C2L-PR: Cross-Modal Camera-to-LiDAR Place Recognition via Modality Alignment and Orientation Voting

Huaiyuan Xu, Huaping Liu, *Senior Member, IEEE*, Shoudong Huang, *Senior Member, IEEE*, and Yuxiang Sun, *Member, IEEE*

***Abstract*—Place recognition is a fundamental technology for vehicle localization. LiDAR-based methods could work under visual appearance-changing conditions, such as season or weather changes, and different times of a day. However, these methods require every vehicle to be equipped with a 3-D LiDAR during the online localization stage, resulting in high costs for the vehicles. To alleviate this issue, we propose a cross-modal place recognition network, which can localize vehicles with visual images obtained from a low-cost monocular camera against a pre-built LiDAR point-cloud database. To this end, we first bridge the modality gap between visual images and point clouds via modality alignment. Then, we propose an orientation voting module to suppress the recognition ambiguity caused by the inconsistent field-of-view between images and point clouds, thereby improving the place recognition accuracy. Experiments are conducted with three public datasets: KITTI, KITTI-360, and Oxford RobotCar, covering over 71.6 KM of vehicle trajectories in 12 urban and suburban regions in two countries. The results demonstrate the superiority of our network.**

***Index Terms*—Cross modality, place recognition, global localization, modality alignment, autonomous vehicles.**

I. Introduction

PLACE recognition is a challenging task for autonomous vehicles. It is commonly formulated as a retrieval problem [1], [2], [3], [4], [5], that is, given an observation from the current place, the algorithms retrieve the matched place from a pre-built database that consists of sensor data collected from previously visited reference places. Place recognition serves as a key component for autonomous vehicle applications, such as localization and mapping [6], [7], [8], [9], [10]. According to the type of used sensors, the existing place recognition methods can be generally classified into two classes: vision-based methods [11], [12], [13], [14], [15], [16] and LiDAR-based methods [17], [18], [19], [20], [21].

The typical pipeline of the vision-based methods is: candidate places for a query image are first retrieved from a database by the nearest-neighbor search, then the candidate places are further re-ranked to find the best matched one with geometric verification [12], [13]. For the nearest-neighbor search, existing methods compare the encoded global descriptors of images by clustering [22], [23] or pooling [24] local descriptors (e.g., patch or keypoint features of images). For geometric verification, it is usually achieved by the spatial matching of local descriptors between images [12]. However, due to the intrinsic limitations of visual cameras, vision-based place recognition (VPR) struggles to reliably work in visual-appearance changing environments [25]. For example, the changes in illumination and weather, like day/night and sunny/rainy, can cause significant visual-appearance differences between two images even if they are taken at the same place.

In contrast, LiDAR point clouds are more robust to visual-appearance changes. The LiDAR-based methods use hand-crafted features [26], [27] or learned features [1], [28] from LiDAR point clouds to evaluate the similarity of places. They have been widely studied in the field of autonomous driving and navigation in large-scale outdoor environments. However, LiDAR-based place recognition (LPR) requires every robot to be equipped with a 3-D LiDAR sensor, thus leading to high costs for vehicles.

To alleviate this issue, in this paper, we propose a cross-modal place recognition network, named C2L-PR, which realizes place recognition by matching a query image from a low-cost on-vehicle monocular camera to a pre-built geo-referenced point-cloud database (see Fig. 1). To this end, images are first mapped into semantic point clouds so that the descriptors of images and point clouds can be extracted from a similar modality. Secondly, the point cloud is split into slices with different field-of-views (FOVs), and then an orientation voting module is proposed to perform slice classification to select the best-matched slice of which FOV is closest to the semantic point cloud. Finally, the

Manuscript received 16 April 2024; revised 15 June 2024; accepted 21 June 2024. Date of publication 4 July 2024; date of current version 31 July 2025. This work was supported in part by Hong Kong Research Grants Council under Grant 15222523, and in part by City University of Hong Kong under Grant 9610675. *(Corresponding author: Yuxiang Sun.)*

Huaiyuan Xu is with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: huaiyuan.xu@polyu.edu.hk).

Huaping Liu is with the Department of Computer Science and Technology, Institute for Artificial Intelligence, Tsinghua University, Beijing 100190, China (e-mail: hpliu@tsinghua.edu.cn).

Shoudong Huang is with the Robotics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: shoudong.huang@uts.edu.au).

Yuxiang Sun is with the Department of Mechanical Engineering, City University of Hong Kong, Kowloon, Hong Kong (e-mail: yx.sun@cityu.edu.hk).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2024.3423392.

