielding commendable results. SegMap [39] segments the point cloud and trains the descriptor with a combination of classification and reconstruction loss. Locus [8] enhances the descriptor with temporal information, ensuring robustness to viewpoint changes. However, the segmentation relies on rich geometric structures and may degrade in sparse point clouds. HiTPR [40] partitions the point cloud into dense voxels, employing a transformer network [41] to enhance the relevance of local neighbors and global contextual dependencies. While effective in challenging cluttered environments, it demands substantial computational resources. Kim *et al.* [42] introduce a Convolutional Neural Network (CNN) classifier for long-term place recognition, while Semantic Scan Context [43] tackles the translation problem by incorporating semantic information. LPD-Net [44] and vLPD-Net [45] use a Graph Neural Network (GNN) for feature aggregation, efficiently capturing underlying geometric and shape properties. Given environmental disparities like noise and occlusion between real and training datasets, incorporating transfer learning is essential for improved adaptation to specific tasks.

C. Map-based Localization

LiDAR 100 FPS [13] constructs an offline map database to simulate vehicle orientation and position within a point cloud map, employing a binary loss function to enhance localization accuracy. OPD [15] further enhances the online localization efficiency by integrating descriptor similarity and Kalman Filtering. HOPN [12] employs the FAST [46] detector on the BEV image and constructs a global descriptor with 3D normals, demonstrating superior localization in large-scale scenarios. Shi *et al.* [47] use RANSAC [48], Euclidean clustering, and line fitting to extract wall segments from the map and single scans, employing point-to-point and point-to-line distance constraints for vehicle localization. However, it is unsuitable for wall-less outdoor environments like cities or highways. CSSC [49] simultaneously encodes elevation and point density, employing a two-phase similarity estimation and

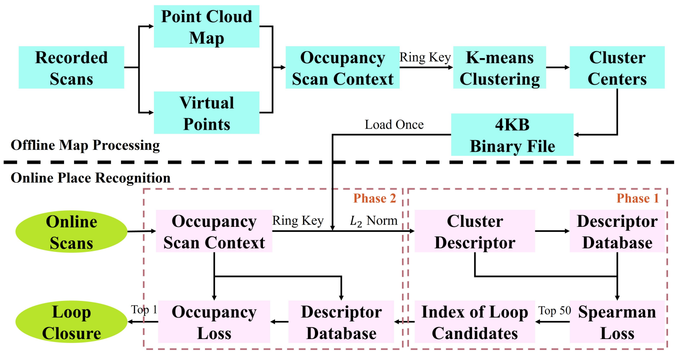


Fig. 2. Framework overview of MCS. It primarily comprises two parts: offline map processing and online place recognition. The system's input is a real-time single-frame LiDAR point cloud, and the output is the data index obtained by loop closure detection.

the Nearest Cluster Distance Ratio (NCDR) to enhance place recognition precision. However, substantial data disparities between single scans and maps led to poor localization outcomes. Recently, researchers have sought to leverage OpenStreetMap, an open-source global map encompassing roads and buildings, for vehicle localization. Several works [50], [51] derive building and road information from OSM and integrate the semantic information into the particle filter framework. Cho *et al.* [14] encode distance information from the building in OSM and LiDAR point cloud into the descriptor, enabling vehicle localization through descriptor comparison. However, these methods struggle in long-straight roads due to the absence of detailed 3D features.

III. Method

A. Overview

Figure 2 showcases our LPR framework. The offline map processing begins by extracting ground points within offline LiDAR scans and registering them to create virtual points. Then, it generates a prior map from the ground-removed point cloud, extracts ring keys for each virtual point, and performs cluster analysis to derive cluster center vectors. Finally, we save these vectors as binary files. The online place recognition module follows a two-phase search process. Upon receiving an online LiDAR scan, it identifies loop closure candidates from the database through the cluster descriptor and Spearman loss. It subsequently evaluates corresponding OcSC descriptors using an occupancy loss to determine the optimal candidate.

B. Offline Map Processing

As shown in Figure 3, the offline map processing module comprises three parts: virtual point creation, prior map generation, and map clustering.

1) Virtual Point Creation: With a recorded point cloud sequence $P_i = \{P_1, P_2, ..., P_n\}$, we employ linefit [52] to extract ground points within each keyframe. We subsequently employ ground truth to transform all ground points to the coordinate system of the first frame. LiDAR SLAM [19] or point cloud registration [53] can be used for relative pose estimation when ground truth is unavailable. To handle the high density and point repetition, we downsample ground points by enforcing a minimum pairwise distance of d_s meters and denote the result as virtual points V. As they derive from ground points, any accessible area within the map can identify its nearest virtual point, characterized by the highest descriptor similarity.

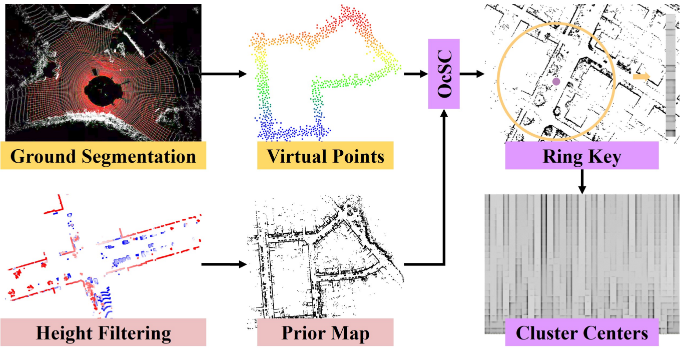


Fig. 3. An illustration of offline map processing. Yellow, pink, and purple represent virtual point creation, prior map generation, and map clustering, respectively. Ground segmentation: white denotes raw LiDAR points and red means segmented ground points. Ring key: the yellow circle marks the range covered by an OcSC descriptor. Cluster centers: we obtain a float-type matrix ranging from 0 to 1, with rows corresponding to OcSC and each column representing a cluster center vector.

2) Prior Map Creation: Given the recorded point cloud sequence $P_i = \{P_1, P_2, ..., P_n\}$, we employ height thresholds (z_{min}, z_{max}) to remove low (ground) and high points (ceilings, bridges) within each keyframe, retaining stable vertical points (buildings, trees). Subsequently, we aggregate all vertical points and get a sparse 3D map by downsampling. Eventually, we ortho-project them onto the xy plane, forming a projected map M_d to depict the horizontal distribution.

3) Occupancy Scan Context: We present a Scan Context [17] variant named Occupancy Scan Context (OcSC). To mitigate ground influence, we employ points within a specified height range to build a binary occupancy matrix before computing the ring key. This approach speeds up descriptor construction, lowering memory requirements while preserving precision.

