Scan Context++: Structural Place Recognition Robust to Rotation and Lateral Variations in Urban Environments

Giseop Kim, Sunwook Choi, and Ayoung Kim, *Member, IEEE*

***Abstract*—Place recognition is a key module in robotic navigation. The existing line of studies mostly focuses on visual place recognition to recognize previously visited places solely based on their appearance. In this article, we address structural place recognition by recognizing a place based on structural appearance, namely from range sensors. Extending our previous work on a rotation invariant spatial descriptor, the proposed descriptor completes a generic descriptor robust to both rotation (heading) and translation when roll–pitch motions are not severe. We introduce two subdescriptors and enable topological place retrieval followed by the 1-degree of freedom semimetric localization, thereby bridging the gap between topological place retrieval and metric localization. The proposed method has been evaluated thoroughly in terms of environmental complexity and scale. The source code is available and can easily be integrated into existing light detection and ranging simultaneous localization and mapping.**

***Index Terms*—Localization, place recognition, range sensors.**

I. Introduction

RECOGNIZING a previously visited place is important for various robot missions [e.g., loop detection in simultaneous localization and mapping (SLAM) [1], global localization for a kidnapped robot [2], or multirobot mapping [3]]. Describing a place with a set of compact representations has been tackled in depth within the computer vision and robotics community, yielding many state-of-the-art visual place recognition methods [4]–[7]. In contrary to the flourishing studies on visual place recognition, studies on range sensors are still missing a solid solution to this global localization problem.

Manuscript received July 7, 2021; accepted September 16, 2021. Date of publication November 10, 2021; date of current version June 7, 2022. This work was supported in part by [Localization in changing city] project funded by NAVER LABS Corporation and in part by the National Research Foundation under Grant NRF-2019K2A9A1A06070173. Portions of this work were presented in part at the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2018 [1]. Code will be available at https://github.com/gisbi-kim/scancontext. This paper was recommended for publication by Associate Editor L. Carlone and Editor F. Chaumette upon evaluation of the reviewers' comments. *(Corresponding author: Ayoung Kim.)*

Giseop Kim is with the Department of Civil and Environmental Engineering, KAIST, Daejeon 34141, South Korea (e-mail: paulgkim@kaist.ac.kr).

Sunwook Choi is with the Autonomous Driving Group, NAVER LABS, Gyeonggi-do 13638, South Korea (e-mail: sunwook.choi@naverlabs.com).

Ayoung Kim is with the Department of Mechanical Engineering, SNU, Seoul 08826, South Korea (e-mail: ayoungk@snu.ac.kr).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TRO.2021.3116424.

Digital Object Identifier 10.1109/TRO.2021.3116424

Recent studies have reported [1], [2], [8]–[11] that structural information could be more effective than appearance, particularly within outdoor environments. These studies had attempted to overcome the major bottlenecks resulting from unstructured, unordered, and sparse range sensor data, which make consuming input data harder than pixelated image data. Existing methods have focused on compactly summarizing a place, but they have rarely achieved invariances in structural place recognition.

Our preliminary version of this article presented in [1] tried to establish this compact representation by capturing the highest structural points when the level of roll–pitch disturbances is not severe (e.g., under 10°) such as a wheeled robot or a slow walking hand-held system. This strategy allowed us to achieve robustness for underlying structural variance (e.g., dynamic objects and seasonal changes) for incoming light detection and ranging (LiDAR) measurements. Although our previous scan context showed meaningful performance, the algorithm failed to achieve invariance in the lateral direction and was inefficient using a brute-force search. Overcoming these limitations in [1], we complete the algorithm to include both rotational and lateral robustness, thereby introducing a generic *structural place recognition* for a range sensor. Second, the modified algorithm improved previously brute-force search to use subdescriptors and expedited the process by the order of magnitude. In summary, our new contributions are the following.

1) **Robustness to Lateral/Rotational Changes:** Missing lateral invariance may be a critical issue in an urban environment where lane-level change is inevitable. To resolve this limitation, we generalized the previous descriptor to include both lateral and rotational robustness simultaneously. This is achieved via *scan context augmentation* based on urban road assumption.
2) **Semimetric Localization:** Combining place retrieval and metric localization, our global place recognition method bridges the gap between topological and metric localization. The proposed method provides not only the retrieved map place index but also 1-degree of freedom (DOF; yaw or lateral) initial guess for metric refinement such as iterated closest point (ICP).
3) **Lightweight and Modules Independence:** As a global localizer, the proposed method does not require prior knowledge or any geometric constraints (e.g., odometry). The implementation is lightweight provided in a single

C++ and header pair and readily integrable to existing SLAM framework.

4) **Real-Time Performance on CPU:** By introducing compact summarizing subdescriptors, *keys*, we achieved substantial cost reduction. The *retrieval key* based tree search eliminates naive pixelwise comparison followed by *aligning key* based prealignment. Our method runs in real-time supporting up to 100 Hz (e.g., average 7.4 ms on `KITTI 00` [12]) without requiring GPU.
5) **Extensive Validation:** We evaluate the proposed method across diverse and challenging test scenarios to validate both in-session and multisession scenarios. We note that the existing precision–recall curve may not fully capture the loop-closure performance for SLAM research missing evaluation on the match distribution. We propose to use distribution–recall (DR) curves to measure not only the recalls but also their diversity for the meaningful loop-closure.

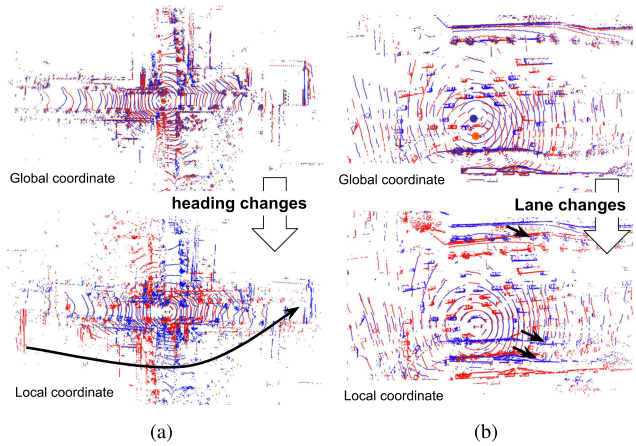


Fig. 1. Sample point cloud undergoing rotational (e.g., reversed revisits) and translational (e.g., lateral lane changes) motion. The red indicates the query scans, and the blue depicts experience in the database. Unlike the view in global coordinates (i.e., world coordinate, top row), the measurement looks different in the local coordinates (i.e., sensor coordinate and bottom row). Also, note that many dynamic objects exist in both scans. (a) Rotational Displacement. (b) Lateral Displacement.

II. Related Works

In this section, we provide a literature review on place recognition in both visual and structural aspects. We briefly review recent place recognition works, focusing on the sensor modality as well as global and local descriptions.

A. Place Recognition for Visual Sensing

For visual recognition, both the local and global aspects of the place summarization were examined. The local description-based methods relied on detecting and describing handcrafted local keypoints (i.e., a small patch) [13], [14]. Using these local descriptors, Bayesian inference [4] or bag-of-words vocabulary tree [7] was applied for place recognition. Cadena *et al.* [15] proposed fusing the bag-of-words and a conditional random field matching of 3-D geometry for a stereo camera system.

Compared to local descriptors, global descriptors are more compact in representation and robust to local noises. The entire image is encapsulated by a single condensed representation (e.g., a fixed-size vector [16], [17] or a downsized image [5]) without maintaining a set of local keypoint descriptors. Similarly, as in local descriptors, recent studies in global descriptors enhanced the performance by exploiting structural information. Oertel *et al.* [11] reported that the use of structural cues when making a global descriptor yields higher performance than appearance-only methods. Mo and Sattar [10] fed reconstructed 3-D sparse points into a LiDAR descriptor pipeline, which outperformed appearance-only based global descriptors.

B. Place Recognition for Range Sensing

1) LiDAR: The early phase of LiDAR-based place recognition focused on 2-D range data [18], [19]. Olson [20], [21] proposed correlative scan matching-based loop-closure detection for 2-D LiDAR. As 3-D LiDAR appeared, 3-D point cloud summarization drew attention. For the initial 3-D LiDAR place recognition methods [22]–[24], local keypoint-based approaches were used, similar to following the early history described above in the visual domain.

A point cloud from a 3-D LiDAR poses challenges in a different aspect. First, the data is unstructured without having a constant and consistent grid density. Second, the data sparsity grows as the range increases, varying the target object density depending on the sensing range. These sensor characteristics make the local descriptions unstable; thus, a coarser summarization unit that is robust to the local noise and inconsistent point density is preferred. M2DP [25] compressed a single LiDAR scan into a global descriptor (i.e., 192D vector) that is robust to noisy input. PointNetVLAD [26] leveraged a learning-based approach to summarize a place into a single vector representation.

However, despite the performance and robustness of global descriptors, one drawback is that they do not secure invariance compared with local-based methods. As reported in [2], these global descriptors were less invariant to the transformation (e.g., heading changes) because transformed local point coordinates may produce different embedding and cause failure in place recognition (Fig. 1). Recently, similar to our approach, a semi-handcrafted heading-invariant feature learning approach named LocNet [27] was proposed. However, compared to LocNet, achieving not only rotational but also translational invariance is required while maintaining the performance of the current state-of-the-art global point cloud descriptors.

In this line of study, a local characteristic such as segment or height was examined. For example, Dubé *et al.* [28], [29] proposed a segment-based global localization method using a handcraft segment descriptor and learned segment embeddings. They recovered a relative transformation between two matched frames through geometric consistency checks, even under severe viewpoint changes such as reverse revisits. Our preliminary work, *scan context* [1], proposed to make a 2-D descriptor based on the height of the surrounding structures. This descriptor obtained rotational invariance and yielded relative yaw as a

by-product. Stemming from this work, some authors [30], [31] tried to simultaneously estimate the relative yaw between two scans and their similarity. Learning-based approaches included semilearned [2], [32] and full learning based [30], [31] methods.

2) Radar: More recently, a long-range perceptible frequency modulated continuous wave (FMCW) radar has been highlighted in robotics applications [33], [34]. Radars provide far longer range and robustness compared to cameras and LiDAR; however, radar place recognition methods are still not mature. Exploiting the image-like format of radar data, some studies leveraged computer vision techniques to describe a radar image at local [35] and global description [36], [37] levels. However, the projection model of the radar image inevitably eliminates height information generating a top-down view. To handle this elevation loss, Hong *et al.* [38] used a LiDAR descriptor M2DP [25] but using the intensity of a pixel in lieu of a point's height. Similarly, [34] showed the feasibility of scan context by replacing the height with the intensity.

III. Requirements for Structural Place Recognition

A. Terminology and Problem Definition

We first define our *place recognition* problem. As a robot traverses an environment, a set of range sensor measurements is streamed with increasing timestamps. We consider every single sensor measurement z_t acquired at a certain spatial location l_t at time t as a *place*. A *map* is a database, a set of all streamed measurements after the time a robot has started a mission. Then, our place recognition can be defined as finding a revisited place within a map for a query place. It is also important to trustfully decide whether there is no revisited place in a map. *Revisitedness* is satisfied for two places, a and b, temporally apart from a certain window size (i.e., $|t_b - t_a| > \delta_t$), if the Euclidean distance between two places' spatial locations is less than a certain threshold (i.e., $|l_b - l_a| < \delta_l$).

To construct such a place recognition system, two submodules are required. The first is description function $f(\cdot)$. To ease handling noisy or heavy raw measurements, a raw measurement z_t is encoded into a more compact form called descriptor $\mathbf{f}_t = f(z_t)$. The second is retrieval that defines a similarity function $\text{sim}(\cdot, \cdot)$ or distance function $D(\cdot, \cdot)$. It takes two descriptors and returns a scalar-value similarity or distance in the descriptor space. Then, the place recognition is reduced to the nearest search problem using the description and similarity functions when a query measurement z and a map are given. One can conclude that two places a and b are the same if the descriptor distance $D(\mathbf{f}_a, \mathbf{f}_b)$ is lower than a threshold τ.

B. Invariance

Most LiDAR place recognition methods [25], [28], [31], [32] have been tested over less complex environments [12], [39] with few dynamic objects or viewpoint changes. The existing research has mostly focused on increasing the discriminability of a descriptor, rather than on defining and overcoming structural diversity. We provide a taxonomical analysis of the potential nuisances for structural place recognition, as shown in Table I.

TABLE I
Required Invariance for Visual and Structural Place Recognition

	Internal Factor			External Factor		Sensor Specification	
Visual	R	T	S	I	W	FOV	
Structral	R	T	SP	D	SV	FOV	NR

R: rotation	I: illumination	FOV: field of view
T: translation	W: weather	NR: number of rays
S: scale	D: dynamic objs	
SP: sparsity	SV: structural variance	

We categorized each invariance in the comparison to the rather widely studied visual place recognition problem for each corresponding invariance type.

1) Internal Factors: The measurement's variation could be derived from a robot itself that we named *internal factors*. This includes rotation, translation, and scale changes of the sensor coordinate mostly induced by ego-motion (R, T, and SP in Table I). Fig. 1 illustrates the sample measurement discrepancy under rotational and translational variance. In terms of scale, the same object looks very different due to the variation in point cloud density caused by the sensing distance.

2) External Factors: Similar to illumination changes (short-term variance) and weather changes (long-term variance) in the visual domain, structures may undergo similar variance in the short-term through occlusions by dynamic objects and in the long-term through permanent structural changes from construction or demolition. This external factor becomes critical as we deploy robots for long-term navigation.

3) Sensor Characteristics: The last factor, sensor characteristics, may be more range-sensor specific. Unlike the highly structured sensor data obtained by cameras, LiDAR point clouds are unstructured, and sensing changes dramatically depending on the sensor's specifications [e.g., range, number of rays, and point cloud resolution depending on field-of-view (FOV)]. Thus, a generic place recognition system should be invariant to sensor specifications.