Digital Object Identifier 10.1109/TIV.2024.3423392

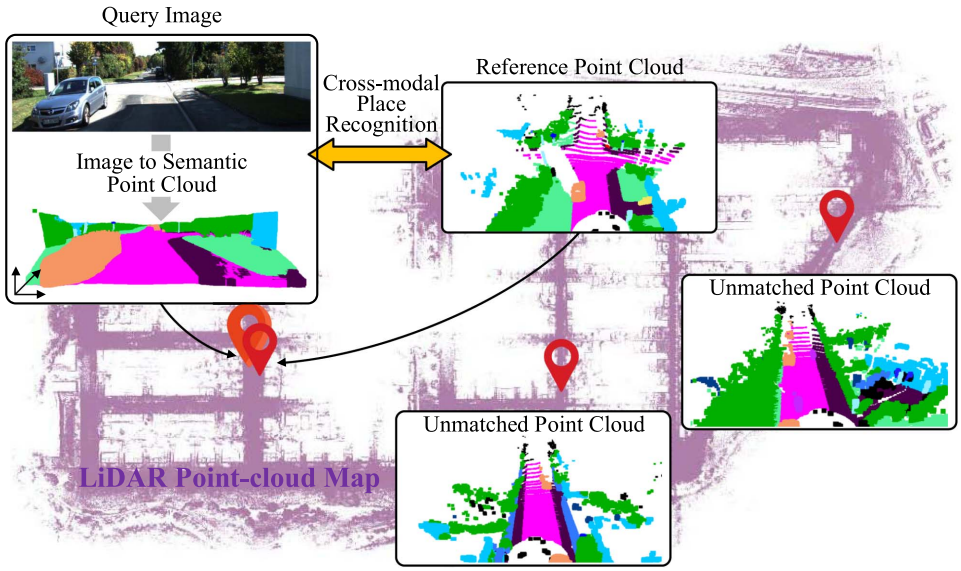


Fig. 1. The figure illustrates our idea of cross-modal place recognition with a visual image against a pre-built LiDAR point-cloud database. Given an on-line query image captured by a vehicle-mounted monocular camera, a semantic point cloud is generated from the image to alleviate the modality gap to find the matched reference point cloud in the database.

similarity between the query place and the reference place is obtained by comparing the semantic point cloud and the point-cloud slice.

We evaluate our cross-modal place recognition performance in 12 urban and suburban areas of two countries from three public datasets: KITTI [29], KITTI-360 [30], and Oxford RobotCar [31]. Two baselines are built upon the well-known NetVLAD [22] and PointNetVLAD [28] techniques. Extensive experiments demonstrate that our C2L-PR outperforms the baselines, baseline variants, and other state-of-the-art camera-to-LiDAR methods, in terms of multiple evaluation metrics. Comprehensive efficiency analysis and ablation studies illustrate the practicality of our C2L-PR and the effectiveness of the network structures, loss functions, and training data. Our contributions are summarized as follows:

1) We introduce a novel cross-modal place recognition network by matching on-line visual images against a pre-built off-line LiDAR point-cloud database. Our code and video demo are available on the Github page[1].
2) We propose a novel modality alignment module to bridge the modality gap between visual images and LiDAR point clouds, so that the two different modalities of data can be compared.
3) We design a FOV selection module to reduce the ambiguities caused by the inconsistent FOV between visual images and point clouds by learning descriptor comparison and orientation voting.
4) We create two baselines and compare them with our network on three public datasets, covering a total of more than 71.6 KM of vehicle trajectories from two countries. The experimental results demonstrate our superiority over the baselines and other camera-to-LiDAR approaches.

[1]Our code and video demo: https://github.com/lab-sun/C2L-PR.

The remainder of this paper is structured as follows. Section II reviews the related work. Section III presents the details of our proposed network. Section IV discusses the experimental results. Conclusions and future work are drawn in the last section.

II. Related Work

A. Vision-Based Place Recognition

Traditional VPR methods are typically based on the bag-of-words (BoW) algorithm [32], which clusters hand-crafted local features of places [33], [34] into a series of visual words, and represents places with statistical word frequency histograms. Besides BoW, another mainstream solution is the vector of locally aggregated descriptors (VLAD) [35], [36], which preserves the distance of local features to cluster centers and thus provides more details when describing places. Further, NetVLAD [22] achieves increased robustness to visual appearance changes by reorganizing VLAD in a data-driven manner. More recently, some researchers resort to learning local and global features jointly to represent a place. For example, Patch-NetVLAD [12] simultaneously learns image-level and patch-level features using NetVLAD. TransVPR [13] improves Patch-NetVLAD by detecting task-related patches and then integrating them into place representations. Besides, other researchers have verified that using maps can enhance VPR performance [37]. Due to the intrinsic limitations of visual cameras, these VPR methods all suffer from performance degradation in illumination-changing environments.

B. LiDAR-Based Place Recognition

Compared with visual cameras, 3-D LiDARs are robust to illumination changes. Scan context (SC) [38] is a hand-crafted descriptor to represent LiDAR point clouds. It reserves the maximum height of points in each segmented point cloud block.

Although SC only encodes the geometric information, it has been proven to be discriminative in various scenes. Some researchers further improved SC by adding intensity information [39] and semantic context [26]. With deep learning, the first end-to-end descriptor, PointNetVLAD [28], was proposed. It employs deep neural networks to encode a point cloud into high-dimensional features, and then the features are fed to the NetVLAD module for feature aggregation. There are also some descriptors that mix hand-crafted and learnable features, such as MinkLoc3D-SI [18] and RINet [19].