Figure 4 outlines the space division and calculation process of OcSC $\mathbf{B}$ and its corresponding ring key R. Starting with an input point cloud P, we derive the down-sampled point cloud P^s by specifying the number of points n_s. Subsequently, we compute the binary occupancy matrix's row i and column j indices for each occupied point $p_i(x_i, y_i, z_i) \in P^s$, as follows:

$$\begin{aligned} i &= \left[\frac{L_{max} - \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2}}{L_r} \right] \\ j &= \left[\frac{atan2(y_i - y_c, x_i - x_c) + 2\pi}{\alpha} \right] \end{aligned} \quad (1)$$

where $p_c(x_c, y_c, z_c)$ is the coordinate of center point while we use $p_c = (0, 0, 0)$ for a single LiDAR scan. $L_{max} = N_r * L_r$, L_r and N_r are the length and number of rings, respectively. $\alpha = \frac{2\pi}{N_s}$, N_s is the number of sectors. $[\cdot]$ represents a $floor$ mathematical operation. We calculate the row-wise average of the binary occupancy matrix to obtain the ring key.

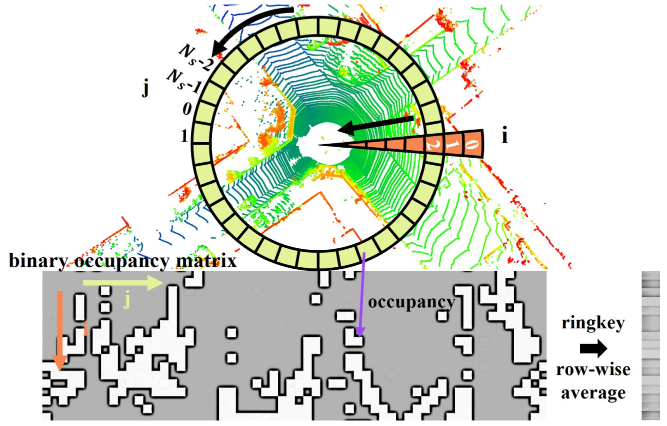


Fig. 4. Occupancy Scan Context and corresponding ring key. The spatial division resembles Scan Context [17], where green indicates a ring and orange signifies a sector. The OcSC descriptor comprises binary elements (0 for unoccupied, 1 for occupied), while the ring key is a non-integer vector.

4) Map Clustering: We begin by using each virtual point $p_i \in V$ as the center point and the projected map M_d as the input point cloud to generate a ring key, thus forming the map ring keys $R^m = \{R_i, i = 1, ..., n_v\}$. n_v is the point number of V. Subsequently, we apply the k-means clustering algorithm to derive the cluster center vectors R^c:

$$R^c = \left\{R_i = \frac{\sum_{j \in idx_i} R^m_j}{n_i}, i = 1, 2, ..., c\right\} \tag{2}$$

where c is a predefined map cluster number. idx_i means the ring key index of the ith cluster. n_i is the ring key number of ith cluster. As depicted in Figure 3, we ultimately transform the point cloud map into a float-type matrix and save it as binary files, occupying mere 4Kb, drastically cutting down the map's memory consumption. During subsequent online place recognition, we load this file just once.

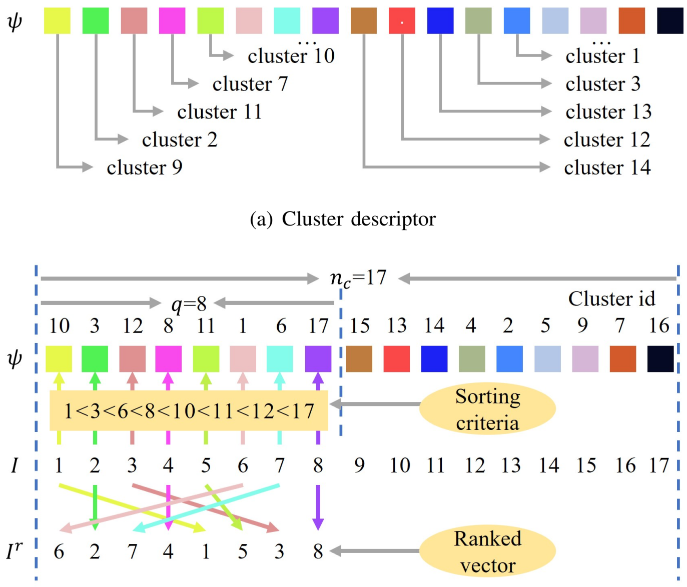
C. Online Place Recognition

We load the offline binary file and introduce a two-phase search framework. In the first phase, we propose a cluster descriptor coupled with a Spearman loss for fast candidate selection. In the second phase, we present a hybrid occupancy loss, combining it with the OcSC descriptor to determine the optimal candidate.

1) First phase search: As depicted in Figure 5, we introduce a cluster descriptor, replacing traditional encoding of intricate geometric details such as height [17], density [28], normals [23], or angles [24] with scan-to-map cluster similarity, thus streamlining calculations.

Upon receiving a real-time LiDAR point cloud scan S, we compute the corresponding OcSC descriptor $\mathbf{B}_s$ and ring key R_s, saving $\mathbf{B}_s$ in the database. We calculate the L_2 norm between R_s and all map cluster centers R^c, then arrange the map clusters in ascending order of distance values, thus yielding the cluster descriptor ψ_s:

$$\begin{cases} \psi_s = (i, j, ..., k), i, j, k \in \{1, 2, ..., c\}, i \neq j \neq k \\ f_2(R_s, R^c_i) < f_2(R_s, R^c_j) < ... < f_2(R_s, R^c_k) \end{cases} \tag{3}$$



(a) Cluster descriptor

(b) Rank vector

Fig. 5. Cluster descriptor and rank vector of cluster Spearman loss. Colored rectangles denote different map clusters.

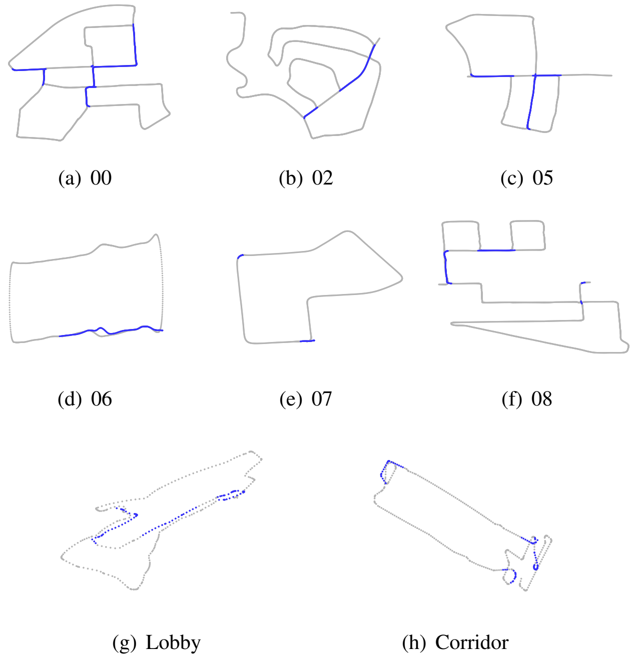
where c is the total number of map clusters. i, j and k are three map cluster indexes. R^c denotes the map cluster centers. f_2 denotes the L_2 norm computation, i.e., $f_2(R_s, R^c_i) = \|R_s - R^c_i\|_2$. As depicted in Figure 5(a), ψ $(1 \times c)$ is an integer vector merely consisting of map cluster indexes. We save the current cluster descriptor ψ_s in the database Ω_ψ.