C. Overview

The proposed method consists of two parts: 1) place description and 2) place recognition. The overall pipeline is illustrated in Fig. 2. The place recognition module consists of place retrieval, semimetric localization, and verification. In the next two sections, we will introduce each module in detail.

IV. Scan Context Descriptor

In this section, we describe a novel spatial descriptor named *scan context descriptor (SCD)*. The pipeline begins with partitioning the raw measurement and projecting them into discretized bins using bird-eye-view (BEV). When dividing into the BEV bins, two types of perpendicular bases (polar and Cartesian) are considered. After partition and coordinate selection, the subset of the measurement is encoded to its associated discretized bin using the bin encoding function. As we present, the invariance of the proposal place recognition module arises from the bin encoding function and the distance function.

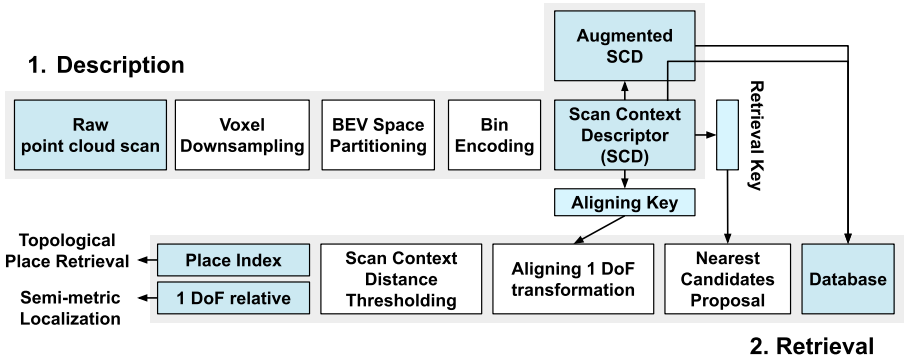


Fig. 2. Overall framework. Given a raw range measurement, the proposed method seeks the corresponding place index from a set of places in the map.

A. Motivation

Our descriptor and search engine were strongly motivated by the revisit patterns in urban environments. We found typical patterns due to the nonholonomic vehicle motion following traffic rules (e.g., lane-keeping). The dominant motion is locally two-dimensional, and the motion occurs with at most two directions that are likely to be disjointed. These typical patterns motivated the choice of two coordinate frames, polar and Cartesian, and the associated matching algorithm.

B. Descriptor Axes and Resolution

We assume that the input is a single scan of 3-D LiDAR. The first phase for generating the descriptor is to partition a downsampled point cloud within a region of interest (ROI). The upper bound of the ROI and the partitioning resolution decide the shape of an SCD. Given the partitioned raw measurements, we project them on 2-D descriptor space; namely, the approach first 1) projects each 3-D point to a 2-D point, 2) parametrizes the 2-D point in polar or Cartesian coordinates, and 3) obtains a scalar value (details in Section IV-C) for each bin by discretizing the 2-D space.

As shown in Fig. 2, we name the horizontal axis the *aligning axis (A-axis)* and the vertical axis the *retrieval axis (R-axis)*. The change along the A-axis corresponds to the columnwise shift; thus, prealignment along the A-axis will allow us to infer a rough metric-level relative pose, overcoming changes in the associated direction. The choice of aligning/retrieval axes determines the type of SCD, as either polar context (PC) or cart context (CC).

1) Polar Coordinates: As introduced in our earlier work [1], the PC adopts polar coordinates using the azimuth θ as the A-axis and the radius r as the R-axis. Because the azimuth is on the A-axis, the PC is robust to rotational variance.

2) Cartesian Coordinates: The CC leverages Cartesian coordinates and uses the lateral direction (y) as the A-axis. The longitudinal direction (or travel direction, x) becomes the R-axis. Naturally, the descriptor is invariant to lateral direction translation.

3) Descriptor Resolution: The resolution of the axes determines the resolution of the descriptor, which is the user parameter of the proposed method. The user parameters are denoted as

$$(\Delta_R, \Delta_A, [R_{\min}, R_{\max}], [A_{\min}, A_{\max}]) \tag{1}$$

where each component indicates an ordered set consisting of the resolution of the R-axis, the resolution of the A-axis, the range of the R-axis, and the range of the A-axis, respectively. Sample parameter sets and their SCD are given in Fig. 3(b). As will be discussed in Section VIII-A, coarse discretization implicitly reduces the influence of dynamic objects, noisy local structures, and computational cost.

4) Independence of the Input Modality: The partitioning is independent of the measurement's distribution or data type (e.g., a BEV image, voxels, or 3-D points). Therefore, our descriptor is generic with regard to any range measurements. The descriptor representation covers not only 3-D point clouds but also other range sensors such as radar [34] by selecting a proper bin encoding function in Section IV-C.

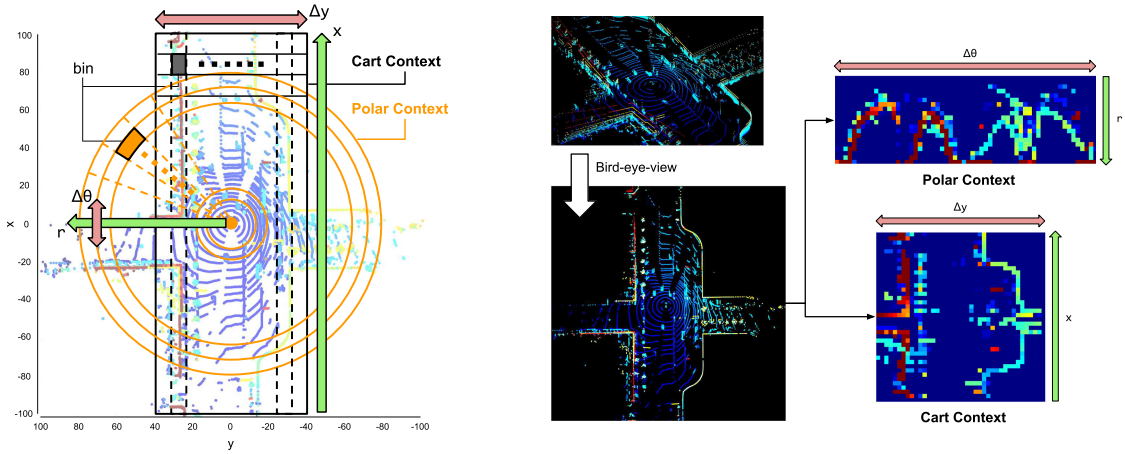
C. Bin Encoding Function

We denote a single disjoint section partitioned by the aligning and retrieval axes as a *bin*. A single bin includes a subset of a robot sensor measurement ($Z_{ij} \in Z$), where the i and j indicate the A-axis and R-axis indexes, respectively. The bin may be empty, $Z_{ij} = \emptyset$, when no range data falls into the bin, in which case we assign a value of 0 to that bin.

For each subset of measurement Z_{ij} for bin (i, j), we assign a representative value using a *bin encoding function* $\psi(\cdot)$. The bin encoding function should be able to encapsulate the subset of the raw data in order to make the descriptor discernable and robust to the nuisances (Table I).

Requirement 1. A *bin encoding function*, $\psi : Z_{ij} \to \mathbb{R}$, is invariant to the internal factors and independent of sensor specifications.

Following our previous work [1], we propose to assign the maximum height of 3-D points within a bin. The intuition behind this selection stems from an urban planning concept called *isovist* [40], [41]. In this concept, the maximally visible structure and its visible volume's polygon shape decide the use of a place and make a place discernable. Focusing on the maximum height instead of structural shape eliminates the sparsity variation caused by the sensing resolution, range, and object size. Notably,



(a) (b)

Fig. 3. (a) Sample point cloud and (b) the associated SCDs. In (a), the yellow and gray color-filled entities depict bins for PC and CC, respectively. The red arrow represents the aligning axis. The green arrow represents the retrieval axis. In (b), each bin color indicates the maximum height in a bin; red is high (e.g., 10 m), blue is low (e.g., 0 m). In (b), the PC (top) has the parameters (20, 60, [0, 80] m, $[0, 360]^\circ$), and the CC (bottom) has the parameters (40, 40, [–100, 100] m, [–40, 40] m).

any other function that meets the above requirement could be used as the encoding function. For the example of an FMCW radar [34], the raw radar intensity value was adopted. Some follow-up studies of our previous work [1] leveraged LiDAR intensity [42], interpolated intensity [34], and difference in the height of 3-D points [10]. We note that heterogeneous LiDAR place recognition in situations where a mapper and a localizer are different (e.g., LiDAR's mounted height varies) is beyond this article's scope because it does not obey the above requirement.

D. Scan Context Descriptor

After the ROI partitioning (Section IV-B) and bin encoding (Section IV-C), each bin contains a representative feature summarizing the data within the bin (i.e., the maximum height for SCD). We accumulate these bin values into a matrix form to complete a 2-D descriptor for a place; the rows and columns of the matrix correspond to the retrieval axis and the aligning axis. The resulting descriptor can be understood as the contour of the skyline of the surrounding structures. Depending on the coordinate selection, we name the resulting 2-D descriptor as *PC* or *CC*.

1) Polar Context: When the polar-coordinate ROI is used, we name the resulting SCD as a *PC*. The PC is designed for rotation-invariant place recognition (e.g., revisit in the reversed direction) because the rotational variation corresponds to columnwise shifts.

2) Cart Context: Similarly, using Cartesian ROI partitioning yields a SCD called a *CC*. In a CC, lateral translation is reflected as columnwise shifts; thus, the CC can handle lateral variation, including a revisit with lane changes.

Each SCD has its own invariance for tackling the internal factors. PC and CC allow one dimension for the A-axis and may be limited when rotation and translation occur simultaneously. To cope with this, we propose hallucinating the R-axis to achieve robustness in both directions (Section V-D).

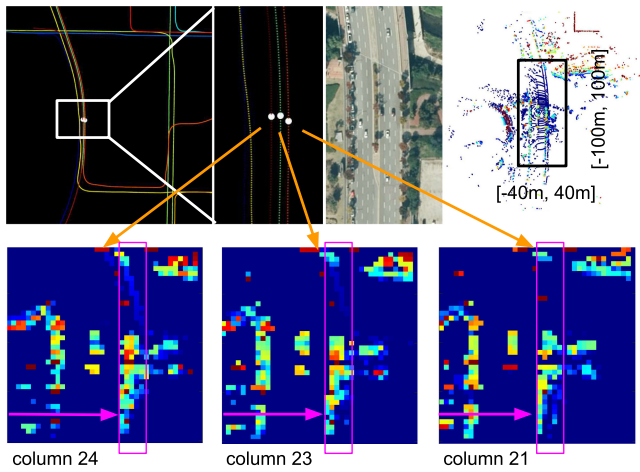


Fig. 4. Three white dots in the top row indicate three sample nodes in the ground-truth trajectory. The vehicle visited the place three times while changing lanes. Below is the sample CC corresponding to each node. Comparing the three sample CCs, the contents are preserved within each column, while only the column orders are shifted among the nodes. The motion-induced change in the descriptor appears as a SCD column order shift in the descriptor space.

E. Distance Between SCDs

Next, we define the proximity between two places by the similarity score of the associated SCD.

1) Alignment Score: As illustrated in Fig. 4, if two SCDs are acquired from the same place, then two descriptors should contain consistent contents within a matrix but may reveal a column order difference. To measure similarity, therefore, we should examine the sum of the columnwise co-occurrences using the cosine similarity between two descriptors. This columnwise comparison is particularly effective for dynamic objects or partial noises. A cosine distance is used to compute a distance between two column vectors, c_Q^j and c_M^j, at the same column

index j. The distance between two descriptors is

$$d(\mathbf{f}_Q, \mathbf{f}_M) = \frac{1}{N_{\mathrm{A}}} \sum_{j=1}^{N_{\mathrm{A}}} \left(1 - \frac{c_Q^j \cdot c_M^j}{\|c_Q^j\| \|c_M^j\|} \right). \tag{2}$$

The subscripts Q and M indicate query and map places, where the descriptor's dimensions are $\mathbf{f} \in \mathbb{R}^{N_R \times N_A}$. In addition, we divide the summation by the number of columns for normalization.

2) Naive Column Alignment: However, the column of the query SCD, $\mathbf{f}_Q$, may be shifted even in the same place (Fig. 4). By simply shifting the order of the query descriptor while the $\mathbf{f}_M$ is fixed, we can calculate distances with all possible column-shifted $\mathbf{f}_Q$ and find the minimum distance. Then, the minimum distance of (2) becomes our desired distance function $D(\cdot, \cdot)$ as

$$\begin{aligned} D(\mathbf{f}_Q, \mathbf{f}_M) &= \min_{n \in [N_{\mathrm{A}}]} d(\mathbf{f}_{Q,n}, \mathbf{f}_M) \\ n^* &= \operatorname*{argmin}_{n \in [N_{\mathrm{A}}]} d(\mathbf{f}_{Q,n}, \mathbf{f}_M) \end{aligned} \tag{3}$$

where $[N_{\mathrm{A}}]$ indicates a set $\{1, 2, \ldots, N_{\mathrm{A}\text{-}1}, N_{\mathrm{A}}\}$ and $\mathbf{f}_{Q,n}$ is a SCD whose columns are shifted from the original one by an amount n. The column-shift process aligns the rotational variance for PC and lateral displacement for CC.

F. Subdescriptors

The above-mentioned naive comparison over the full 2-D descriptor is computationally expensive. To alleviate this cost, we introduce two subdescriptors. From the full 2-D descriptor, SCD, we extract 1-D vectors by summarizing the descriptor in the row and the column direction. Each subdescriptor plays a major role in place recognition and semimetric localization.