C. Multi-Modal Place Recognition

Using multi-modal data fusion in place recognition has demonstrated enhanced performance compared to single-modal methods [40], [41], [42]. Oertel et al. [40] respectively extracted 2-D and 3-D features from images and point clouds, and then fed them to fully connected layers after concatenation to achieve multi-modal information fusion. Furthermore, Lai et al. [41] designed an additional weight branch to evaluate the contributions of image and point-cloud features, leading to adaptive data integration. In addition to merging modalities at the feature level, Bernreiter et al. [42] demonstrated that multi-modal information fusion can also be performed in the original input. Recently, Wang et al. [43] distilled knowledge from other modalities into a student single-modal network, so that the single-modal network can achieve performance close to that of multi-modal methods.

D. Cross-Modal Place Recognition

Different from multi-modal place recognition that fuses data from multiple modalities to enhance the performance, cross-modal place recognition [44], [45], [46] retrieves matched places with a modality of data against a database built with the other modality of data. Recently, there is an increasing interest in cross-modal place recognition. Mithun et al. [47] explored place recognition from cameras to aerial LiDAR, Samano et al. [48] realized image-to-OpenStreetMap place recognition, and Cattaneo et al. [49] achieved visual localization on a 3-D map by training joint descriptors between images and point clouds. However, these methods either require that the query and reference data have the same FOV [47], [48] or do not consider FOV variations [49]. Different from these methods, we mitigate the impact of FOV inconsistency between the monocular camera and LiDAR by selecting an appropriate FOV.

The major challenge for cross-modal methods is how to reduce the modality gap between two modals of data, so that they can be compared and matched. The *Align-to-2D* strategy converts 3-D point clouds to 2-D images by spherical projection [1], [50] or rendering [47], and compares them in 2-D. However, the *Align-to-2D* strategy could project multiple 3-D points onto the same pixel and some pixels may have no projected 3-D point. Different from *Align-to-2D*, we adopt the *Align-to-3D* strategy, which converts different modalities of data into the same 3-D semantic point cloud modality, so that different modalities can be compared. In our method, each pixel has estimated depth, so we can convert input 2-D images into 3-D point clouds.

Compared to single-modal vision-based place recognition [13], [22], the database of cross-modal camera-to-LiDAR place recognition presents greater robustness. The image databases of VPR methods are sensitive to environmental conditions, such as changes in illumination, weather, and seasons. In contrast, by leveraging LiDAR sensing, the point-cloud database of our C2L-PR is robust to condition changes, meanwhile providing precise geometric details of the environment. Moreover, compared to single-modal LiDAR-based place recognition [26], [28] and multi-modal place recognition (MMPR) [40], [41], our C2L-PR is cost-effective. LPR and MMPR require each vehicle to be mounted with an expensive LiDAR sensor, and even more types of sensors. In contrast, our cross-modal solution only requires deploying a cheap camera in each vehicle, thereby reducing the manufacturing cost for vehicle equipment manufacturers and being friendly to the industry.

III. The Proposed Method

A. Method Overview

Our problem is formulated as follows: given two places that are respectively represented by an online captured RGB image from a vehicle-mounted camera, and an offline LiDAR point cloud from a pre-built database, our task is to measure the similarity between the two modalities of data to determine whether they are from the same place. The major challenges here are the large modality gap between image and point cloud, as well as the large difference in terms of FOV between the two modals of data, making them hard to be directly compared.

Fig. 2 provides the overview of our method. We can see that our C2L-PR mainly consists of two modules: modality alignment and FOV selection. The two modules are designed to address two challenges: 1) The modality alignment module reduces the modality gap between the two data by converting them to a similar modality (i.e., semantic point cloud); 2) The FOV selection module mitigates the FOV difference by selecting the best-matched FOV.

B. Modality Alignment

Our modality alignment adopts the *Align-to-3D* strategy, which aligns both the two input modalities into the same 3-D semantic point cloud modality. Specifically, for the input RGB image, we first calculate pixel depth by using the monocular depth estimation network [51], and obtain the semantic segmentation map via the semantic image segmentation network [52]. With the calculated depth data and the prior-known camera intrinsic parameters, we apply the pinhole camera model to construct a point cloud, in which each point corresponds to a pixel that has valid depth information. Then, we overlay the point cloud with the semantic segmentation map to produce the 3-D semantic point cloud.

For the input LiDAR point cloud, we find the semantic segmentation map by using semantic point-cloud segmentation networks. However, in our task, the input LiDAR point cloud is from an offline pre-built database, so it is feasible to use hand-labeled semantic ground truth instead of using semantic segmentation. So, we directly overlay the ground truth on the 3-D point clouds to construct 3-D semantic point clouds for the input LiDAR point-cloud data.