We then present a cluster Spearman loss, described in Eq. (4) to identify the top 50 candidates Ω_{50} from the cluster descriptor database Ω_ψ. The Spearman's rank correlation coefficient originates from the sociology field, gauging the strength and direction of monotonic relationships between two variables. In this paper, we formulate a cluster-specific Spearman loss by employing the index location as the rank score and the cluster descriptor's corresponding elements as the sorting criteria. Given two cluster descriptors ψ_{src} and ψ_{dst}, we first generate an index vector $I = (0, 1, ..., q-1)$. q signifies selecting the first q columns of the cluster descriptor for computation, satisfying $q \leq c$. As described in Figure 5(b), we sort I in ascending order according to corresponding elements from the cluster descriptor to obtain two ranked vectors I^r_{src} and I^r_{dst}, respectively. We then calculate the following Spearman loss $\mathcal{L}_s$:

$$\mathcal{L}_s(\psi_{src}, \psi_{dst}) = 1 - \left|1 - \frac{6\sum_{i=1}^{q}(I^r_{src}(i) - I^r_{src}(i))^2}{q(q^2-1)}\right| \tag{4}$$

where $I_{src}(i)$ and $I_{dst}(i)$ denote the ith elements of I_{src} and I_{dst}, respectively.

2) Second phase search: The first phase merges cluster descriptor and Spearman loss for fast loop closure candidate identification. Here, we introduce a hybrid occupancy loss $\mathcal{L}_o$, coupled with the OcSC descriptor $\mathbf{B}$, to identify the optimal


(a) 00 (b) 02 (c) 05

(d) 06 (e) 07 (f) 08

(g) Lobby (h) Corridor

Fig. 6. Vehicle trajectories of all experimental datasets. The gray points denote the trajectory and the blue ones denote loop closures.

candidate:

$$v^* = \underset{v\in\Omega_{50}}{\operatorname{argmin}}\, \mathcal{L}_o(\mathbf{B}_u,\mathbf{B}_v)$$

$$\mathcal{L}_o(\mathbf{B}_u,\mathbf{B}_v) = \underset{m\in N_s}{\operatorname{argmin}}\, 1 - \left(\alpha\frac{\phi(\mathbf{B}_u,\mathbf{B}_v^m)}{N_r * N_s} + \beta\frac{\phi(\mathbf{B}_u,\mathbf{B}_v^m)}{N_i}\right) \tag{5}$$

where u is the current query scan and v is a candidate one. $\mathbf{B}_v^m$ means a m column-shifted matrix similar to [17]. The function ϕ counts the elements in $\mathbf{B}_u$ and $\mathbf{B}_v^m$ where corresponding values are 1. N_s is the sector number in the OcSC descriptor parameter. $\alpha + \beta = 1.0$ and N_i represents the total count of occupied elements in the binary matrix $\mathbf{B}_u$.

The occupancy loss consists of two components. $\frac{\phi(\mathbf{B}_u,\mathbf{B}_v^k)}{N_r * N_s}$ focuses on the quantity weight. In areas with ample vegetation, like rural roads, α ensures higher scores for candidate data by factoring in the greater total occupancy quantity. $\frac{\phi(\mathbf{B}_u,\mathbf{B}_v^m)}{N_i}$ describes the similarity ratio weight. In narrow or symmetrical spaces, β ensures two data receive higher scores based on their ratio, even with a smaller occupancy total quantity. At $\alpha = 0$, our focus is solely on quantity, while the emphasis shifts entirely to similarity ratio at $\alpha = 1$. This hybrid loss design partially overcomes our method's sensitivity to threshold parameters. Eventually, users can define an acceptance threshold to decide if the best match forms a loop.

IV. Experiments

A. Datasets and Implementation

1) Datasets: We evaluate our approach on the well-known KITTI [54] odometry and two self-collected indoor sequences. Figure 6 shows vehicle trajectories, while Table I summarizes dataset details.

TABLE I
Details of experimental datasets. Lob represents the lobby and Cor denotes the corridor.

	KITTI						Indoor	
	00	02	05	06	07	08	Lob	Cor
Num. of scans	4541	4661	2761	1101	1101	4071	353	466
Num. of loops	882	329	557	272	147	406	88	92
Num. of clusters	50	50	50	50	50	50	50	50

- Outdoor datasets: The KITTI dataset encompasses 11 LiDAR sequences across urban, rural, and highway settings. The recording platform comprises a vehicle equipped with a stereo camera system, a Velodyne HDL-64E LiDAR, and an OXTS RT 3003 localization system. We choose six sequences (00, 02, 05, 06, 07, and 08). If the pose distance between the query and candidate data is under eight meters, we classify it as a true loop.
- Indoor datasets: We utilize a TurtleBot equipped with a RoboSense RS-LiDAR-16 to collect two indoor sequences, i.e., lobby and corridor. Given the robot's slow pace, we select keyframes at 0.1m intervals, with a loop threshold set at three meters. We rely on LOAM [19]-calculated odometry pose as the relative ground truth.

2) Implementation: We select five baseline methods, i.e., Fast Histogram [27], M2DP [28], Scan Context [17], LiDAR Iris [34], and CSSC [49]. Our method establishes two pipelines, namely MCS and MCF-BF. MCS employs the full two-phase search for place recognition, while MCF-BF relies solely on the second phase. We implement our method in C++ on Ubuntu 20.04 LTS, running on a computer with an Intel Core i9-10850K CPU.

B. Metrics

We comprehensively validate the proposed method's performance using the following standard metrics:

1) Recall @top-k: It indicates the percentage of accurately recognized places using the top k loop candidates. We evaluate the recall of the top 1-50 candidates in our experiments.

2) F_1 Score: This score measures the harmonic mean of precision (R) and recall (R):

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{6}$$

3) Runtime: We perform a comprehensive time analysis for each component, encompassing descriptor extraction, nearest neighbor search, and the entire sequence.

4) Memory consumption: We examine the memory usage of individual descriptors and the map data.

C. Parameter Experiments

To comprehensively validate the proposed method, we conduct parameter experiments involving OcSC descriptor parameters (L_r, N_r, N_s), virtual point sampling distance (d_s), map cluster number (c), and the cluster number (q) for Spearman loss. We evaluate the recall @top-20 in the first-phase search.

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3360321

TABLE II

RECALL @TOP-5 WITH RESPECT TO DESCRIPTOR RESOLUTIONS IN THE SEQUENCE 06. L_r REPRESENT THE RING'S LENGTH WHILE d_s IS THE SAMPLING DISTANCE OF VIRTUAL POINTS. N_r AND N_s DENOTE THE NUMBER OF RINGS AND SECTORS, RESPECTIVELY. BOLD MARKS THE BEST RESULTS IN EACH COLUMN, WITH UNDERLINE DENOTING THE SECOND-BEST ONES.