1) Retrieval Key: The first subdescriptor introduced is the *retrieval key*, $\mathbf{v} \in \mathbb{R}^{N_R}$, a vector whose dimension is equal to the number of SCD rows, N_{R}. Given any function that $f_R(\cdot)$ maps a column of a SCD to a single real number, we *squeeze* the column dimension of an SCD by applying the $f_R(\cdot)$ for each row in an SCD. Additionally, the following condition is required. For a given row r of the SCD, a retrieval key function is defined based on Requirement 2.

Requirement 2. A *retrieval key function* $f_R : r \to \mathbb{R}$ is permutation invariant.

With this requirement, we can create a subdescriptor that is unaffected by the column order; this means we can produce a consistent subdescriptor independent of the internal factors of the nuisances (e.g., rotation or lane changes). Practically, we used the $L1$ norm for our experiments, but any other function can be used that maps a vector to a single real number by obeying the above requirement. The $L0$ norm was used in our previous work [1].

2) Aligning Key: Similarly as with the retrieval key, we introduce the *aligning key* $\mathbf{w} \in \mathbb{R}^{N_A}$ as another subdescriptor of the SCD, which is a vector whose dimension is equal to the number of SCD columns N_{A}. Although no requirement is needed for the aligning key, we adopted the same $L1$ norm when summarizing a column.

V. Three-Stage Place Recognition

Our place recognition algorithm consists of three parts: 1) place retrieval using a *retrieval key*, 2) semimetric localization via prealignment using an *aligning key*, and 3) full SCD comparison for potential refinement and localization-quality assessment.

A. Place Retrieval Using a Retrieval Key

Existing widely adopted solutions leveraged past trajectory or motion uncertainties to reduce the search space [29], [31]. Differing from them, we pursue global localization without prior knowledge. We solely rely on the descriptor itself while minimizing computational costs from global search by introducing subdescriptors.

Using all extracted retrieval keys in the map, we construct a *k-d* tree for fast search and retrieve the closest place in terms of the retrieval key. Potentially, the top k candidate indexes then may be retrieved to be verified at the full SCD comparison phase. Interestingly, we empirically found that using only the best candidate ($k = 1$) yields meaningful performance, outperforming the case using multiple candidates. A discussion on the candidate set size (k) will be presented in Section VIII-B. As a result of the tree search, we topologically retrieve the corresponding map place for the query.

B. Semimetric Localization Using an Aligning Key

Given a retrieved candidate place, the typical SLAM framework would proceed to metric-level localization by finding the relative pose between the query and candidate place recognized by the place retrieval module. Well-known approaches would include ICP and its variants, which compare two scans to find the optimal pose, minimizing an alignment cost. Despite their popularity, these metric localization methods may suffer local minima and required a good initial guess.

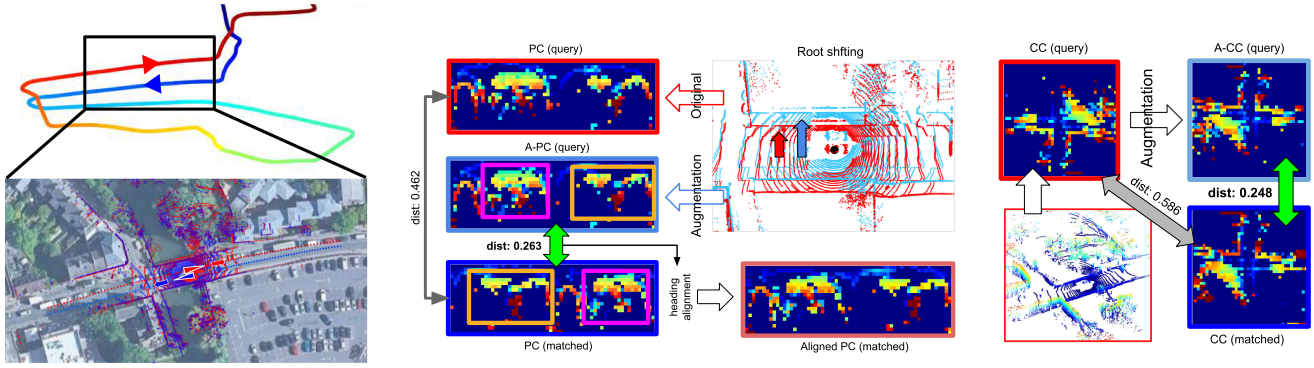
In the second phase of our place recognition algorithm, we exploit the aligning key and determine the partial relative pose through the prealigning phase. The naive brute-force version of the alignment (3) is computationally proportional to the number of columns N_{A}, which is heavier than the simple and frequently used $L2$ norm. We propose conducting brute-force aligning by using query and target aligning keys, instead of using the full SCDs. The prealignment using the aligning key procedure is formalized as

$$\hat{n}^* = \operatorname*{argmin}_{n \in [N_{\mathrm{inv}}]} d_{\mathbf{w}}(\mathbf{w}_{Q,n}, \mathbf{w}_M) \tag{4}$$

where $\hat{n}^*$ is the estimated shift for the best alignment between the query and target SCDs. We simply propose using $d_{\mathbf{w}}$ as the $L2$ distance between two vectors. This computed column shift $\hat{n}^*$ can serve as a good initial value for further localization refinement such as ICP. The evaluation of this initial guess is given in Section VII-E.

C. Full Descriptor Based False Positive Rejection

The final step of place recognition is to compare the full SCD to reject the potential false positive. As will be shown in



(a) (b) (c)

Fig. 5. Illustration of the SCD augmentation in V-D. (a) Sample from `Oxford 2019-01-15-13-06-37` shows the revisit case with both rotational and translational change. (b) Polar context augmentation includes explicit recalculation of the descriptor by changing the vehicle's center pose. The original (red) pose-based descriptor shows a larger distance than the shifted pose-based descriptor does. Only a single virtual vehicle case is shown for visualization. Please be noted that PC can recognize a place even under viewpoint changes [i.e., switched colored boxes in Fig. 5(b)]. (c) Cart context augmentation consists of simple sequential flips. Similarly, as in PC, the augmented descriptor shows a closer distance to the map than the original descriptor.

Section VIII-B, using a full descriptor may deteriorate the spatial discernibility. Using the previously computed initial column shift $\hat{n}^*$, the original search space in (3) is shrinked to only the neighborhood $\mathcal{N}(\cdot)$ of the *prealigned* shift $\hat{n}^*$

$$D(\mathbf{f}_Q, \mathbf{f}_M) = \min_{n \in \mathcal{N}(\hat{n}^*)} d(\mathbf{f}_{Q,n}, \mathbf{f}_M). \tag{5}$$

This reduced search space may be insecure when the variation over columns is poor, when the upper vertical FOV is low [12], and when our maximum height bin encoding function hardly makes a diversified distribution. This can be overcome by developing a more discerning bin encoding function. However, as will be shown in experiments (see Section VII), we found that even an extremely tight choice of neighbor, $\mathcal{N}(\hat{n}^*) = \{\hat{n}^*\}$ (i.e., assuming the prealigning as the best alignment) is empirically enough and outperforms other methods.

Finally, we go over the k candidates proposed by the *k-d* tree and search for candidates satisfying an acceptance threshold to select it as the revisited place

$$c^* = \underset{c_k \in \mathcal{C}}{\operatorname{argmin}}\, D(\mathbf{f}_Q, \mathbf{f}_M^{c_k}),\ \text{s.t}\ D < \tau \tag{6}$$

where $\mathcal{C}$ is the candidate index set extracted from the *k-d* tree, τ is the acceptance threshold, and c^* is the index of the recognized place. Because we use $k = 1$, this full descriptor similarity score performs as the validity check to confirm that $D < \tau$ before accepting the candidate as the correct match.

D. Augmentation of the Scan Context Descriptor

Because we construct a descriptor from the BEV, the dominant motion complexity is reduced to 3-degree of freedom (DOF) which is then summarized in a 2-D descriptor. This indicates that both descriptors are deficient in certain DOFs. For example, PC is written in the polar coordinates and loses the translational component; CC is described in the Cartesian coordinates and lacks the rotational component. This deficiency is critical when revisit occurs in a combined motion. A typical example would be revisiting in a reversed route from the opposite lane. To overcome this limitation and impose robustness along the fixed axis, we created virtual SCDs to augment a place, thereby achieving pseudo-invariance along the deficient direction.

1) Augmented PC: We aimed to cover lane changes (2-m spaced lanes) and a reversed route (180° heading change). During this augmentation process, a PC is synthetically duplicated by assuming virtual lateral displacement. Our particular interest is lane change, and we synthetically considered two virtual vehicle positions that are laterally 2-m apart. Two additional augmented polar contexts (A-PCs) are generated with respect to these virtual vehicle poses and root-shifted point clouds. This *root shifting* is in the same way as in our previous work [1].

2) Augmented CC: For CC, the augmentation is as simple as a double flip on both axes. The lacking rotational component should encompass lane changes, and we flip the descriptor on both axes to create the augmented cart context (A-CC).

Both the A-PC and the A-CC are illustrated in Fig. 5. The augmented descriptors' place index is assigned as identical to its original one. For matching, empirically, we found that maintaining a single *k-d* tree containing both original and augmented keys outperforms using multiple *k-d* trees.

E. Computational Complexity

Among all of the introduced modules, the neighbor search is the most computationally demanding. Tree construction consumes periodic resources and the add-on augmentation step requires increased time computation proportional to the number of the augmentations. As will be shown in Section VII-G, the number of augmentations and periodic tree maintenance are negligible. Even the main computational bottleneck of the retrieval module is extremely lightweight.

Naive descriptor comparison, as described in (2) and (3), requires the computation of $\mathcal{O}(N_A \cdot N_R \cdot N_A)$. This cost is substantially reduced by prealignment, as described in (4) and (5), eliminating linear search through N_A elements. The reduced computational cost becomes $\mathcal{O}(N_A \cdot N_R \cdot 1)$. Approximating $N_A \sim N_R \sim N$, this reduction can be regarded as a reduction

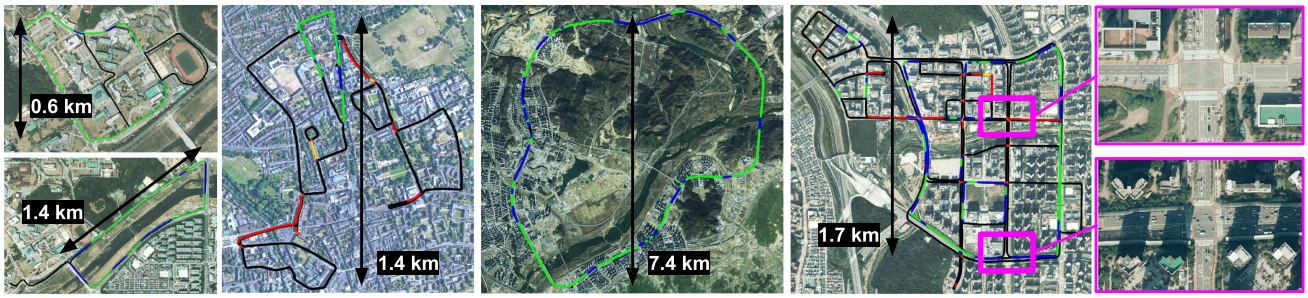


Fig. 6. Dataset trajectories overlaid on each aerial map. The first column shows `KAIST 03 (MulRan)` and `Riverside 02 (MulRan)`, followed by `Oxford Radar RobotCar`, the `Sejong` sequence in `MulRan`, and `Pangyo (NAVER LABS)`. The magenta boxes on the right show the wide roads in the `Pangyo` sequence of the `NAVER LABS` dataset. The total length of each sequence and their characteristics are summarized in Table III.

TABLE II
IMPLEMENTATION DETAILS

Parameter	PC / A-PC	CC / A-CC
Down sampling	$0.5 \times 0.5 \times 0.5\text{m}^3$	
ROI	$([0\text{ m}, 80\text{ m}], [0°, 360°])$	$([-100\text{ m}, 100\text{ m}], [-40\text{ m}, 40\text{ m}])$
Resolution	20×60 (4 m, 6°)	40×40 (5 m, 2 m)
Candidate # (k)	1	
Augmentations #	2	1
Augmentation	$\pm$2m root shiftings in the lateral direction	double flip

from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$ with the descriptor dimension N. For example, the CC in Fig. 3(b) is a square matrix with the format $N_A = N_R = N$.

F. Implementation Details

The used parameters are listed as in Table II. Here, ROI and grid size determines the resolution. For example, 20×60 for PC indicates $80/20 = 4$ m and $360/60 = 6°$ resolution. Similarly, 40×40 for CC indicates $200/40 = 5$ m and $80/40 = 2$ m resolution, for the R-axis and the A-axis, respectively. The discussion on parameter selection will be given in Section VIII.

VI. DATASET AND EVALUATION CRITERIA

For the evaluation, we chose trajectories to cover broad revisit types including rotation and lateral changes. We describe the datasets and evaluation criteria below.

A. Datasets

In total, eight sequences were selected from four publicly available datasets covering diverse environments: `KITTI Odometry` [12], `MulRan` [34], `Oxford Radar RobotCar` [33], and `NAVER LABS`[1] datasets. The detailed characteristics of each sequence and the environment will be provided in the following subsections. The overlaid trajectories on the aerial map, as shown in Fig. 6, illustrate the trajectory shape, scale, and surrounding environments (excluded well-known `KITTI` sequences). The details of the four datasets are summarized in Table III.

1) KITTI: KITTI Odometry[2] [12] is the most widely used dataset for LiDAR place recognition [25], [29]–[32]. This dataset provides 64-ray LiDAR scans (Velodyne HDL-64E). We selected two sequences, `00` and `08`, with a sufficient number of loops. Note that sequence `08` is only composed of reverse loops.