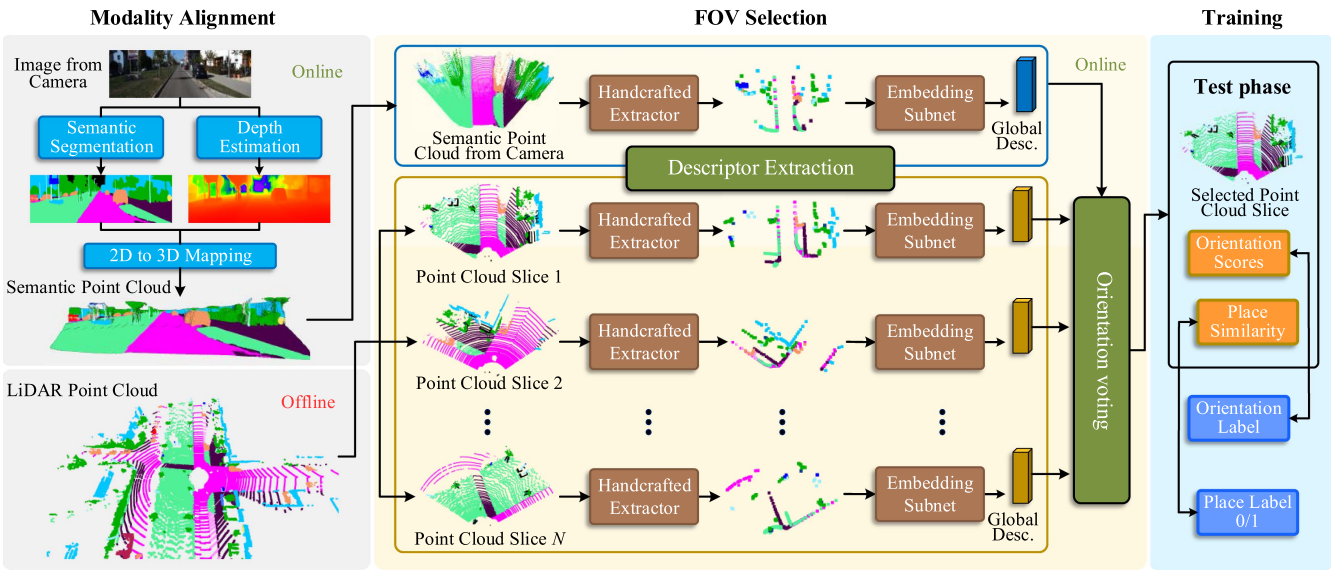


Fig. 2. The pipeline of our proposed C2L-PR. It mainly consists of the modality alignment module and the FOV selection module. A LiDAR point-cloud database is first pre-built offline. Then, our C2L-PR finds the place from the database by querying an online captured RGB image.

C. FOV Selection

The 3-D LiDAR has 360° FOV, while the camera has around 90° FOV [29]. Considering that only overlapped views contribute to place recognition, we propose an FOV selection module to select the best-matched FOV.

1) Point Cloud Slicing: Given a LiDAR point cloud $\mathcal{P}$ and a view angle φ for slicing, we can obtain $N = 2\pi/\varphi$ point-cloud slices. The view angle θ_p of a point p in a point-cloud slice $\mathcal{P}_i$ is within the range $\varphi(i-1) \leq \theta_p < \varphi i,\ i = 1, \ldots, N$, where i is the index of the slice. However, this naive slicing strategy has a major issue, that is, there might be no slice containing the complete FOV of the camera, if the slice does not happen to cover the camera FOV, as shown in the left part of Fig. 3. As a result, the LiDAR point-cloud slice can only be compared with part of the data from the camera, making the image data not fully utilized.

To alleviate this issue, we propose a partially overlapping slicing strategy, in which two adjacent point-cloud slices overlap with each other by $\varphi/2$. In this way, there would be a higher chance for at least one of our slices to cover the camera FOV as much as possible. We can now get $N = 4\pi/\varphi$ slices. The view angle θ_p of point p is within:

$$\begin{cases} \frac{\varphi}{2} \cdot (i-1) \leq \theta_p < \frac{\varphi}{2} \cdot (i+1), i = 1, \ldots, N-1 \\ 2\pi - \frac{\varphi}{2} \leq \theta_p < 2\pi \ \text{ or } \ 0 \leq \theta_p < \frac{\varphi}{2},\ i = N \end{cases}. \tag{1}$$

2) Descriptor Extraction: We detect the semantic contour of a point cloud as the object feature, and then learn a global descriptor for place recognition. Specifically, the object feature is extracted based on object classes and distances in a handcrafted manner [19]. Given a point-cloud slice or semantic point cloud, we divide it into S smaller sectors by equally dividing the azimuths from the bird-eye view. We select C semantic classes. For each class in each sector, we only keep the nearest distance between the points and the point-cloud center to build an object feature. For example, a point-cloud slice with a 180° FOV can be divided into 180 sectors with 1° for each. Considering that we have C semantic classes, from each sector, we can extract a vector $v_b \in \mathbb{R}^{C\times 1}$, in which v_{ab} represents the nearest Euclidean distance from the points of class a in sector b to the point-cloud center. We concatenate these 1-D vectors into a 2-D array as the object feature $F \in \mathbb{R}^{C\times S}$:

$$F = \{v_{ab} | a = 1, \ldots, C \text{ and } b = 1, \ldots, S\} \in \mathbb{R}^{C\times S}. \tag{2}$$

Note that $v_{ab} = 0$ if there is no class a in sector b. Then, we feed the constructed object feature to the embedding sub-network to generate a global descriptor.