L_r	N_r	N_s															
		d_s=1.0m				d_s=2.0m				d_s=3.0m				d_s=4.0m			
		30	60	120	180	30	60	120	180	30	60	120	180	30	60	120	180
2.0	10	0.798	0.835	0.787	0.776	0.820	0.838	0.864	0.831	0.919	0.886	0.871	0.893	0.849	0.879	0.864	0.901
	20	0.890	0.871	0.904	0.901	0.919	0.926	0.952	0.941	0.919	0.930	0.919	0.941	0.938	0.952	0.945	0.938
	30	0.934	$\underline{0.949}$	$\underline{0.941}$	**0.967**	$\underline{0.952}$	0.941	$\underline{0.960}$	**0.971**	0.963	0.967	0.956	0.949	0.963	0.971	0.963	$\underline{0.971}$
	40	0.938	0.938	**0.945**	**0.938**	**0.971**	**0.967**	0.934	0.945	**0.989**	0.963	**0.982**	0.960	0.956	0.978	0.963	0.963
3.0	10	0.860	0.868	0.875	0.846	0.886	0.868	0.886	0.882	0.926	0.923	0.919	0.926	0.912	0.926	0.945	0.934
	20	0.938	0.915	0.923	0.912	0.949	0.930	0.926	0.938	$\underline{0.985}$	0.974	$\underline{0.967}$	$\underline{0.971}$	$\underline{0.978}$	**0.985**	0.963	0.967
	30	$\underline{0.949}$	0.923	0.923	0.919	0.941	0.952	**0.967**	$\underline{0.967}$	0.978	0.967	$\underline{0.967}$	0.967	0.963	0.956	$\underline{0.971}$	0.952
	40	**0.952**	**0.960**	0.930	0.934	0.945	$\underline{0.956}$	0.952	0.963	0.974	0.967	0.963	0.963	0.971	$\underline{0.982}$	0.960	0.963
4.0	10	0.846	0.853	0.864	0.857	0.904	0.875	0.915	0.879	0.908	0.904	0.886	0.926	0.926	0.945	0.945	0.956
	20	0.926	0.915	0.919	0.945	0.941	0.938	0.923	0.945	0.974	0.956	0.963	**0.974**	$\underline{0.978}$	0.963	0.974	**0.974**
	30	0.915	0.893	0.923	0.926	0.949	0.949	0.941	0.941	0.963	**0.982**	0.960	0.967	**0.982**	0.952	0.949	0.949
	40	0.908	0.897	0.904	0.919	0.945	0.945	0.949	$\underline{0.967}$	0.978	$\underline{0.978}$	$\underline{0.967}$	0.963	**0.982**	0.956	0.952	0.967
5.0	10	0.813	0.827	0.831	0.824	0.897	0.912	0.904	0.912	0.949	0.949	0.945	0.926	0.971	0.956	0.960	0.963
	20	0.901	0.901	0.919	0.919	0.956	0.926	0.926	0.941	0.974	0.967	$\underline{0.967}$	0.952	0.971	0.971	0.949	0.941
	30	0.882	0.871	0.875	0.882	0.952	0.938	0.945	**0.971**	0.949	0.956	0.952	0.938	0.971	0.949	0.941	0.963
	40	0.886	0.897	0.890	0.915	0.934	0.952	0.949	0.952	0.967	0.960	$\underline{0.967}$	0.930	0.971	0.971	**0.982**	0.963

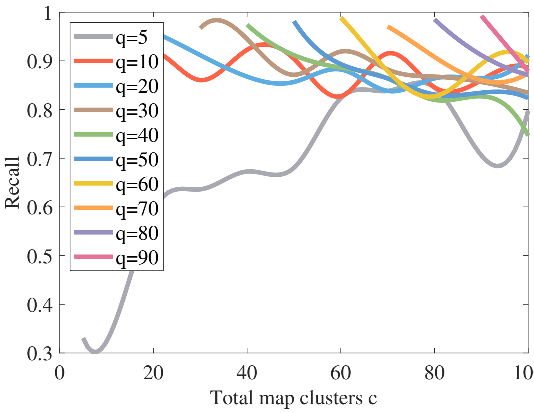


Fig. 7. Recall @top-20 with respect to the cluster number in sequence 06. c is total map clusters, and q denotes those used for cluster Spearman loss.

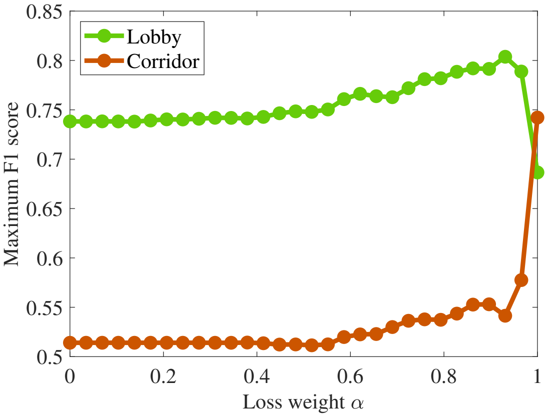


Fig. 8. Maximum F_1 score for various α in the lobby and corridor.

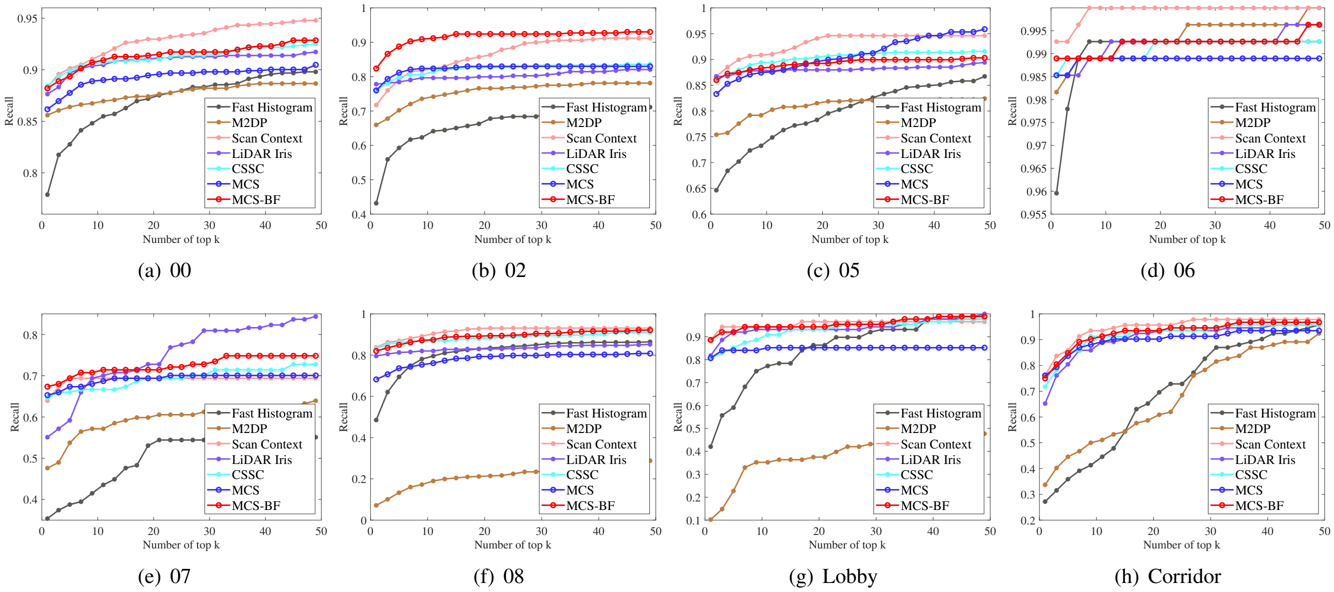
1) Descriptor and virtual points' resolution: Table II displays the recall assessment for various descriptor parameters and virtual point sampling distances (d_s) on KITTI 06. Setting c and q to 50, we perform 256 experiments, varying ring length L_r (2m, 3m, 4m, 5m), ring number N_r (10, 20, 30, 40), sector number N_s (30, 60, 120, 180), and sampling distance d_s (1m, 2m, 3m, 4m). The results indicate our methods obtain the optimal stability at a 2m ring length and a 2m sampling distance. However, we align the descriptor resolution ($L_r = 4m$, $N_r = 20$, $N_s = 60$) with the Scan Context in the following experiments for fair comparisons.