2) MulRan: The multimodal range dataset (MulRan) [34] was specifically designed to support place recognition evaluation and contains a large number of loop events. This dataset provides 64-ray LiDAR scans (Ouster OS1-64) in 12 sequences covering a campus for a planned city. We chose three sequences: `KAIST`, `Riverside`, and `Sejong`.

`KAIST 03` is a campus environment with few dynamic objects and multiple well-distributed buildings. `Riverside 02` involves travel on roads along the riverside. This sequence includes few surrounding structures and many perceptually similar unstructured objects such as roadside trees, which are frequently repeated throughout the sequence. More critically, this sequence has multiple lane changes at the revisit phase [the blue parts in Fig. 10(d)], which enable us to quantitatively assess the methods' robustness under lateral changes. The third environment in MulRan, the `Sejong` sequence, encompasses the long circular route of a master-planned city called Sejong [43]. As a planned city, its environment reveals slowly varying structural changes even within a relatively short period of time. We chose `Sejong 01` and `Sejong 02` and examined the multisession loop-closure capability and the robustness over a temporal gap (between June 2019 and August 2019).

3) Oxford Radar RobotCar: The `Oxford Radar Robot Car` [33] dataset, which we simply call `Oxford`, is a radar extension of the Oxford RobotCar dataset [44]. This extension provides range data from an FMCW radar and two 32-ray 3D LiDARs (Velodyne HDL-32E) mounted at the left and right sides of the radar. For each place, we constructed a single point cloud by concatenating scans from the left and right LiDARs (their center is a new sensor coordinate) and used this newly generated scan for the evaluation. The sites of `Oxford` mostly have a maximum of two lanes and no expected heavy lateral displacement. Instead, the sequence contained reverse revisits occurring simultaneously with small lane changes [i.e., the red

[1][Online]. Available: https://hdmap.naverlabs.com/ and https://challenge.naverlabs.com/

[2][Online]. Available: http://www.cvlibs.net/datasets/kitti/eval_odometry.php

TABLE III
DATASETS' DETAILS WITH RESPECT TO THE INVARIANCE TAXONOMY IN TABLE I

Induced variance				*T*	*R*	*D*	*SV*	*NR*	*FOV*
Dataset	**Sequence**	**Path length (km)** (# revisits / # total)	**Avg/Max Speed** (km/h)	**Non-same direction Revisits** (ratio)	**Lane Changes**	**Dyn Objs**	**Inter-session**	**Sensor** (# rays)	**HFOV** (°)
KITTI	00	3.71 (852 / 4541)	28.5 / 47.9	Y (3%, 22 / 852)	N	★	N	64	Full
	08	3.21 (102 / 2377)	27.5 / 46.7	Y (100%, 102 / 102)	★	★	N	64	Full
MulRan	KAIST 03	6.25 (2055 / 4224)	26.3 / 54.4	N (0%, 0 / 2055)	N	★	N	64	290
	Riverside 02	6.61 (2174 / 4870)	35.6 / 66.6	N (0%, 0 / 2174)	★★★	★★★	N	64	290
	Sejong 02 to Sejong 01	23.16 (17907 / 18090)	40.0 / 67.4	N (0%, 0 / 17907)	★★	★★	Y	64	290
Oxford Radar RobotCar	2019-01-11-13-24-51	9.93 (2117 / 8192)	24.4 / 42.2	Y (43%, 901 / 2117)	★	★★	N	32	Full
	2019-01-15-13-06-37 to 2019-01-11-13-24-51	8.89 (7391 / 7391)	25.1 / 50.8	N (0%, 0 / 7391)	★	★★	Y	32	Full
NAVER LABS	Pangyo	31.37 (7025 / 21648)	23.8 / 41.5	Y (29%, 2021/7025)	★★★	★★★	N	32	Full

places in Fig. 12(a)]. This dataset enabled us to evaluate the robustness to concurrent rotation and lateral changes.

Among the repeatedly recorded sequences over the same site, we selected two sequences (`2019-01-11-13-24-51` and `2019-01-15-13-06-37`) whose INS and GPS signals were secured over the entire trajectory. The sequence `2019-01-11-13-24-51` was used for an intrasession place recognition validation as shown in Fig. 12. The selected sequences were also used to validate the intersession place recognition performance, which is named `2019-01-15-13-06-37 to 2019-01-11-13-24-51` and is visualized in Fig. 15(a). We can see all global relocalizations (i.e., revisits) arose within the same direction.

4) Naver Labs: The last evaluation sequence is a long single trajectory through highly urbanized environments, named `Pangyo`, from the `NAVER LABS` dataset[3] made by NAVER LABS. The long 31-km sequence includes tall buildings, wide roads (the magenta boxes in Fig. 6), and multiple revisits per place. More than half of the same-direction revisits occurred in different lanes accompanied by rotation changes. We used `Pangyo` to validate a method's comprehensive performance and scalability.

B. Correctness Criteria

The measure of the each place strongly depends on the applications and the target environment (e.g., indoors or outdoors). In this evaluation, we aimed to include changes of up to three lanes (approximately 8 m), which frequently occur in complex urban sites. By doing so, the robot recognizes a place even when revisiting occurs at a laterally separated location. Second, in SLAM applications, coarse global loop detection typically followed by the pose regression module generates a metric constraint between the query and the map. If the loop candidate is detected too broadly (e.g., 25 m in [36]), then the accompanied fine localization module may fail. Considering these two aspects, we count the detected place as correct if a query place and a detected loop candidate place are less than 8-m apart. We prepared $1-1.5$ m equidistant sampled measurements to avoid redundant frames during stop sections and to enable each place to contribute the same. The numbers of nodes for each sequence used for the evaluation are reported in Table III.

[3][Online]. Available: https://hdmap.naverlabs.com/ and https://challenge.naverlabs.com/

C. Evaluation Metrics

1) Precision–Recall Curve: We used the precision–recall curve as a main evaluation metric [6]. As argued in [6], for a place recognition system, increasing potential matches is important, even if a few false predictions occur [45]. We also examined the maximum F1 score [46], the harmonic mean of precision and recall, as our evaluation metric.

2) Recall Distribution: We would like to note that the precision–recall curve may not fully reveal the performance toward loop-closure in a SLAM framework. The spatial and temporal distributions of the loop-closure are essential for the SLAM, while the precision–recall curve could be limited to measuring the distribution of place recognition. *Not all recalls should be credited equally* from the point of view of SLAM loop-closure. To value more distributed loop detections, we formulated the true revisits as the reference loop distribution and measuring Kullback-Leibler (KL) divergence against it.

As illustrated in Fig. 7(a), we constructed a histogram of the loop-closure event with respect to the translational and rotational variance between the nearest one in a map and a query pose. The sample revisit events collected from `Oxford 2019-01-15-13-06-37` contains two major groups. In this toy example, we simulated three algorithms showing different recall distributions and measured KL divergence with respect to the ground-truth recall distribution. In Fig. 7(b), few loop-closures are found from group 2 for the leftmost case. The other two showed better distributed loop-closure detections with respect to the internal factor variation, providing spatially unbiased localization performance. Even with smaller revisit detection, the middle case yielded better distribution showing lower KL-D value. During the evaluation, we show arrows to indicate that higher precision (↑), higher F1 score (↑), and lower KL-D (↓) imply better performance.

Potentially, the Wasserstein distance (a.k.a. the earth mover's distance) or Jensen–Shannon divergence could be the measure to use as also discussed in detail in [47]. However, we chose to use KL-D because we need to compare the relative distance between

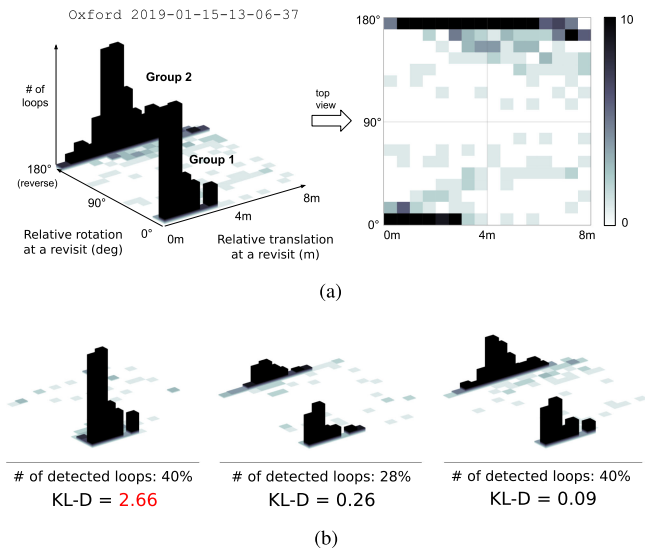


Fig. 7. (a) True distribution of the loop-closures from the `Oxford 2019-01-15-13-06-37` sequence seen in a perspective view (left) and a top-down view (right). The grid size used in the visualization is (0.5 m, 10°). The loop-closure events are majorly grouped into two. (b) Three sample algorithms showing different detected event distributions.

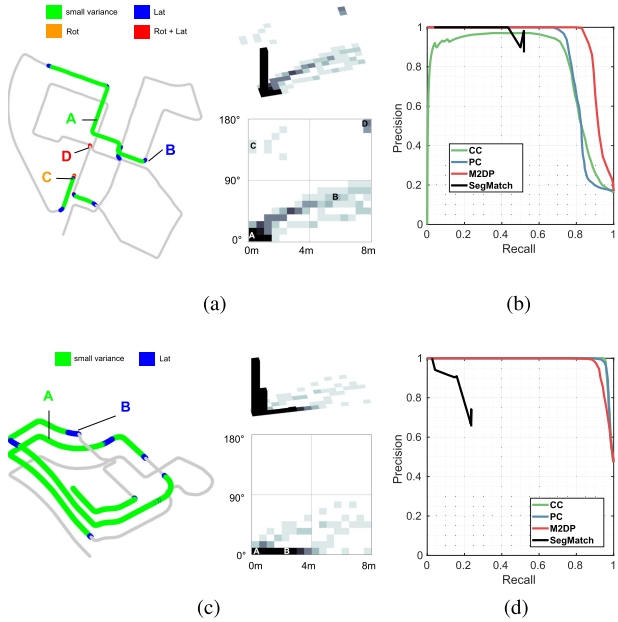


Fig. 8. (Left) Marked trajectories of the revisited places, (middle) the true revisit distributions, and (right) the PR curves. In (a) and (c) for `KITTI 00` and `KAIST 03`, the trajectories are color-coded by revisit types. Most of the loop-closure events are concentrated in the lower left of the true-revisit distribution, revealing small rotational and translational variance. This concentrated distribution is depicted as a single peak in the perspective view.

methods while having the GT distribution as the reference. Measuring relative information is favored over the symmetry. Here, we used the ground-truth loop-closure as the reference distribution and measure relative entropy against this reference.

D. Comparison Targets

We compared the proposed methods against two other methods: *M2DP* and *SegMatch*. All of the comparison targets are agnostic to sensor type (e.g., ray numbers) and run on CPU.

1) SCD: We present the performance of PC [1], CC, A-PC, and A-CC. For the proposed methods, we only retrieved a single candidate from the *k-d* tree ($k = 1$). Downsample point cloud using a $0.5\ \mathrm{m}^3$ voxel is used to make an SCD (Table II). The evaluation curves were acquired by changing the threshold of the SCD distance.

2) M2DP: Identical to our methods, M2DP [25] only requires a point cloud from a single scan as an input. We followed the code and the parameters provided,[4] with one difference. We empirically found that applying 0.1-m cubic voxel downsampling *a priori* boosts M2DP's performance, and we made this modification to secure better performance. The query descriptor is compared to all of the map descriptors in terms of Euclidean distance, which was used as a threshold.

3) SegMatch: Among the three options in SegMatch [29], we used the eigenvalue to describe a segment, which is the same as the author's configuration designed for the `KITTI` dataset. We excluded the learning-based version, SegMap [28], because our method works on CPU and for a fair comparison. The evaluation curves for SegMatch are acquired by changing the segment feature distance threshold. Unlike the other global localization methods (ours and M2DP), SegMatch requires odometry information. During the evaluation, we leverage ground-truth to provide odometry. As will be seen, despite the exploitation of highly accurate odometry, SegMatch failed to overcome severe variance, while our method reliably localized without requiring any geometric prior. Not being a global descriptor as M2DP and SCD are that SegMatch only had a short range in its PR and DR curve. This is because the parameters in SegMatch are tuned to local segmentation and do not substantially affect recall.

[4][Online]. Available: https://github.com/LiHeUA/M2DP

VII. Experimental Evaluation

Next, we validated our spatial descriptor and place recognition algorithm on various datasets. As addressed in Fig. 1 and Table I, coping with multiple variations of a place is crucial for loop detection and global localization. To clearly state the associated invariance, we color-coded routes depending on the revisit types.