Fig. 4 shows the embedding sub-network, which uses a learnable multi-layer structure to encode the object feature. Each layer has a 1-D convolution layer, BatchNorm, and an attention block to extract a feature $F_x \in \mathbb{R}^{h_x \times w_x}$, where x represents the layer index. The output of the multi-layer structure is the global descriptor which is obtained by aggregation and concatenation of the features. We learn a linear projection for aggregation. A linear projection refers to a linear transformation for the feature by multiplying a projection factor $\mathcal{T}_x \in \mathbb{R}^{1\times w_x}$. So, the feature array needs to be transposed before the linear projection. After

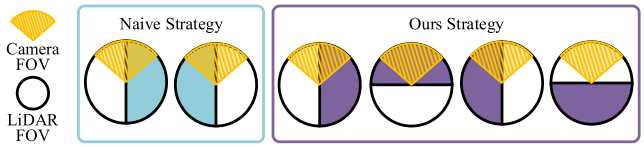


Fig. 3. Comparison of two different LiDAR point-cloud slicing strategies, taking the view angle $\varphi = 180°$ for slicing as an example. The naive slicing strategy gets 2 LiDAR point-cloud slices, but neither covers the full FOV of the camera. In contrast, our proposed strategy can obtain 4 slices, by which adjacent slices partially overlap. So, there would be a higher chance for one of our slices to cover the camera FOV as much as possible.

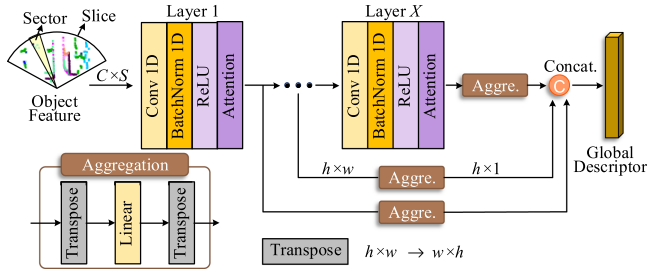


Fig. 4. The structure of learning-based embedding. Given an object feature, it outputs a global descriptor. The embedding network consists of X layers. The size $h \times w$ of layer output varies with different layers. Each output is aggregated through a trainable aggregation block to yield a $h \times 1$ vector. Then, these vectors are concatenated to form a global descriptor.

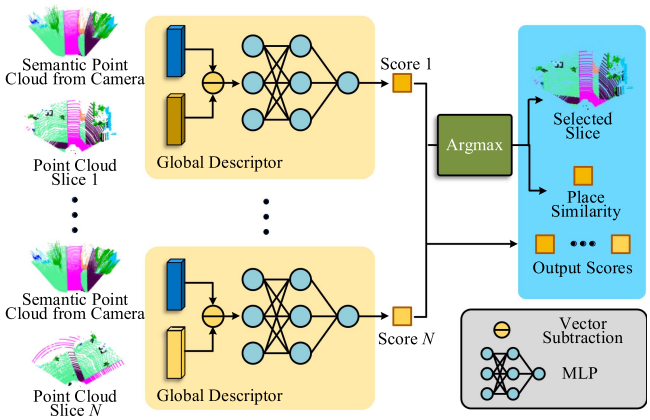


Fig. 5. The structure of the orientation voting module. It learns to identify the class corresponding to the closest FOV to the camera. N classes correspond to N different FOVs. The output of the multi-layer perceptron (MLP) is the maximum value of the matching scores, which is used as place similarity.

each aggregation, a descriptor D_x is obtained:

$$D_x = \left(\mathcal{T}_x\left(F_x\right)^T\right)^T \in \mathbb{R}^{h_x \times 1}. \tag{3}$$

We further concatenate these descriptors to get the final place descriptor $D \in \mathbb{R}^{(\sum_{x=1}^{X} h_x)\times 1}$. Such object-based representation extraction combines the advantages of handcrafted descriptors and learning-based descriptors. The former perceives the geometric and semantic context of the scene, while the latter learns a place descriptor in a data-driven manner.

3) Orientation Voting: Among the N point-cloud slices, we find the best one of which the FOV is closest to the semantic point cloud derived from the image. To this end, we propose the orientation voting module, shown in Fig. 5. It treats voting as a classification task, that is, regarding N point-cloud slices corresponding to different FOVs as N classes. The module learns to recognize the correct class, which corresponds to the closest camera FOV. Let D denote the descriptor of the semantic point cloud from the image, and D_i denote the descriptor of the point-cloud slice, where i is the index of the FOV class. For the i-th class, the L-1 distance between D and D_i is passed through a shared multi-layer perceptron (MLP) to obtain a score representing the likelihood of the semantic point cloud belonging to this class. Then, the optimal class c is obtained by taking the argmax function over all scores:

$$c = \arg\max_i(\text{MLP}(||D, D_i||_1)), \tag{4}$$

where the class c corresponds to the best slice $\mathcal{P}_c$.

The computational complexity of our algorithm scales linearly with the number of point-cloud slices N. Equations (1) and (2) are independent of N. Since adjacent slices overlap each other by half, each 3-D point appears in only two slices and is processed twice, meaning that the computational complexity of (1) and (2) depends on the number of points rather than the number of slices. Equations (3) and (4) are related to N. N slices correspond N times of descriptor extraction using (3), and N times of descriptor comparison in (4). So, the computational complexity of our algorithm in terms of the number of slices is $\mathcal{O}(N)$.