2) Cluster number: Figure 7 describes the evaluation regarding cluster number (q, c) on KITTI 06. As c must exceed q, the curves exhibit varying lengths, and their starting points do not commence at x = 0. Results reveal a positive correlation between q and recall under constant c. However, with a fixed q, fluctuations occur between c and the recall. We create 50 (c) map clusters for all sequences and set $q = c$ in the subsequent experiments.

3) Loss weight: As shown in Figure 8, we choose the corridor and lobby to evaluate the impact of occupancy loss weight (α) on the maximum F_1 score. The sparse 16-line LiDAR and confined indoor space result in a limited total occupancy quantity of OcSC descriptors. As α increases, both curves improve overall. The corridor's loops, enclosed by walls and stairs, exhibit higher descriptor occupancy quantity, notably boosting the F_1 score (0.2) when α approximates 1. The lobby's open space and nearly symmetrical layout lead to similar OcSC occupancy quantity per frame, solely relying on quantitative weights lowering performance (0.15). To maintain a balanced robustness across diverse scenes, we select the inflection point (α=0.85) of the two curves for subsequent experiments.

D. Comparison with SOTA Methods

1) Recall @top-k: Figure 9 depicts recall curves for our two pipelines (MCS, MCS-BF) and five baseline methods on eight sequences. Our curves are comparable to or surpass the



(a) 00 (b) 02 (c) 05 (d) 06

(e) 07 (f) 08 (g) Lobby (h) Corridor

Fig. 9. The recall across different loop candidates.

TABLE III
RECALL AT TOP 1, 5, 20, AND 50. BOLD MARKS THE BEST RESULTS IN EACH COLUMN, WITH UNDERLINE DENOTING THE SECOND-BEST ONES.

	Methods	KITTI 02				KITTI 07				Lobby				Corridor			
		@1	@5	@20	@50	@1	@5	@20	@50	@1	@5	@20	@50	@1	@5	@20	@50
baseline	Fast Histogram	0.432	0.593	0.675	0.717	0.354	0.388	0.537	0.558	0.420	0.591	0.864	0.977	0.272	0.359	0.663	0.957
	M2DP	0.660	0.702	0.766	0.781	0.476	0.537	0.605	0.646	0.102	0.227	0.375	0.477	0.337	0.446	0.598	0.924
	Scan Context	0.717	0.790	0.863	0.912	0.640	**0.694**	0.694	0.694	**0.886**	**0.943**	**0.966**	0.966	**0.761**	**0.859**	**0.957**	**0.978**
	LiDAR Iris	0.778	0.784	0.799	0.821	0.551	0.592	**0.728**	**0.844**	0.818	0.920	0.932	**1.000**	0.652	0.804	0.924	0.957
	CSSC	0.760	0.799	0.827	0.836	0.646	0.660	0.694	0.735	0.807	0.852	0.932	0.977	0.717	0.848	0.935	0.957
ours	MCS	0.760	0.812	0.830	0.830	0.653	0.673	0.694	0.701	0.807	0.841	0.852	0.852	**0.761**	0.837	0.902	0.935
	MCS-BF	**0.824**	**0.888**	**0.924**	**0.930**	**0.673**	**0.694**	0.714	0.748	**0.886**	0.920	0.943	0.989	0.750	0.848	0.935	0.967

baseline methods, showing smoother trends. In Sequence 00, Fast Histogram attains a top-1 recall of 0.779, whereas MCS and MCS-BF achieve higher values of 0.862 and 0.882, respectively. M2DP undergoes significant performance degradation in the corridor. It attains a top-1 recall of only 0.337, which is markedly inferior to MCS (0.761). Scan Context excels at sequence 05, which achieves 0.867 at top 1. MCS (0.833) and MCS-BF (0.860) follow closely behind, but MCS achieves a better performance (0.958) at top 50 than Scan Context (0.946).

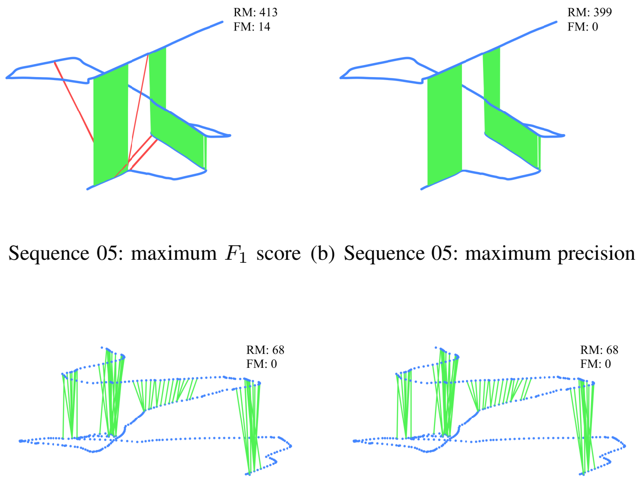
Table III details recall on two outdoor (02, 07) and two indoor sequences (lobby, corridor). MCS-BF excels in sequence 02, boasting a @1 recall approximately 5% higher than the second-ranked LiDAR Iris. In sequence 07, MCS-BF exhibits a 3% improvement over CSSC at @1 and @5, while slightly lower values than LiDAR Iris by 0.01 at @20. MCS-BF and Scan Context achieve an optimal recall of 0.886 at @1 in the lobby. MCS and Scan Context demonstrate optimal performance with a recall @1 of 0.761 in the corridor.

2) Maximum F_1 score: Table IV showcases maximum F1 scores across five sequences, and our methods exhibit superior performance. The confined indoor space features similar heights, which potentially degrades height-based descriptors. Our two pipelines surpass Scan Context, CSSC, and LiDAR Iris by 0.05 in the corridor and achieve the optimal performance (0.854) in the lobby. Likewise, our methods surpass M2DP, Scan Context, and LiDAR Iris by 0.02 in the sequence 00. We secure the second-best performance in sequence 06 (0.983), deviating by merely 0.002 from the top-performing Scan Context. The results demonstrate our methods generally outperform or are on par with mainstream methods.