A. Revisit With Small Variance

Among the eight sequences in Table III, `KITTI 00` and `MulRan KAIST 03` are relatively *easy* sequences, containing small rotational/translation variance and few dynamic objects. For `KITTI 00` [Fig. 8(b)], M2DP showed the highest performance with respect to both precision and recall. SegMatch revealed quite lower recall compared to the others; however, the distribution of the recognition was sufficient to construct a globally consistent map. In particular, SegMatch successfully recognized the loop at the middle crossroad, where a composite change (both *rotational and lateral*) existed [see Fig. 9(a)], while

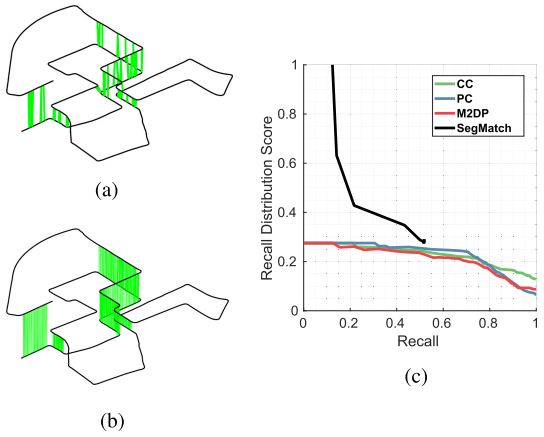


Fig. 9. (a) and (b) Matched pairs (green) of SegMatch and ours (PC) for KITTI 00 at maximum precision of Fig. 8(b). (c) Distribution–recall curve, named DR curve, in terms of KL divergence with respect to the recall rate. Ideally, a flat curve with constant 0 KL divergence for all recall rates would indicate perfectly distributed recalls. A lower score in the DR curve indicates better performance (↓). In this DR curve, CC, PC, and M2DP all showed low KD divergence scores for all recall rates, which gradually decayed as recall increased. SegMatch detected sparse loop-closures, and the recall was lower than that of the other methods. This appears as a larger KL divergence at low recall. However, the KD divergence decreased dramatically and reached a similar level of other methods. This indicates that SegMatch proposed more efficient loop-closures, achieving similar distribution score (i.e., KL divergence) with a smaller number of detections.

the other methods failed to do so. Both PC and CC showed similar performance because `KITTI 00` barely has any rotations or lane changes at the loops. The PC matched pairs at the 100% precision are visualized in Fig. 9(b). In `MulRan KAIST 03` [Fig. 8(d)], all of the methods successfully recognized the loops because this sequence is for a campus environment with almost no lane changes and few dynamic objects.

B. Revisit With Rotational or Lateral Variance

Next, we examined sequences showing dominant variance in either the rotational or lateral direction. Their performance is summarized in Fig. 10.

1) `KITTI 08`*:* This sequence only contains reverse revisits, with half of them further including simultaneous lane change. This appears as the concentrated distribution of revisit events in Fig. 10(a). M2DP and CC failed due to the severe rotational variance while PC showed substantially better precision. SegMatch yielded enough precision, but the recall is limited. For this sequence with rotational variance, we examined the A-CC to see the improvement of the augmentation.

2) `MulRan Riverside 02`*:* In this sequence, the vehicle revisits a place with multiple lane changes but in the same direction. This variance is clearly captured in Fig. 10(d). In terms of precision–recall, CC outperformed the other techniques by large margin [Fig. 10(e)]. The time-elevation graph in Fig. 11 shows true/false matches for the sequence. CC outperformed the others in challenging regions with few false positives (red), which can potentially be treated using existing robust back ends [45], [48], [49]. As with the improvement of A-CC in `KITTI 08`, the augmentation (A-PC) improved the PC under lateral variance.

C. Concurrent Rotational and Lateral Variance

The more complex case includes concurrent rotational and lateral variance. We used `Oxford` and `Pangyo` to evaluate performance under composite variance. We excluded SegMatch for the composite cases because of their high dependency on odometry. Enhancing other prior modules for place recognition is beyond the scope of this article.

1) `Oxford`*:* As shown in Fig. 12, the performances of the original PC and CC without augmentation are steeply limited at a certain recall, even with increased thresholds. Interestingly, the unrecognized recalls at this steep point matched the ratio of nonsame direction revisits (43%) shown in Table III. Applying associated augmentation to PC and CC showed improved precisions at the higher recalls, with large margins for both descriptors. Overall, the A-CC generally showed higher precision than the A-PC did. In Fig. 13, true/false matches for each method are visualized. For a fair comparison, we pinned the recall at 50% for all methods to measure each method's accuracy and effectiveness quantitatively.

Note the importance of the distribution shown in Fig. 12. According to this plot, CC outperforms PC in terms of precision and maximum F1 score, except for the distribution score. This indicates that the increased precision of CC is concentrated in easy regions, while PC can detect difficult loops that may critically contribute to SLAM performance [see Fig. 13(b)]. However, the restricted performance of CC was alleviated by A-CC, as can be seen in the improved distribution score as shown in Fig. 12(d). This improvement is also depicted in Fig. 13(e), in which A-CC detects well-distributed loop-closures.

2) `NAVER LABS Pangyo`*:* This `Pangyo` sequence includes sporadic lane changes during revisits, accompanied by rotational change. This composite variance (both *rotational and lateral*) is inevitable in an urban environment when the reverse route necessarily involves a lane change. The `Pangyo` sequence encompasses abundant types of variance as can be seen in Fig. 14. Overall, augmentation yielded substantial improvement when the revisit underwent composite variance. A-PC showed the best performance for rotational change [Fig. 14(g)] and M2DP was meaningful for lateral variance. However, under concurrent rotational and lateral variance, A-CC proved its validity over other methods.

D. Multisession Capability

So far, we have investigated revisits within a single session. Here, we consider place recognition in multisession scenarios toward long-term autonomy. To validate our methods in multisession scenarios, we chose two sequences from a dataset with sufficient temporal differences.

The first pair was `Oxford 2019-01-15-13-06-37` to `Oxford 2019-01-11-13-24-51`. We used `Oxford 2019-01-11-13-24-51` as a map and tested the loop-closure performance of `Oxford 2019-01-15-13-06-37` as a query sequence. Another pair used for testing multisession loop-closure showed a larger temporal gap of two months. We chose `Sejong 01` in the `MulRan` dataset as a map using `Sejong 02` as a query.

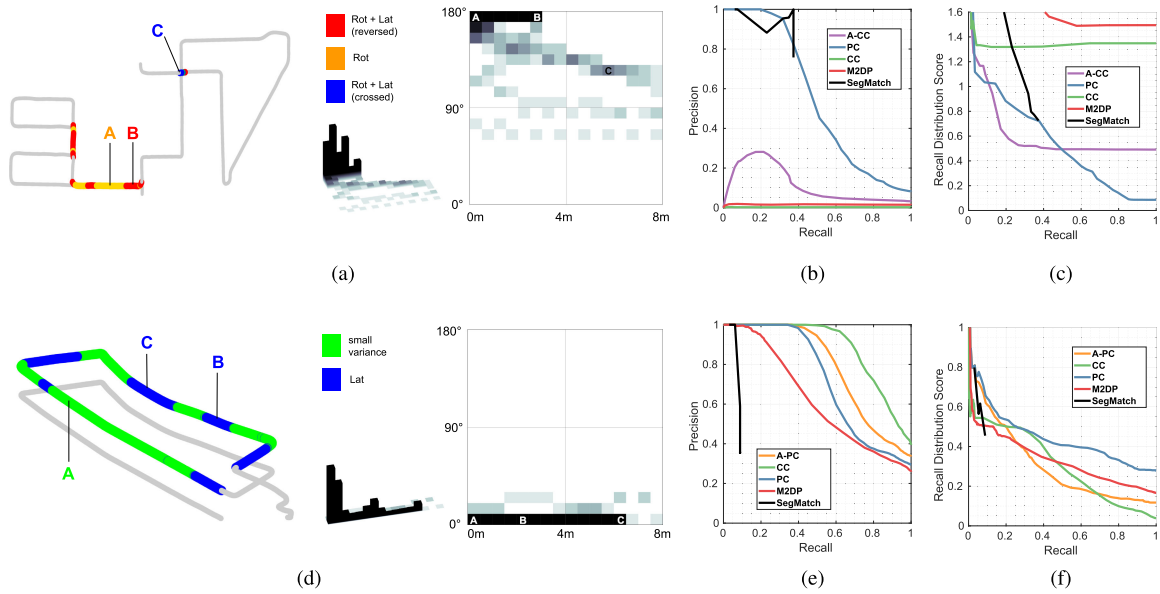


Fig. 10. (a) and (d) Trajectories and revisit distributions of two sequences showing either predominantly rotational or lateral variance from `KITTI 08` and `MulRan Riverside 02`. (a) Most of the revisits occurred in the reversed direction for `KITTI 08`, with a concentrated distribution on the upper-left quadrant. (b) and (c) PR curve (↑) and DR curve (↓). PC is the most robust method for KITTI 08. (d) `MulRan Riverside 02` contains lateral variations with little rotational change. (e) and (f) CC is better capable of handling lateral variations. Higher precision (↑) and lower recall distribution score (↓) for all recalls indicate better performance.

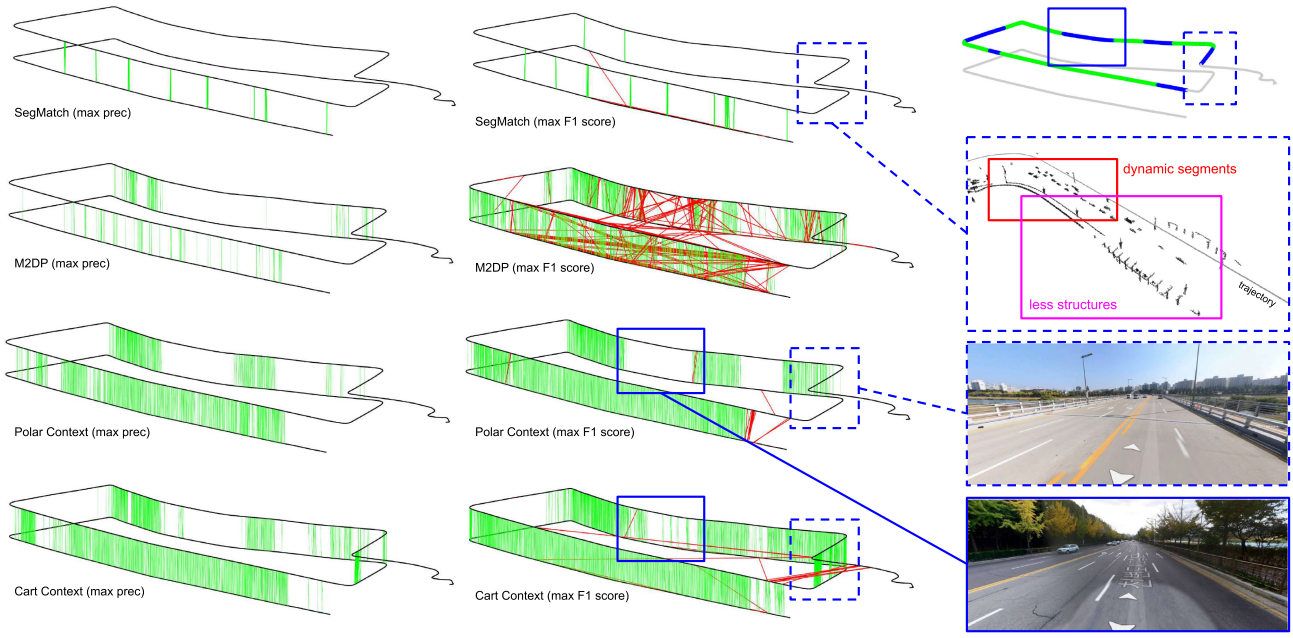


Fig. 11. Time-elevation plots with true/false matches visualization for `Riverside 02`. We plotted correct (green) and incorrect (red) matches at the maximum precision (100% for all methods) and the maximum F1 score. The solid blue box indicates the area with challenging multiple lane changes with repeated trees. The dotted blue box is the featureless bridge environment crossing a river. CC successfully found loops at those regions, while M2DP proposed many incorrect matches, and PC did not find loop-closures in these regions.

As can be seen in Fig. 15, these revisits mostly included lateral variance but with a temporal gap. For the `Oxford` pair, all of the methods successfully detected loops. The `Sejong` pair was more challenging because the lateral change included multiple lane changes. The loop-closure results for the `Sejong` pair are further visualized in Fig. 16. M2DP seemed to show meaningful performance but included wrong loop-closures. CC showed the best performance for the multisession scenarios. Obvious improvements could be made via augmentation, although this was excluded from the multisession scenarios.

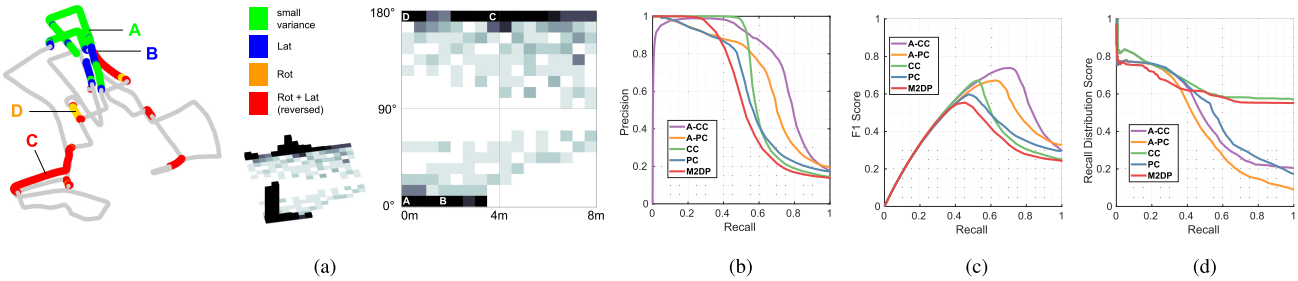


Fig. 12. (a) `Oxford 2019-01-11-13-24-51` revealed concurrent rotational and lateral variance. The trajectory is color-coded by variance types. The true revisit distribution shows reversed revisits with lane changes. In addition to the PR and DR curves, we show the F1-R curve, which shows the change of F1 score with respect to the recall. (b)–(d) PR curve, F1-R curve, and DR curve. From the DR curve, we can easily see that augmentation not only increases the number of recalls with higher precision but also increases the diversity of loop conditions.