IV. Experimental Results and Discussions

A. Implementation Details

Our C2L-PR network is implemented with PyTorch and trained on a 12 GB NVIDIA GeForce RTX 3060 GPU. In the modality alignment module, we use a semantic segmentation network [52] and a depth estimation network [51] with pre-trained weights on the KITTI dataset [29] to generate semantic point clouds from camera images. In the FOV selection module, we extract the object feature within 50 m from the point-cloud center, and with 12 semantic classes, including vegetation, sidewalk, building, fence, trunk, car, terrain, other ground, pole, traffic sign, road, and parking. The embedding sub-network consists of 6 layers. The attention block employs a 1-D convolution to compute the importance of each semantic class. Furthermore, we separate the semantic point cloud into 8 overlapping slices, each with a 90° FOV.

B. Datasets

We train and evaluate our C2L-PR on three public datasets: 1) The KITTI dataset [29], which consists of 11 sequences (00 to 10). We select one of 5 sequences (00, 02, 05, 06, 07) as the testing set, while the remaining 10 sequences are used for training and validation. Each sequence is recorded in RGB images and LiDAR point clouds, and has ground-truth poses. The semantic labels of LiDAR point clouds come from the semantic KITTI dataset [53]; 2) The KITTI-360 dataset [30], which contains 11 sequences covering over 73.7 KM in several suburban areas. Its sequence length and total number of loop closures are approximately two times and seven times than those of the KITTI dataset, which poses a greater challenge. To analyze the generalizability across different datasets, we train our network on KITTI and then test it on KITTI-360 without any fine-tuning. Fig. 6 and Table I present the statistical comparison between the two datasets, from which we can find that KITTI-360 has a much larger scale than KITTI; 3) The Oxford RobotCar dataset [31], which contains a 10 KM data collection route through the urban environments in central Oxford. This route is divided into training/validation/testing sets in a ratio of

TABLE I
STATISTICS OF KITTI AND KITTI-360

Dataset	KITTI					KITTI-360					
Trajectory	00	02	05	06	07	00	02	04	05	06	09
Number of Images	4541	4661	2761	1101	1101	11518	14607	11587	6743	9699	14056
Number of Scans	4541	4661	2761	1101	1101	11465	11831	8427	6539	9568	13836
Number of Loops	804	315	448	270	57	2555	2496	1592	1990	2499	4844

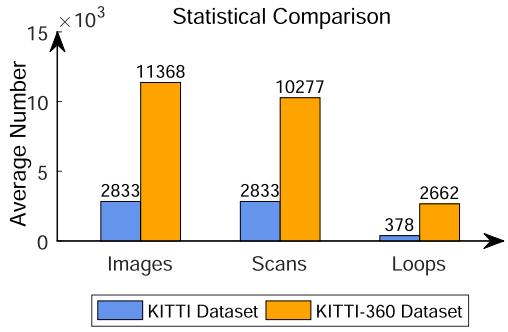


Fig. 6. Statistical comparison between the KITTI dataset and the KITTI-360 dataset. The scale of KITTI-360 is much larger than KITTI. The average number is the average image/scan/loop number for all trajectories in Table I.

4:1:1, where the testing area is spatially non-overlapping with the training and validation areas. We utilize KITTI for pre-training and fine-tune the network on Oxford RobotCar to validate the effectiveness of our C2L-PR across different datasets.

C. Evaluation Metrics

We adopt the maximum F1 score and Recall@K to evaluate the performance of cross-modal place recognition [19]. A predefined parameter ε is used to threshold the place similarity scores to generate the recall value (Rec) and precision value (Pre) [54]. An F1 score is calculated as [54], [55]:

$$\mathrm{F1} = 2 \times (\mathrm{Rec} \times \mathrm{Pre}) \,/\, (\mathrm{Rec} + \mathrm{Pre}), \tag{5}$$

which indicates the ability of the algorithm to balance precision and recall. We here take the maximum value of F1 scores.

The Recall@K metric counts the proportion of correctly retrieved places. One correct place recognition is defined as at least one of the K retrieved places whose distance d_{mn} from the ground-truth place is within a threshold τ.

$$\mathrm{Recall@}K = \frac{1}{M} \sum_{m=1}^{M} \Lambda \left(\exists \, d_{mn} < \tau, n = 1, \ldots, K \right), \tag{6}$$

where M is the number of queries, and Λ is a function that returns 1 if the variable is true, and 0 otherwise. τ is set to 5 m.

D. Loss Functions

We treat our task as a classification problem. Given an input place pair (i.e., a pair of an image and a LiDAR point cloud), we compute binary cross-entropy using the predicted place similarity s_t and label y_t. The binary cross-entropy loss $\mathcal{L}_{bce}$ is the average over all samples:

$$\mathcal{L}_{bce} = \frac{1}{\Pi} \sum_{t} - \left[y_t \log\left(s_t\right) + \left(1 - y_t\right) \log\left(1 - s_t\right) \right], \tag{7}$$

where Π is the number of input samples. $y_t = 1$ for positive samples, and $y_t = 0$ for negative samples. Note that a positive sample refers to a place pair in which two places are less than 3 m, while a negative sample is greater than 20 m. The ratio of the positive and negative samples for training is $1:1$.