TABLE IV
MAXIMUM F_1 SCORE. BOLD MARKS THE BEST RESULTS IN EACH COLUMN, WITH UNDERLINE DENOTING THE SECOND-BEST ONES.

	Methods	KITTI			Indoor	
		00	06	07	Lobby	Corridor
baseline	Fast Histogram	0.737	0.894	0.309	0.584	0.475
	M2DP	0.881	0.949	0.464	0.491	0.427
	Scan Context	0.879	**0.985**	0.457	0.848	0.512
	LiDAR Iris	0.879	0.978	0.464	0.852	0.506
	CSSC	0.859	0.978	0.468	0.805	0.506
ours	MCS	0.882	0.978	0.502	0.792	0.550
	MCS-BF	**0.905**	0.983	**0.511**	**0.854**	**0.551**

Fig. 10 depicts our MCS-BF's place recognition results



(a) Sequence 05: maximum F_1 score (b) Sequence 05: maximum precision

(c) Lobby: maximum F_1 score (d) Lobby: maximum precision

Fig. 10. Place recognition in case of maximum F_1 score and maximum precision. Blue denotes trajectory points. Green signifies correct matches, while red denotes incorrect ones. RM indicates correct matches, and FM represents false matches. Our method simultaneously achieves maximum accuracy and F_1 score in the lobby, which renders the place recognition results in (c) and (d) identical.

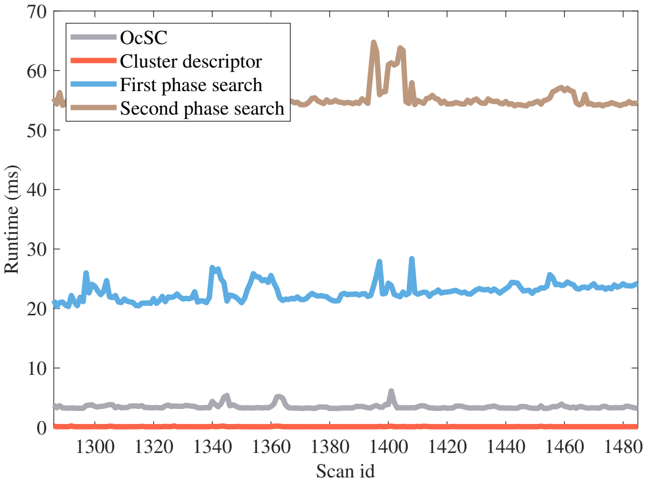


Fig. 11. Detailed runtime analysis of MCS in sequence 05. The average time consumption for OcSC, cluster descriptor computation, first-stage search, and second-stage search is 3.44ms, 0.15ms, 22.80ms, and 56.23ms, respectively.

under two cases: maximum F_1 score and maximum precision. In sequence 05, our method attains 413 correct matches and 14 false matches at the maximum F_1 score. At maximum precision, it accurately identifies 399 correct matches without false matches. Notably, in the lobby, our method simultaneously achieves the maximum F_1 score and maximum precision, identifying 68 correct matches without false matches. These results underscore the efficacy of our method in accurately detecting loops across diverse indoor and outdoor settings.

3) Runtime: We categorize outdoor sequences as short (07), medium (05), and long (08) based on their trajectory length. As the 16-line LiDAR merely yields approximately 28,000 points per frame, we designate the corridor sequence as a sparse type. Table V summarizes the runtime evaluation of each method for descriptor extraction, nearest neighbor search, and the overall sequence computation. As described in Figure 11, we choose 200 loop closure scans from sequence 05 for a detailed frame-by-frame runtime analysis.

Descriptor creation. The Velodyne HDL-64E LiDAR captures roughly 120,000 points per frame within KITTI sequences. Our method demonstrates exceptional speed, producing the OcSC descriptor in 3.4ms—134 times faster than M2DP (456.4ms), around 15 times faster than Scan Context (50.2ms), three times faster than CSSC (9.9ms), and twice as fast as LiDAR Iris (7.9ms). In the sparse corridor, our method (1.6ms) outperforms M2DP (153.1ms) by 96 times, exceeds Scan Context (15.6ms) by ten times, triples the speed of LiDAR Iris (4.2ms), and doubles CSSC's (2.8ms). As illustrated in Figure 11, the construction of our cluster descriptor requires a mere 0.15ms, which enables ultra-fast calculation between real-time scans and maps.

Nearest neighbor search. Nearest neighbor search is the process of identifying the best candidate. M2DP compares multiple 192*1 vectors directly, achieving the best speed of approximately 2ms (KITTI) and 0.36ms (corridor). Scan Context employs a kd tree to accelerate the search, taking 51ms (KITTI) and 43ms (corridor). Our method, CSSC, and LiDAR IRIS avoid tree construction, but we leverage a Spearman loss to expedite the search, making MCS 20 times faster than MCS-BF. MCS takes around 83ms in KITTI, which is 131 times faster than LiDAR Iris (10785.4ms) and 67 times faster than CSSC (5560.5ms).

Overall sequence computation. Table V demonstrates MCS's superior speed in calculating the overall sequences. Our method, leveraging an ultra-lightweight descriptor and clustering-based Spearman loss search, expedites place recognition tasks even without a kd tree. MCS averages 72s on KITTI, surpassing Scan Context and LiDAR Iris by five and 63 times, respectively. In the corridor sequence, our pipelines shine, with MCS finishing in 7.3s—nine times faster than M2DP and CSSC.

4) Memory consumption: Low offline storage memory facilitates efficient data management and transfer, while lightweight running memory enhances computational efficiency. Table VI details the memory usage and data type of MCS's components. Offline denotes disk space used for storing offline map data, while online refers to program memory consumption during real-time operations. In running memory, our binary OcSC descriptor (20×60) only takes 1200 bits (150 bytes). Under equivalent resolution parameters, Scan Context necessitates 4800 bytes, which is 32 times larger than that required by OcSC. Our ring key (20×1) and cluster descriptor (1×50) requires 80 and 200 bytes, respectively.

Table VII summarizes offline memory comparisons between our method and point cloud maps across various downsampling voxel resolutions. Regarding offline memory, we create 50 clusters per sequence, transforming the dense point cloud map into a 50×20 matrix, saved as a binary file. Taking sequence 02 as an example, our method successfully compress a 599m*946m urban-level point cloud map, originally occu-

TABLE V
RUNTIME ANALYSIS. DESC, SEAR, AND TOTA DENOTE THE TIME CONSUMPTION OF DESCRIPTOR CREATION, NEAREST NEIGHBOR SEARCH, AND OVERALL SEQUENCE COMPUTATION. BOLD MARKS THE BEST RESULTS IN EACH COLUMN, WITH UNDERLINE DENOTING THE SECOND-BEST ONES.