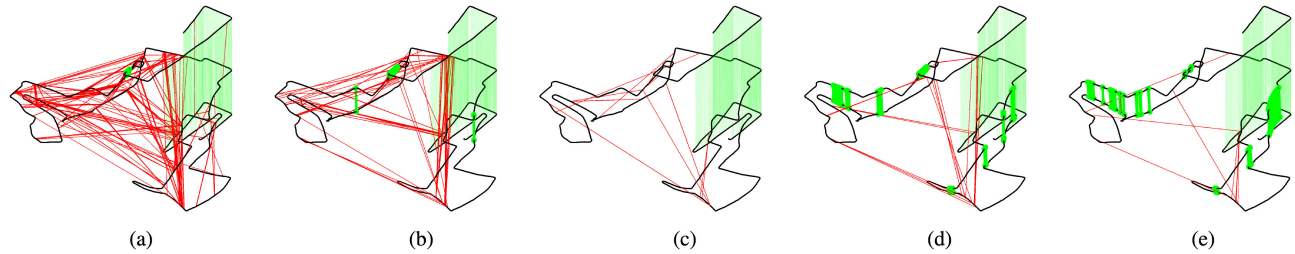


Fig. 13. (a) M2DP (b) PC (c) CC (d) A-PC (e) A-CC. Time-elevation match graph for `Oxford 2019-01-11-13-24-51`. Both true and false loop detections at recall of 50% are visualized. The black line is the sequence trajectory whose height represents the time. The falsely connected matches are red, the true matches at *easy* revisit are green, and the true matches at revisit with variance are drawn as green (bold).

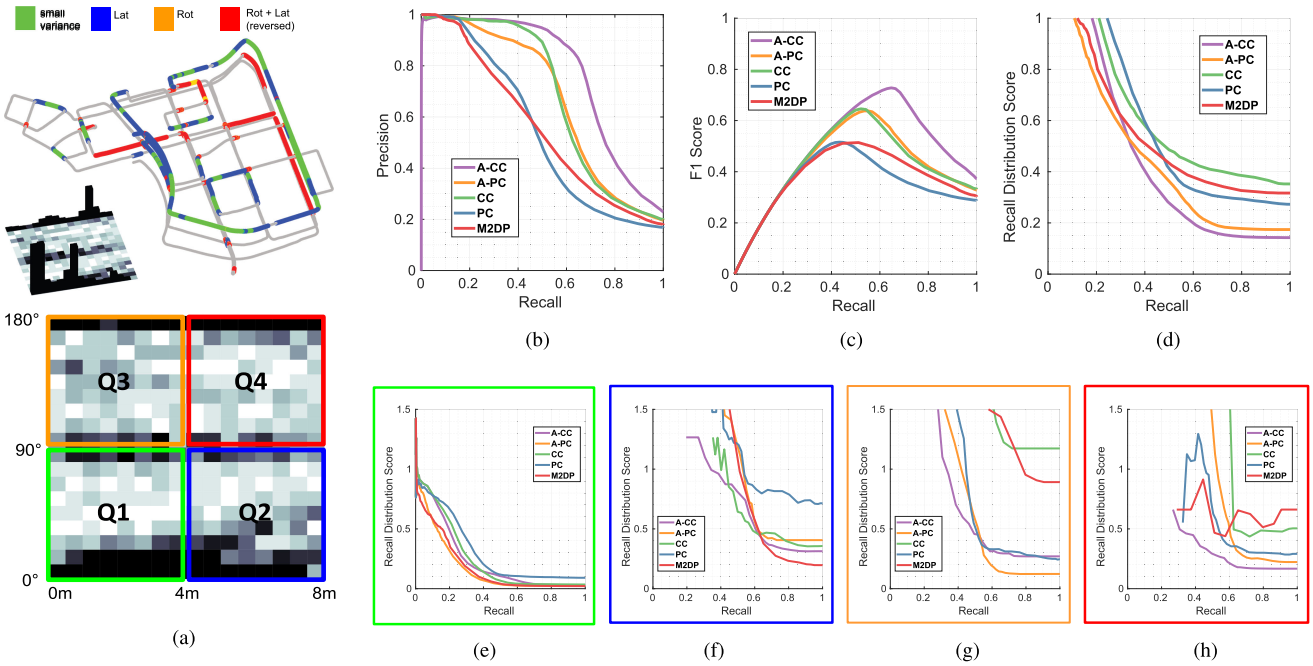


Fig. 14. (a) Trajectory color-coded by variance types. Unlike other sequences, `Pangyo` includes all types of variance. Each quadrant indicates easy (Q1), lateral-dominant (Q2), rotational-dominant (Q3), and composite (Q4) cases. For the sequence, (b) PR curve, (c) F1-R curve, and (d) DR curve are given. On the bottom row, we examine a detailed view for each quadrant from the top-down view of the revisit distribution. Each quadrant represents the predominant group of revisits in one type of variance.

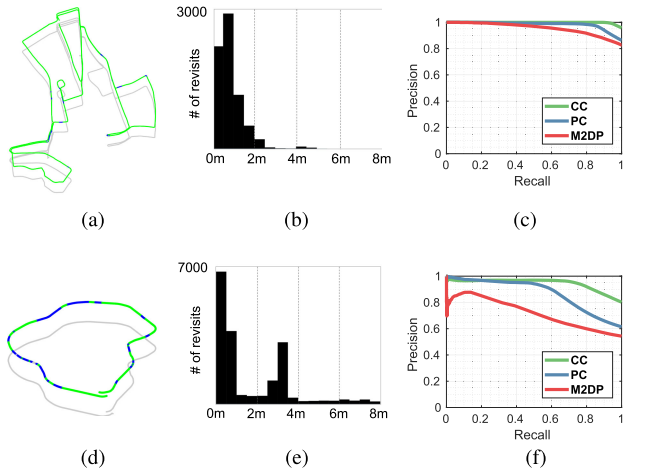


Fig. 15. Loop-closure detection under multisession scenarios. (Top) A pair from the `Oxford` dataset and (bottom) a pair from the `Sejong` of `MulRan` dataset were used. The revisits mostly had lateral changes. Single peak in Fig. 15(b) and double peaks in Fig. 15(e) describe the number of laterally displaced revisits in each sequence.

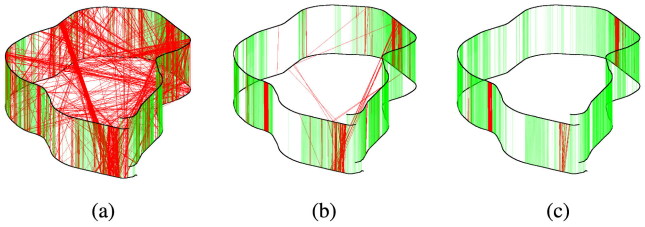


Fig. 16. (a) M2DP (b) PC (c) CC. Intersession place recognition visualization registering the query sequence (`Sejong 02`) to the map sequence (`Sejong 01`). A set of true (green)/false (red) loop detection results at recall of 50% are visualized. The black line is the sequence trajectory whose height represents the time. The performance during severe lane change [blue in Fig. 15(d)] is of interest. Only CC had green matches even under major lateral variance while suppressing perceptual aliasing (red lines).

E. Metric Localization Evaluation and Quality Assessment

Together with the retrieved place, the proposed method is capable of estimating the relative 1-D pose between query and map places. This is important when the topological place retrieval is combined with the metric localization because this initial estimate can be exploited in further metric refinement.

From the aligning key registration, we estimate a 1-D relative pose (i.e., rotation for PC and lateral displacement for CC). Using the ground truth pose provided in `Pangyo` sequence, we plot the estimated 1-D relative pose against the true relative pose from the ground truth. As can be seen in Fig. 17(a) and (c), the estimation yielded a meaningful relative pose inference of 1.03° for A-PC and 0.84 m for A-CC on average.

We can further examine the quality of this metric localization using the full descriptor similarity score. In the proposed method, we utilized this similarity score as the second barometer to exclude the retrieval with large SCD distance (i.e., small similarity). To assess the metric evaluation quality, we present a scatter plot between the RMSE of the relative estimation from ICP and the SCD distance in Fig. 17.

TABLE IV
ATEs (MEAN/MAX) OF ODOMETRY AND SCAN CONTEXT INTEGRATED SLAM SYSTEM

	KAIST 03		Riverside 02	
Methods	Trans. (m)	Rot. (deg)	Trans. (m)	Rot. (deg)
LeGO-LOAM	20.7 / 42.7	4.9 / 9.9	47.7 / 130.5	6.9 / 12.8
SC-LeGO-LOAM	3.4 / 8.8	2.2 / 8.2	15.2 / 50.5	4.2 / 8.7

F. External Module Dependence and SLAM Integration

Being lightweight and independent to an external module would be needed in a global localizer. We aimed to develop a stand-alone module without requiring prior information such as odometry. During evaluation, we found that SegMatch's place recognition performance is affected by the odometry quality. We also empirically discovered that the SegMatch hardly made recalls for harsh environments such as the `Riverside 02 (MulRan)` sequence when a good quality of frame-to-frame odometry is barely obtainable due to many dynamic objects. In the previous evaluations, although we fed the ground truth as the odometry to ensure their best performance, the performance was restricted for less structured and repeated environments.

The proposed implementation is lightweight provided in a single C++ and header file pair. Thus, ours is easy to combine with any keyframe-based pose-graph SLAM system because the scan context based place recognition's atomic element is a single keyframe measurement. Along our open source place recognition module,[5] we also made a real-time LiDAR SLAM system publicly available. It is written in C++ and named SC-LeGO-LOAM[6] integrated with LeGO-LOAM [50]. As in Table IV, the scan context based loop detection and pose-graph optimization with iSAM2 [51] successfully alleviated the odometry trajectory's drifts. For a detailed demonstration, we refer to the attached multimedia file.

G. Computational Cost

The proposed place descriptor generation and recognition modules are both fast. The per module computational costs are visualized in Fig. 18 for two sequences. PC's computation costs are reported in Fig. 18 because the computational costs for PC and CC are almost the same under a similar resolution and only the coordinate selections differed. These time consumptions are measured while running the scan context integrated real-time LiDAR SLAM (Section VII-F) on Intel i9-9900 CPU (3.10 GHz) and 64-GB RAM.

As can be seen in Fig. 18(b), the mean computational time is less than 10 ms. The most time-consuming task is the *k-d* tree reconstruction, which is performed periodically in batches. However, a graph plots the conservative case when we repeatedly rebuild the tree every other 10 s. This could be elongated depending on the application to reduce the total cost and is not even required for the multisession scenario.

The mean execution time is even shorter at `Pangyo` despite its large scale because `Pangyo` used 32-ray LiDAR with fewer

[5][Online]. Available: https://github.com/gisbi-kim/scancontext
[6][Online]. Available: https://github.com/gisbi-kim/SC-LeGO-LOAM

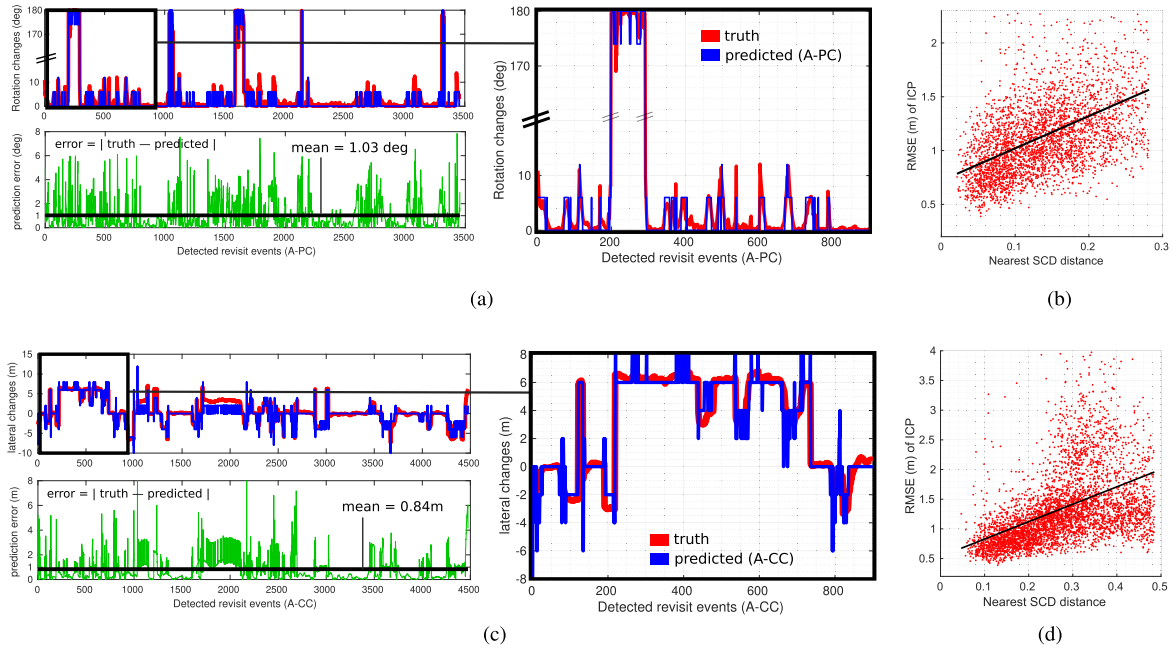


Fig. 17. Semimetric localization at max F1 score on `Pangyo` sequence. (a) and (c) Relative 1-D pose estimation via aligning key matching of A-PC (for relative rotation) and A-CC (for relative translation). Relative rotation (A-PC) and lateral translation (A-CC) are depicted in blue, while the red line indicates the true relative displacement obtained from the ground truth. Error is illustrated in green below the estimation plot. (b) and (d) Plotting RMSE between relative pose after ICP and ground truth against the scan context descriptor distance reveals correlation.