In addition, we calculate the cross-entropy loss $\mathcal{L}_{ce}$ on $|\mathrm{Pos}|$ positive place pairs to train FOV selection:

$$\mathcal{L}_{ce} = \frac{1}{|\mathrm{Pos}|} \sum_{j} \sum_{i=1}^{N} -y_{ij} \log\left(s_{ij}\right), \tag{8}$$

where for the positive place pair j, y_{ij} is the ground-truth label showing whether the point-cloud slice i is the best slice. s_{ij} is a prediction score. Furthermore, the distance loss $\mathcal{L}_{dis}$ is computed to force the descriptor distance d_t of a positive place pair to be closer and that of a negative place pair to be farther:

$$\mathcal{L}_{dis} = \frac{1}{\Pi} \sum_{t} (y_t \cdot d_t^2 + (1 - y_t) \cdot \max(m - d_t^2)), \tag{9}$$

where d_t is the L1 norm of two descriptors and m is a constant of 0.2. The final loss is the unweighted summation of $(\mathcal{L}_{bce}, \mathcal{L}_{ce}, \mathcal{L}_{dis})$.

E. Compared Methods

According to the *Align-to-2D* and *Align-to-3D* modality alignment strategies, we construct two baseline methods: the NetVLAD-baseline and the PointNetVLAD-baseline, based on the popular single-modal methods NetVLAD [22] and PointNetVLAD [28]. The input data is converted into a similar modality before feeding them to NetVLAD and PointNetVLAD, as shown in Figs. 7 and 8. Besides the baselines, some baseline variants and three state-of-the-art camera-to-LiDAR methods are also compared with our method.

1) Baselines: The NetVLAD-baseline follows the *Align-to-2D* alignment strategy. The input LiDAR point cloud is projected spherically onto a 2-D range image and a semantic segmentation map [1], [50]. Similarly, the input RGB image is also converted into a 2-D semantic segmentation map [52]. We concatenate semantic maps with convolutional neural network (CNN) features, and then perform convolutional fusion. A further discussion is presented in Section IV-H. In this way, NetVLAD can be used to extract global descriptors for place recognition. We also evaluate several baseline variants where NetVLAD is replaced with PatchNetVLAD [12] and CosPlace [14], respectively.

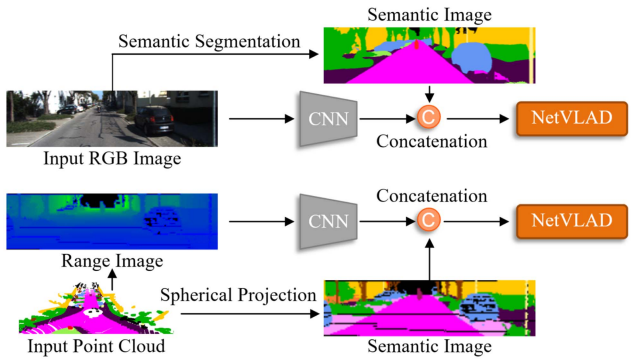


Fig. 7. The pipeline of the NetVLAD-baseline. We convert the input RGB image and LiDAR point cloud into 2-D semantic images and range images. Then, the NetVLAD network [22] is used to recognize places.

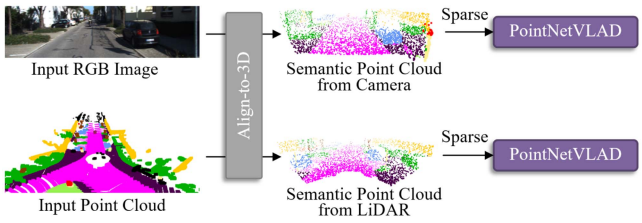


Fig. 8. Pipeline of the PointNetVLAD-baseline. We convert the input RGB image and LiDAR point cloud into semantic point clouds using our *Align-to-3D* strategy, and then sparsify them before the PointNetVLAD network [28].

In contrast, the PointNetVLAD-baseline uses our *Align-to-3D* modality alignment method. Since directly implementing PointNetVLAD to process huge point clouds could lead to high memory cost, it is necessary to perform sparsification first. Therefore, we randomly sample points from the point cloud before performing PointNetVLAD. The random sampling process adopts a downsampling filter which selects 4,096 points from the original point cloud in a uniformly distributed sampling manner. We also build a baseline variant by replace PointNetVLAD with Semantic Scan Context [26], a semantic point-cloud place recognition method. Like PointNetVLAD, it needs to first align different input modalities using our *Align-to-3D* strategy, and then compare point clouds.

In the NetVLAD-baseline, the height of the range image is determined by the highest scan of LiDAR sensing. Environmental information above this highest scan is not recorded in the range image. For this reason, in the network structure of the NetVLAD-baseline, we use the LiDAR FOV to select the part of the RGB image whose height is consistent with the range image, and then compare it with the range image. Similarly, in the structure of the PointNetVLAD-baseline, it also uses the LiDAR FOV to perform the same operation on the RGB image.

To ensure a fair comparison, the input point clouds of the NetVLAD-baseline and the PointNetVLAD-baseline also come from the pre-built point-cloud database. Their semantic labels are also given.