	Methods	KITTI 07 (short)			KITTI 05 (medium)			KITTI 08 (long)			Average			Indoor corridor (sparse)		
		Desc (ms)	Sear (ms)	Tota (s)	Desc (ms)	Sear (ms)	Tota (s)	Desc (ms)	Sear (ms)	Tota (s)	Desc (ms)	Sear (ms)	Tota (s)	Desc (ms)	Sear (ms)	Tota (s)
baseline	M2DP	444.2	**1.4**	489.7	466.3	**2.8**	1480.6	458.6	**2.5**	1873.4	456.4	**2.2**	1821.2	153.1	**0.4**	71.4
	Scan Context	43.5	43.9	176.7	54.8	55.8	221.4	52.2	52.3	693.5	50.2	50.7	363.9	15.6	42.8	40.3
	LiDAR Iris	7.8	5864.0	872.0	8.1	13381.9	7479.7	7.8	13110.2	5360.5	7.9	10785.4	4570.7	4.2	1451.9	135.7
	CSSC	9.4	3022.9	467.3	10.2	6935.7	3926.0	10.2	6722.9	2822.3	9.9	5560.5	2405.2	2.8	726.8	68.6
ours	MCS	3.5	72.7	**27.6**	**3.4**	87.1	**90.3**	3.4	87.8	**99.0**	3.4	82.5	**72.3**	1.7	59.5	**7.8**
	MCS-BF	**3.3**	910.1	149.7	**3.4**	2068.4	1193.4	**3.3**	2009.7	875.2	**3.3**	1662.7	739.4	**1.5**	230.7	23.4

TABLE VI
MEMORY ANALYSIS OF OUR OFFLINE STORAGE AND ONLINE RUNNING.

	Offline storage	Online running		
	Map data	OcSC	Ring key	Cluster descriptor
Dimension	50×20	20×60	20×1	1×50
Data type	float	binary	float	integer
Memory	**4** Kb	150 bytes	80 bytes	200 bytes

TABLE VII
OFFLINE MEMORY CONSUMPTION: OURS VS. POINT CLOUD MAPS

Seq.	Raw	Sampling resolution					Ours
		0.1m	0.2m	0.3m	0.4m	0.5m	
00	6.58Gb	1.03Gb	299Mb	136Mb	77.2Mb	49.6Mb	**4**Kb
02	8.04Gb	1.48Gb	429Mb	192Mb	107Mb	68.4Mb	**4**Kb
05	4.04Gb	565Mb	157Mb	71.7Mb	40.8Mb	26.3Mb	**4**Kb
06	1.87Gb	263Mb	72.3Mb	32.7Mb	18.6Mb	12.1Mb	**4**Kb
07	1.35Gb	186Mb	55.3Mb	26.1Mb	15.2Mb	9.99Mb	**4**Kb
08	5.77Gb	1.24Gb	409Mb	196Mb	114Mb	75.1Mb	**4**Kb

pying 1.48Gb (0.1m resolution) and 429Mb (0.2m resolution), into a compact 4Kb file. Our map management solution handles memory issues in high-precision maps, opening new pathways for lightweight robot navigation.

V. CONCLUSION

In this paper, we propose a novel LiDAR-based Place Recognition (LPR) framework called MCS, which combines scan-to-scan and scan-to-map comparisons. We present a lightweight LiDAR descriptor named OcSC aimed at speeding up construction processes and reducing memory usage without sacrificing accuracy. Our map management solution efficiently compresses large-scale maps' memory footprint to just 4Kb, advancing lightweight map navigation research. We also pioneer a cluster descriptor that characterizes the similarity between scans and maps, bypassing heavy matching computations. We devise a two-phase search algorithm that merges cluster descriptor and Speaman loss for fast loop candidate identification and incorporates OcSC with an occupancy loss to determine the best candidate. Experimental results demonstrate superior performance in terms of recall, F_1 score, and especially runtime compared to mainstream methods.

REFERENCES

[1] K. Nielsen and G. Hendeby, "Multi-hypothesis slam for non-static environments with reoccurring landmarks," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 4, pp. 3191–3203, 2023.

[2] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 3, pp. 194–220, 2017.

[3] Y. Gao, H. Jing, M. Dianati, C. M. Hancock, and X. Meng, "Performance analysis of robust cooperative positioning based on gps/uwb integration for connected autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 1, pp. 790–802, 2023.

[4] P. Shi, Q. Ye, and L. Zeng, "A novel indoor structure extraction based on dense point cloud," *ISPRS International Journal of Geo-Information*, vol. 9, no. 11, p. 660, 2020.

[5] A. Chalvatzaras, I. Pratikakis, and A. A. Amanatiadis, "A survey on map-based localization techniques for autonomous vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 2, pp. 1574–1596, 2023.

[6] M. Roth and D. M. Gavrila, "intrapose: Monocular driver 6 dof head pose estimation leveraging camera intrinsics," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 8, pp. 4057–4068, 2023.

[7] E. K. Stathopoulou and F. Remondino, "A survey on conventional and learning-based methods for multi-view stereo," *The Photogrammetric Record*, vol. 38, no. 183, pp. 374–407, 2023.

[8] K. Vidanapathirana, P. Moghadam, B. Harwood, M. Zhao, S. Sridharan, and C. Fookes, "Locus: Lidar-based place recognition using spatiotemporal higher-order pooling," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 5075–5081.

[9] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4470–4479.

[10] M. Wang, R. Liu, J. Yang, X. Lu, J. Yu, and H. Ren, "Traffic sign three-dimensional reconstruction based on point clouds and panoramic images," *The Photogrammetric Record*, vol. 37, no. 177, pp. 87–110, 2022.

[11] R. Liu, M. Wang, G. Hou, W. Wu, C. Zhao, and Q. Ge, "The classification of airborne lidar building point clouds based on multi-scale and multi-level cloth simulation," *The Photogrammetric Record*, 2023.

[12] L. Luo, S.-Y. Cao, Z. Sheng, and H.-L. Shen, "Lidar-based global localization using histogram of orientations of principal normals," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 771–782, 2022.

[13] P. Shi, J. Li, and Y. Zhang, "Lidar localization at 100 fps: A map-aided and template descriptor-based global method," *International Journal of Applied Earth Observation and Geoinformation*, vol. 120, p. 103336, 2023.

[14] Y. Cho, G. Kim, S. Lee, and J.-H. Ryu, "Openstreetmap-based lidar global localization in urban environment without a prior lidar map," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4999–5006, 2022.

[15] P. Shi, J. Li, and Y. Zhang, "A fast lidar place recognition and localization method by fusing local and global search," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 202, pp. 637–651, 2023.

[16] Q. Ye, P. Shi, K. Xu, P. Gui, and S. Zhang, "A novel loop closure detection approach using simplified structure for low-cost lidar," *Sensors*, vol. 20, no. 8, p. 2299, 2020.

[17] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4802–4809.

[18] R. Ren, H. Fu, H. Xue, X. Li, X. Hu, and M. Wu, "Lidar-based robust localization for field autonomous vehicles in off-road environments," *Journal of Field Robotics*, vol. 38, no. 8, pp. 1059–1077, 2021.

[19] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[20] J.-E. Deschaud, "Imls-slam: Scan-to-model matching based on 3d data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 2480–2485.