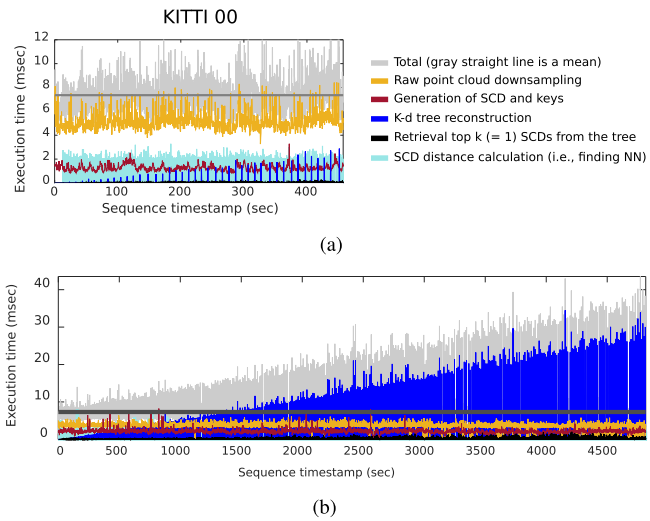


Fig. 18. Computational time visualization per module when tested over `KITTI 00` and `Pangyo` (31 km length). The average time cost (7.36 ms and 7.31 ms, respectively) is depicted as a straight dark gray line. Conservative batch tree update every other 10 s was performed.

points than `KITTI 00`. This also indicates that the overall computational complexity is $\mathcal{O}(1)$ though periodic batch tree rebuilding scales linearly with the map $\mathcal{O}(N)$, with N being the number of nodes in the map.

The time cost comparisons with the other methods are given in Table V. The timing for ours and M2DP were measured using Matlab, while SegMatch was copied from [29]. SegMatch spent most of its time on segmentation. M2DP was the most lightweight. A-PC only requires extra description times during the augmentation phase as no extra costs managing retrieval keys are needed. Despite requiring GPU (GTX 1080 Ti), PointNetVLAD (Section VIII-E) was more expensive than ours. The retrievals were fast for M2DP and PointNetVLAD because a fixed-length vector's comparison of Euclidean distance is very lightweight.

TABLE V
TIME COST COMPARISON

	Ours (PC)	Ours (A-PC)	M2DP	SegMatch	PointNetVLAD
Desription	1.6	4.8	4.3	430.2	33.3
Retrieval	6.7	6.7	1.5	365.8	0.7
Total	8.3	11.5	5.8	796.0	34.0

All units are in ms.

VIII. DISCUSSION

Beyond the evaluation of the proposed global localization method, we provide ablation studies and interpretations.

A. Descriptor Resolution

We examined the descriptor resolution and corresponding performances. As in Table VI, the lower resolution yielded better performance. Therefore, we used the baseline resolution for the following subsections.

B. Analysis on Retrieval Key Performance

Candidate numbers. In Section V-A, we leveraged the *k-d* tree to propose k candidates for retrieval and only a single answer is selected after the full descriptor based false positive rejection

TABLE VI
PERFORMANCE COMPARISON WITH RESPECT TO THE DESCRIPTOR RESOLUTION AT `Oxford 2019-01-11-13-24-51`

Polar Context (PC)				Cart Context (CC)			
Resolution	P (↑)	R (↑)	D (↓)	Resolution	P (↑)	R (↑)	D (↓)
10×30 (8 m, 12°)	0.65	0.41	0.56	40×20 (5 m, 4 m)	0.94	0.50	0.62
20×60* (4 m, 6°)	0.81	0.47	0.58	40×40* (5 m, 2 m)	0.93	0.53	0.60
20×120 (4 m, 3°)	0.76	0.50	0.57	40×60 (5 m, 1.3 m)	0.90	0.52	0.61
40×60 (2 m, 6°)	0.77	0.50	0.58	60×40 (3.3 m, 2 m)	0.92	0.52	0.60
40×120 (2 m, 3°)	0.76	0.50	0.59	80×80 (2.5 m, 1 m)	0.92	0.51	0.61
60×180 (1.3 m, 2°)	0.74	0.49	0.61	120×120 (1.6 m, 0.6 m)	0.91	0.52	0.60

P: precision, R: recall, D: KL-D at F1 max. The baseline resolution is marked with *.

TABLE VII
PERFORMANCE COMPARISON WITH RESPECT TO THE NUMBER OF FIRST STAGE'S CANDIDATES AT `Oxford 2019-01-11-13-24-51`

	Polar Context (PC)			Cart Context (CC)		
K	P (↑)	R (↑)	D (↓)	P (↑)	R (↑)	D (↓)
1*	**0.81**	0.48	**0.58**	**0.93**	**0.53**	**0.60**
10	0.76	**0.49**	0.60	0.87	0.51	0.61
50	0.72	0.49	0.59	0.76	0.51	0.61
100	0.71	0.49	0.59	0.70	0.49	0.62

P: precision, R: recall, D: KL-D at F1 max. The baseline resolution is marked with *.
The bold value means the best performance among the values of K.

TABLE VIII
FULL DESCRIPTOR SIMILARITY (BRUTE-FORCE SEARCH) AND RETRIEVAL KEY COMPARISON

	Polar Context (PC)		Cart Context (CC)	
Sequences	Retrieval key	Full descriptor	Retrieval key	Full descriptor
`KITTI 00`	0.84	0.85	0.80	0.34
`KAIST 03`	0.99	0.99	0.99	0.99
`Riverside 02`	0.72	0.74	0.88	0.84
`KITTI 08`	0.55	0.46	0.00	0.00

The measure is an area under a precision-recall curve (AUC).

(Section V-C). In this subsection, we examine the effect of k on performance. We first note that the increase in k does not mean to relax the success criteria, but rather the number of candidates in the first step of our algorithm. Interestingly, as in Table VII, all statistics outperformed others when we only chose the best candidate. The full descriptor may suffer confusion showing the best performance at $k = 1$. Though this may seem contrary to a general belief for better performance under more candidates, the result indicates the reduced spatial discernibility of the full descriptor. Based on this investigation, we used $k = 1$ for all experiments conducted earlier.

Retrieval key vs. full descriptor brute-force search. Additionally, we analyzed how the performance varies if an entire database is compared (i.e., brute-force) using the full descriptor-based distance (5). Through the multiple tests in Table VIII, the performance difference between the retrieval key based and the brute-force search is negligible although the brute-force search requires heavier computations following $\mathcal{O}(n)$ (e.g., almost 1 s for 4500 frames of `KITTI 00`).

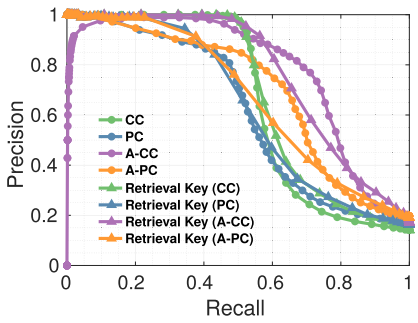


Fig. 19. Effect of the retrieval key in terms of the PR curve (`Oxford 2019-01-11-13-24-51`). The same SCD is depicted with the same color. A-CC and A-PC showed improvements when additional full descriptor similarity was considered.

TABLE IX
PERFORMANCES (AUC) WITH RESPECT TO CORRECTNESS THRESHOLD

	Polar Context (PC)				Cart Context (CC)			
Sequences	8 m*	4 m	2 m	1 m	8 m*	4 m	2 m	1 m
`KITTI 00`	0.84	0.88	0.91	**0.94**	0.81	0.85	0.87	**0.88**
`KAIST 03`	0.99	0.99	**0.99**	0.96	0.99	0.99	**0.99**	0.96
`Riverside 02`	0.72	0.73	**0.79**	0.73	0.88	**0.89**	0.87	0.81
`KITTI 08`	**0.55**	0.46	0.38	0.23	0.00	0.01	0.01	0.00

The baseline(*) threshold is 8 m.
The bold value means the best performance among the thresholds.

Full descriptor's effect. Nevertheless the confusion of the full descriptor based similarity proven in Table VII, this additional similarity validation enhanced the precision for the augmentation cases as well as its semimetric localization capability. In Fig. 19, the augmented scan context's precision was improved by eliminating less accurate matches via the supplemental similarity verification using a full descriptor.

C. Correctness Criteria

The performance tends to be improved when more tight criteria are applied as in Table IX. This phenomenon occurred because laterally displaced queries, which are generally difficult to recognize, are considered as correct rejections when they are missed. However, precise localization was more difficult in reversed revisits (i.e., `KITTI 08` in Table IX) because the previously correctly recognized queries (e.g., within 4–8 m) are considered false alarms. From these findings, 8 m was used for the criteria of correctly recognized places during our main evaluations in Section VII and the ablations in Section VIII to successfully cope with laterally translated revisits. Even if a place is recognized from a slightly distant place (e.g., 4–8 m apart), the proposed method can close a loop successfully because it provides a semimetric localization result.

D. Robustness to Roll–Pitch and Height Perturbations

The previously used datasets are mainly from wheeled platforms with little roll–pitch and height perturbations. However, sensor measurement variation can occur in terms of rotational and height variations between two scans. In this regard, we added the additional experiments on roll–pitch perturbed simulations

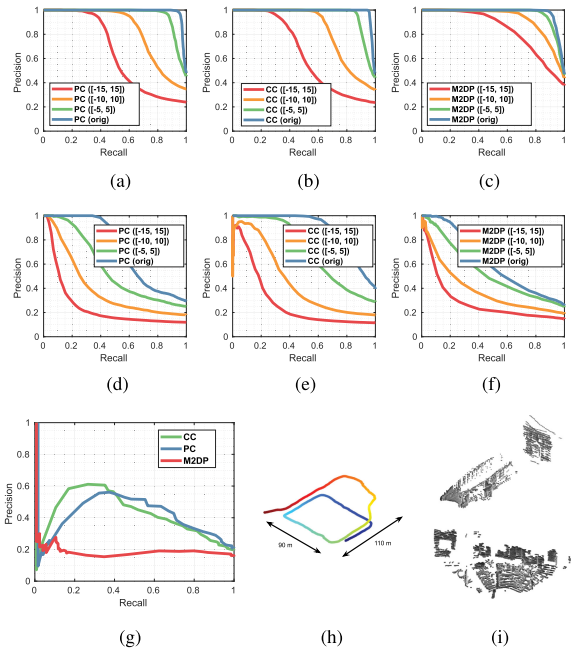


Fig. 20. (a)–(f) Perturbation simulation results for KAIST 03 ((a)–(c)) and Riverside 02 ((d)–(f)). (g), (h), (i) Real-world hand-held LiDAR dataset result, its time-elevated trajectory, and two example scans.

and a real-world hand-held LiDAR experiment. We randomly prerotated an input scan with respect to both roll and pitch for the simulation. In the real-world hand-held LiDAR dataset, the height of the measurement origins varies slightly while a human navigator walks.

Simulations. The degree of prerotations is divided into three levels: $[-5°, 5°]$, $[-10°, 10°]$, and $[-15°, 15°]$. The simulations are conducted for two sequences KAIST 03 and Riverside 02, and performance losses are clearly observed for all methods. For the KAIST 03 sequence, M2DP showed smaller performance drops than ours. However, in Riverside 02, the performance degradations became clear with respect to the degree of perturbation for all three methods. We believe that this topic, robust place recognition under severe roll–pitch variations, has still not been studied much, and it could be a valuable future academic research topic.

Hand-held data. Second, the result of real-world hand-held LiDAR data is given in Fig. 20(g). We used KA Urban Campus 1 sequence provided in LiLi-OM [52], which was acquired from a slowly walking human navigator. It has the same direction revisits and narrow front horizontal FOVs ($\sim 70°$). In this real-world data, ours outperformed M2DP by a large margin and showed that mild (e.g., human walking) roll–pitch and height perturbations are acceptable. Hence, the proposed method may not be restricted in wheeled platforms and work for a hand-held traverse under mild roll–pitch motions.

E. Comparison to Deep Learning Based Methods

We also provide comparisons to recent deep learning based approaches, SegMap [28] and PointNetVLAD [26][7]. For both methods, we used pretrained weights that the authors released. SegMap revealed hampered performance compared to SegMatch. This could be due to the limitation in generalization capability over unseen environments. PointNetVLAD showed comparable performance in the environment under little rotational and translational variations [i.e., Fig. 21(a)] but failed when the variation increased [Fig. 21(b) and (c), respectively].

[7]For the input processing, we follow [2]. An input is a ground-removed, zero-centered 4096 points within a [-25 m, 25 m] cubic region.

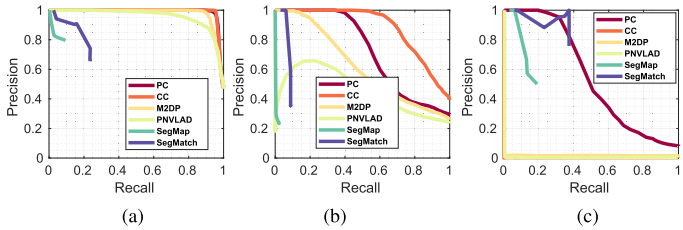


Fig. 21. Comparisons with deep learning based methods, SegMap, and PointNetVLAD. (a) KAIST 03 (b) Riverside 02 (c) KITTI 08.

F. Failure Cases

We illustrate sample cases when the proposed method succeeded and failed to localize against the map. As shown in Fig. 22(a), the proposed method has overcome lateral and/or rotational discrepancy between map and query scans. The SCD is successfully localized to the map even with many dynamic objects (e.g., cars). However, when a tall and large object (e.g., bus) appears very proximal to the sensor on both query and map scans, the localization might fail as in Fig. 22(c). The other failure case was found when the vehicle was moving along a corridor-like place [Fig. 22(b)].

G. Which SCD to Use?

The final question to answer is which SCD to use and in what case. Based on the evaluation, generally, A-CC yielded the best performance even under the composite variance (i.e., *Rot* + *Lat*) as in the case of Oxford (Fig. 12) and Pangyo (Fig. 14). Therefore, CC and A-CC are more preferred when the target environment is an urban road. We recommend using PC or A-PC for more general environments and when semimetric localization capability is more critical. Because classic ICP is much more sensitive to the rotational component of the initialization, PC would be a better choice despite a little sacrifice in precisions from CC (but still comparable performance). For patrolling robots and shuttles that repeat the same route with minimum variance, PC would exhibit more meaningful performance as proved in multisession scenarios (Fig. 15).

H. Limitations and Potential Extension

1) Invariance in One Direction: The proposed method is natively invariant in one direction and we chose rotation and lateral direction to be invariance axes. This limitation was overcome by a robust search scheme and augmentation.

2) Leveraging Deep Learning for Scan Context Descriptor: The proposed descriptor itself is in ordered 2-D format, and inputting this into a deep network is very straightforward. As

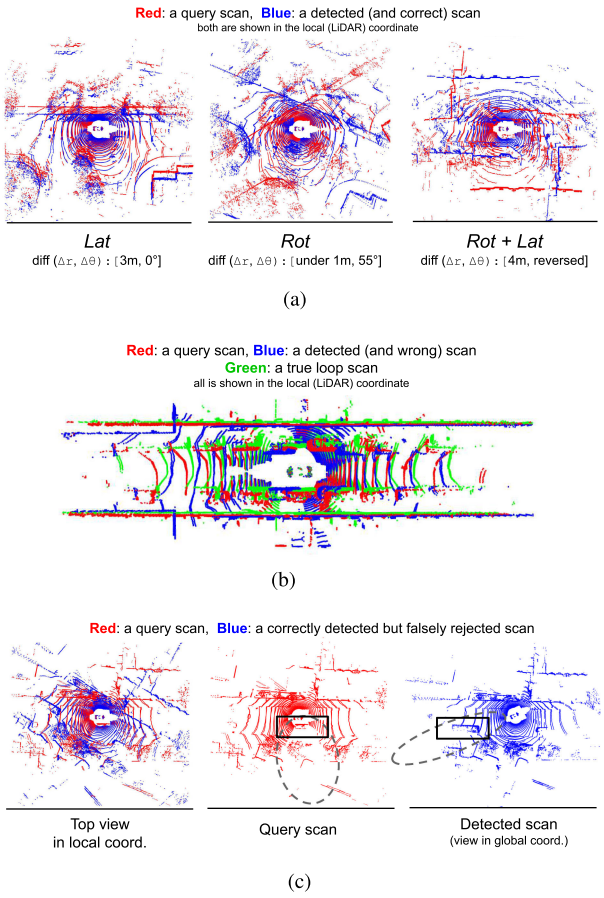


Fig. 22. (a) Successful examples in `Oxford 2019-01-11-13-24-51` acquired by A-PC. (b) and (c) Failure cases. (b) Perceptually aliased place from a corridor-like place. (c) When both query ($\sim 60°$ loss) and map ($\sim 30°$ loss) places undergo severe occlusions due to a tall and large object nearby (dotted ellipses), a quarter of the entire scan overlap is lost deteriorating localization capability.

reported in [2] and [53], the descriptor is learnable and provides meaningful performance by only using a small network. This type of approach would particularly be beneficial when GPU is available and the localization is almost a memorization problem.

3) Application to Nonurban Environment: The proposed method is most powerful in an urban environment where the descriptor can encode the nearby structural variance. The proposed descriptor is 1-channel with height value but easily expandable. For example, [42] considered LiDAR intensity value as an additional channel to successfully operate in an indoor environment. Combining deep learning with indoor application yielded meaningful performance with dense pedestrian traffic [54]. We think incorporating point cloud distribution or semantic labels as additional channels would further enhance the scan context beyond the urban environment such as indoor and natural environments.

4) Application to Other Range Sensors: The proposed descriptor is not limited to LiDAR sensors but also applicable to general range sensors including radars. As we reported a potential extension to radar sensors in [34], the descriptor can be applied to radars.

5) Generalizability Over Measurement Variation: Future studies examining the sensor difference between the mapping and localization phase would also be meaningful. LiDAR measurement varies depending on the hardware choice and mounting configuration. Achieving generalizability over measurement variation would be needed.

IX. Conclusion

In this article, we presented a global place recognition module combining topological and metric localization. As a global localizer, the proposed method can be a solution to a kidnapped robot problem serving as a place recognizer at a *wake-up* phase. We also showed the invariance of *Scan Context++* in both the rotational and lateral directions. Via the evaluation, we validated that the proposed localizer achieved discriminability and real-time performance without necessitating prior knowledge.

References

[1] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Intl. Conf. Intell. Robots Sys.*, 2018, pp. 4802–4809.
[2] G. Kim, B. Park, and A. Kim, "1-day learning, 1-year localization: Long-term LiDAR localization using scan context image," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1948–1955, Apr. 2019.
[3] S. Saeedi, M. Trentini, M. Seto, and H. Li, "Multiple-robot simultaneous localization and mapping: A review," *J. Field Robot.*, vol. 33, no. 1, pp. 3–46, 2016.
[4] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with fab-map 2.0," *Int. J. Robot. Res.*, vol. 30, no. 9, pp. 1100–1123, 2011.
[5] M. J. Milford and G. F. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2012, pp. 1643–1649.
[6] S. Lowry *et al.*, "Visual place recognition: A survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.
[7] D. Gálvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.
[8] T. Cieslewski, E. Stumm, A. Gawel, M. Bosse, S. Lynen, and R. Siegwart, "Point cloud descriptors for place recognition using sparse visual information," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 4830–4836.
[9] Y. Kim, J. Jeong, and A. Kim, "Stereo camera localization in 3 d LiDAR maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Sys.*, 2018, pp. 1–9.
[10] J. Mo and J. Sattar, "A fast and robust place recognition approach for stereo visual odometry using LiDAR descriptors," 2020, *arXiv:1909.07267*.
[11] A. Oertel, T. Cieslewski, and D. Scaramuzza, "Augmenting visual place recognition with structural cues," 2020, *arXiv:2003.00278*.
[12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.
[13] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[14] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.
[15] C. Cadena, D. Gálvez-López, J. D. Tardós, and J. Neira, "Robust place recognition with stereo sequences," *IEEE Trans. Robot.*, vol. 28, no. 4, pp. 871–885, Aug. 2012.
[16] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford, "On the performance of convnet features for place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 4297–4304.
[17] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog*nit., 2016, pp. 5297–5307.
[18] G. D. Tipaldi and K. O. Arras, "FLIRT - interest regions for 2D range data," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2010, pp. 3616–3622.

[19] G. D. Tipaldi, L. Spinello, and W. Burgard, "Geometrical flirt phrases for large scale place recognition in 2 d range data," in *Proc. IEEE Int. Conf. Robot. Autom*., 2013, pp. 2693–2698.
[20] E. B. Olson, "Real-time correlative scan matching," in *Proc. IEEE Int. Conf. Robot. Autom*., 2009, pp. 4387–4393.
[21] E. Olson, "Recognizing places using spectrally clustered local matches," *Robot. Auton. Syst.*, vol. 57, no. 12, pp. 1157–1172, 2009.
[22] B. Steder, G. Grisetti, and W. Burgard, "Robust place recognition for 3D range data based on point features," in *Proc. IEEE Int. Conf. Robot. Autom*., 2010, pp. 1400–1405.
[23] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., 2010, pp. 2155–2162.
[24] M. Bosse and R. Zlot, "Place recognition using keypoint voting in large 3D lidar datasets," in *Proc. IEEE Int. Conf. Robot. Autom*., 2013, pp. 2677–2684.
[25] L. He, X. Wang, and H. Zhang, "A novel point cloud descriptor and its application in loop closure detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., 2016, pp. 231–237.
[26] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., 2018, pp. 4470–4479.
[27] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong, "3 d lidar-based global localization using siamese neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1380–1392, Apr. 2020.
[28] 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, 2020.
[29] 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. Autom*., 2017, pp. 5266–5272.
[30] L. Schaupp, M. Bürki, R. Dubé, R. Siegwart, and C. Cadena, "OREOS: Oriented recognition of 3D point clouds in outdoor scenarios," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., 2019, pp. 3255–3261.
[31] X. Chen *et al.*, "OverlapNet: a siamese network for computing LiDAR scan similarity with applications to loop closing and localization," *Autonom. Robots*, pp. 1–21, 2021.
[32] H. Yin, L. Tang, X. Ding, Y. Wang, and R. Xiong, "LocNet: Global localization in 3D point clouds for mobile vehicles," in *Proc. IEEE Intell. Veh. Symp.*, 2018, pp. 728–733.
[33] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar RobotCar dataset: A radar extension to the oxford RobotCar dataset," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 6433–6438.
[34] G. Kim, Y. S. Park, Y. Cho, J. Jeong, and A. Kim, "MulRan: Multimodal range dataset for urban place recognition," in *Proc. IEEE Intl. Conf. Robot. Autom*., 2020, pp. 6246–6253.
[35] D. Barnes and I. Posner, "Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar," in *Proc. IEEE Int. Conf. Robot. Autom*., 2020, pp. 9484–9490.
[36] Ş. Săftescu, M. Gadd, D. De Martini, D. Barnes, and P. Newman, "Kidnapped radar: Topological radar localisation using rotationally-invariant metric learning," 2020, *arXiv:2001.09438.*
[37] M. Gadd, D. De Martini, and P. Newman, "Look around you: Sequence-based radar place recognition with learned rotational invariance," 2020, *arXiv:2003.04699*.
[38] Z. Hong, Y. Petillot, and S. Wang, "RadarSLAM: Radar based large-scale SLAM in all weathers," 2020, *arXiv:2005.02198.*
[39] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *Int. J. Robot. Res.*, vol. 30, no. 13, pp. 1543–1552, 2011.
[40] M. L. Benedikt, "To take hold of space: Isovists and isovist fields," *Environ. Plan. B: Plan. Des.*, vol. 6, no. 1, pp. 47–65, 1979.
[41] G. Kim, A. Kim, and Y. Kim, "A new 3 d space syntax metric based on 3 d isovist capture in urban space using remote sensing technology," *Comput. Environ. Urban Syst.*, vol. 74, pp. 74–87, 2019.
[42] H. Wang, C. Wang, and L. Xie, "Intensity scan context: Coding intensity and geometry relations for loop closure detection," in *Proc. IEEE Int. Conf. Robot. Autom*., 2020, pp. 2095–2101.
[43] A. Licha, "Sejong smart city: On the road to be a city of the future," in *Proc. Int. Conf. Comput. Urban Plann. Urban Manag.*, 2019, pp. 17–33.
[44] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford RobotCar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.
[45] N. Sünderhauf and P. Protzel, "Switchable constraints for robust pose graph SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., 2012, pp. 1879–1884.
[46] H. Schütze, C. D. Manning, and P. Raghavan, *Introduction to Information Retrieval*. Cambridge, U.K.: Cambridge University Press, 2008.
[47] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Mach. Learn. Res.*, vol. 70, 2017, pp. 214–223.
[48] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, "Robust map optimization using dynamic covariance scaling," in *Proc. IEEE Int. Conf. Robot. Autom*., 2013, pp. 62–69.
[49] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *Proc. IEEE Int. Conf. Robot. Autom*., 2020, pp. 1689–1696.
[50] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., 2018, pp. 4758–4765.
[51] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the Bayes tree," *Int. J. Robot. Res.*, vol. 31, no. 2, pp. 216–235, 2012.
[52] K. Li, M. Li, and U. D. Hanebeck, "Towards high-performance solid-state-lidar-inertial odometry and mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5167–5174, Jul. 2021.
[53] X. Xu, H. Yin, Z. Chen, Y. Li, Y. Wang, and R. Xiong, "Disco: Differentiable scan context with orientation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2791–2798, Apr. 2021.
[54] M. Y. Chang, S. Yeon, S. Ryu, and D. Lee, "Spoxelnet: Spherical voxel-based deep place recognition for 3 d pointclouds of a crowded indoor space," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst*., 2020, pp. 8564–8570.

Giseop Kim (Student Member, IEEE) received the B.S. and M.S. degrees in civil and environmental engineering from the Korea Advanced Institute of Science & Technology (KAIST), Daejeon, Korea, in 2017 and 2019, respectively. He is currently working toward the Ph.D. degree with the Department of Civil and Environmental Engineering, KAIST.

His research interests include LiDAR simultaneous localization and mapping and long-term map management.

Sunwook Choi received the B.S. degree in electronic engineering, the M.S. degree in information and communication engineering, and the Ph.D. degree in electronic engineering from Inha University, Incheon, Korea, in 2007, 2009, and 2014, respectively.

From 2014 to 2017, he was a Research Engineer with Cognitive Computing Group (Deep Learning Research Lab.), NAVER Corp., Seongnam-si, Korea. Since 2017, he has been with the Autonomous Driving Group, NAVER LABS, Seongnam-si, where he is a Senior Research Engineer. His research interests include simultaneous localization and mapping, perception for autonomous driving, and deep learning for computer vision.

Ayoung Kim (Member, IEEE) received the B.S. and M.S. degrees in mechanical engineering from Seoul National University (SNU), Seoul, Korea, in 2005 and 2007, respectively, and the M.S. degree in electrical engineering and the Ph.D. degree in mechanical engineering from the University of Michigan (UM), Ann Arbor, MI, USA, in 2011 and 2012, respectively.

She was an Associate Professor with the Department of Civil and Environmental Engineering with joint affiliation at KI robotics, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea, from 2014 to 2021. Currently, she is an Associate Professor with the Department of Mechanical Engineering, SNU. Her research interests include visual simultaneous localization and mapping, and navigation.