[21] P. Shi, Y. Zhang, and J. Li, "Lidar-based place recognition for autonomous driving: A survey," *arXiv preprint arXiv:2306.10561*, 2023.

[22] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Aligning point cloud views using persistent feature histograms," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3384–3391.

[23] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[24] S. Salti, F. Tombari, and L. Di Stefano, "Shot: Unique signatures of histograms for surface and texture description," *Computer Vision and Image Understanding*, vol. 125, pp. 251–264, 2014.

[25] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 433–449, 1999.

[26] N. Muhammad and S. Lacroix, "Loop closure detection using small-sized signatures from 3d lidar data," in *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011, pp. 333–338.

[27] T. Röhling, J. Mack, and D. Schulz, "A fast histogram-based similarity measure for detecting loop closures in 3-d lidar data," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 736–741.

[28] L. He, X. Wang, and H. Zhang, "M2dp: A novel 3d point cloud descriptor and its application in loop closure detection," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 231–237.

[29] L. Perdomo, D. Pittol, M. Mantelli, R. Maffei, M. Kolberg, and E. Prestes, "c-m2dp: A fast point cloud descriptor with color information to perform loop closure detection," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, 2019, pp. 1145–1150.

[30] G. Kim, S. Choi, and A. Kim, "Scan context++: Structural place recognition robust to rotation and lateral variations in urban environments," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1856–1874, 2022.

[31] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2095–2101.

[32] F. Ou, Y. Li, and Z. Miao, "Place recognition of large-scale unstructured orchards with attention score maps," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 958–965, 2023.

[33] Y. Fan, X. Du, L. Luo, and J. Shen, "Fresco: Frequency-domain scan context for lidar-based place recognition with translation and rotation invariance," in *2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2022, pp. 576–583.

[34] Y. Wang, Z. Sun, C.-Z. Xu, S. E. Sarma, J. Yang, and H. Kong, "Lidar iris for loop-closure detection," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5769–5775.

[35] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.

[36] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, 2018.

[37] Y. Xia, Y. Xu, S. Li, R. Wang, J. Du, D. Cremers, and U. Stilla, "Soe-net: A self-attention and orientation encoding network for point cloud based place recognition," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 343–11 352.

[38] W. Zhang and C. Xiao, "Pcan: 3d attention map learning using contextual information for point cloud based retrieval," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 12 428–12 437.

[39] R. Dube, A. Cramariuc, D. Dugas, H. Sommer, M. Dymczyk, J. Nieto, R. Siegwart, and C. Cadena, "Segmap: Segment-based mapping and localization using data-driven descriptors," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 339–355, 2020.

[40] Z. Hou, Y. Yan, C. Xu, and H. Kong, "Hitpr: Hierarchical transformer for place recognition in point cloud," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2612–2618.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[42] G. Kim, B. Park, and A. Kim, "1-day learning, 1-year localization: Long-term lidar localization using scan context image," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1948–1955, 2019.

[43] L. Li, X. Kong, X. Zhao, T. Huang, W. Li, F. Wen, H. Zhang, and Y. Liu, "Ssc: Semantic scan context for large-scale place recognition," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2092–2099.

[44] Z. Liu, S. Zhou, C. Suo, P. Yin, W. Chen, H. Wang, H. Li, and Y. Liu, "Lpd-net: 3d point cloud learning for large-scale place recognition and environment analysis," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2831–2840.

[45] Z. Qiao, H. Hu, W. Shi, S. Chen, Z. Liu, and H. Wang, "A registration-aided domain adaptation network for 3d point cloud based place recognition," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1317–1322.

[46] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*. Springer, 2006, pp. 430–443.

[47] P. Shi, Q. Ye, Z. Shaoming, and D. Haifeng, "Localization initialization for multi-beam lidar considering indoor scene feature," *Acta Geodaetica et Cartographica Sinica*, vol. 50, pp. 1594–1604, 2021.

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

[49] D. Xu, J. Liu, Y. Liang, X. Lv, and J. Hyyppä, "A lidar-based single-shot global localization solution using a cross-section shape context descriptor," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 189, pp. 272–288, 2022.

[50] B. Suger and W. Burgard, "Global outer-urban navigation with openstreetmap," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1417–1422.

[51] F. Yan, O. Vysotska, and C. Stachniss, "Global localization on openstreetmap using 4-bit semantic descriptors," in *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 2019, pp. 1–7.

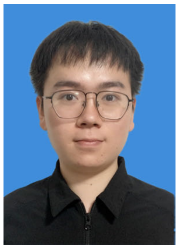
[52] M. Himmelsbach, F. v. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3d point clouds for ground vehicles," in *2010 IEEE Intelligent Vehicles Symposium*, 2010, pp. 560–565.

[53] J. Li, P. Shi, Q. Hu, and Y. Zhang, "Qgore: Quadratic-time guaranteed outlier removal for point cloud registration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 11 136–11 151, 2023.

[54] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.

Pengcheng Shi received the B.S. degree in remote sensing science and technology from Liaoning Technical University, China, in 2018, and the M.S. degree in surveying and mapping engineering from Tongji University, China, in 2021. Currently, he is pursuing the Ph.D. degree with the School of Computer Science, Wuhan University. His research interests include simultaneous localization and mapping (SLAM), LiDAR-based place recognition, map localization, and point cloud registration.

This article has been accepted for publication in IEEE Transactions on Intelligent Vehicles. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TIV.2024.3360321



Yilin Xiao received the B.S. degree in process equipment and control engineering from the Dalian university of Technology, Dalian, China, in 2019, and the M.S. degree in computer science from Wuhan University, wuhan, China, in 2022. He is currently an algorithm engineer with Huawei Technologies Co., Ltd, Wuhan, China. His current research interests include image matching, 3D reconstruction, image retrieval and Slam.

Wenqing Chen received the B.S. degree in the hehai university and M.S. degree in Harbin Institute of technology. He is currently an algorithm engineer with Huawei Technologies Co., Ltd, Wuhan, China. His current research interests include slam, 3d perception and autonomous navigation.

Jiayuan Li received the B.Eng., M.Eng., and Ph.D. degrees from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China. He is currently an Associate Researcher with Wuhan University. He has authored more than 40 peer-reviewed articles in international journals. His research is mainly focused on SLAM, image matching, and point cloud registration. He was awarded the Best Youth Author Award by ISPRS in 2021 and the Talbert Abrams Award by ASPRS in 2018.

Yongjun Zhang received the B.S. degree in geodesy, the M.S. degree in geodesy and surveying engineering, and the Ph.D. degree in geodesy and photography from Wuhan University, Wuhan, China, in 1997, 2000, and 2002, respectively. He is currently the Dean of the School of Remote Sensing and Information Engineering, Wuhan University. He has published more than 180 research articles and one book. He was selected as World's Top 2% Scientists 2022 (career-long impact and single-year impact). His research interests include aerospace and low attitude photogrammetry, image matching, combined block adjustment with multisource data sets, object information extraction and modeling with artificial intelligence, integration of LiDAR point clouds and images, and 3D city model reconstruction. He is the Co-Editor-in-Chief of The Photogrammetric Record.