2) Camera-to-LiDAR Methods: We compare our C2L-PR with three recent camera-to-LiDAR place recognition methods: $(LC)^2$ [56], i3dLoc [46], and I2P-Rec [45]. To align the input modalities, $(LC)^2$ and i3dLoc transform the input data into 2-D perspective-view images. In contrast, I2P-Rec converts cross-modal data into 3-D space and performs place recognition in the bird's eye view. In the identical testing format of 2D-to-3D place recognition, our C2L-PR is compared to baselines, baseline variants, and state-of-the-art camera-to-LiDAR methods.

F. Performance Evaluation

We evaluate the methods from five aspects: cross-modal place retrieval, cross-modal positive/negative place identification, generalization, robustness across different datasets, and single-modal testing. Evaluations are conducted on the aforementioned three public datasets: KITTI, KITTI-360, and Oxford RobotCar.

1) Place Retrieval: Trajectories of an autonomous vehicle might have loop closures. If loop closure occurs, place recognition algorithms are expected to find the corresponding place from the past traveled places. We test 5 sequences (00, 02, 05, 06, 07) with loop closures from the KITTI dataset. The quantitative results in terms of the Recall@K ($K = 1, 5, 10$) metric are presented in Table II, where the smaller the N, the more strict the metric. We can see that our method outperforms comparative methods on 4 sequences, and presents comparable performance to the NetVLAD-baseline and CosPlace on the KITTI-07 sequence. From the I2P-Rec paper [45], the method cannot perform place recognition with inconsistent orientations. That is, the original I2P-Rec requires that a vehicle must pass the same place with the same direction every time, which conflicts with the reality that such a revisit might occur with an opposite direction or other directions. To handle this random direction issue, we incorporate the proposed orientation voting scheme into the I2P-Rec framework. After unifying place recognition scenarios, we can observe that our C2L-PR outperforms I2P-Rec in terms of all recall metrics. Besides, our C2L-PR considers the importance of semantics, which also leads to better performance than I2P-Rec that ignores semantic information. Fig. 9 displays sample qualitative results. The baselines retrieve wrong places in some cases, while our method can retrieve the correct places. As we can see from the first column in Fig. 9, the NetVLAD-baseline recognizes some of correct objects, such as tree trunks, but their locations are incorrect. This might be due to its lack of depth estimation for the trunks. Moreover, the PointNetVLAD-baseline can recognize large-area scene elements, such as grass, but ignores smaller elements. This might be caused by the sparsification of point clouds.

2) Positive/Negative Recognition: Given a pair of places in two modalities, cross-modal place recognition methods should accurately identify whether they are from the same place or not. So, we pick all positive place pairs and some negative place pairs with a ratio of neg-β from the testing set to evaluate this recognition capability. From the dataset, we can construct many more negative samples than positive samples. However, during training, we use the same number of positive and negative samples, such that many negative samples are not seen by the network. Thus, during testing, negative samples will be more

TABLE II
COMPARISON BETWEEN THE BASELINES IN TERMS OF RECALL@K ($K = 1, 5, 10$) (%) ON THE KITTI DATASET

Method	KITTI 00			KITTI 02			KITTI 05			KITTI 06			KITTI 07		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
NetVLAD-baseline	25.1	38.9	47.6	12.7	28.9	40.6	23.9	45.7	57.1	20.7	40.0	52.6	48.4	59.0	**63.9**
PatchNetVLAD† [12]	30.2	53.2	64.9	8.3	23.5	35.6	38.4	58.1	65.6	32.6	61.1	72.2	47.5	57.4	62.3
CosPlace† [14]	20.5	39.3	50.7	5.7	10.5	17.5	9.9	26.8	38.2	8.1	24.1	38.1	45.1	55.7	**63.9**
$(LC)^2$ [56]	31.2	59.3	63.8	15.3	29.9	39.2	36.5	56.4	66.9	28.1	52.3	68.0	40.1	55.1	60.9
PointNetVLAD-baseline	62.4	73.0	79.5	20.0	29.8	38.1	35.4	48.1	54.9	26.7	51.9	64.1	16.4	28.7	34.4
i3dLoc [46]	49.3	69.0	75.4	14.9	27.6	41.0	34.4	60.4	72.8	35.6	60.4	76.1	44.3	45.9	48.4
Semantic SC‡ [26]	56.6	72.4	78.5	26.0	43.8	51.7	31.7	57.5	65.0	32.2	56.7	68.9	43.4	60.6	62.5
I2P-Rec [45]	53.4	70.1	76.1	23.5	44.1	48.9	35.1	59.4	66.6	42.2	67.0	76.3	38.5	55.7	60.7
C2L-PR (Ours)	**66.2**	**79.6**	**83.7**	**32.7**	**46.7**	**54.9**	**38.8**	**61.0**	**73.8**	**44.4**	**70.0**	**78.1**	**51.6**	**60.7**	63.1

The best results are in bold, and the second best results are underlined.†means to replace NetVLAD in the NetVLAD-baseline with the other relevant networks. ‡represents replacing PointNetVLAD in the PointNetVLAD-baseline with the other relevant methods.

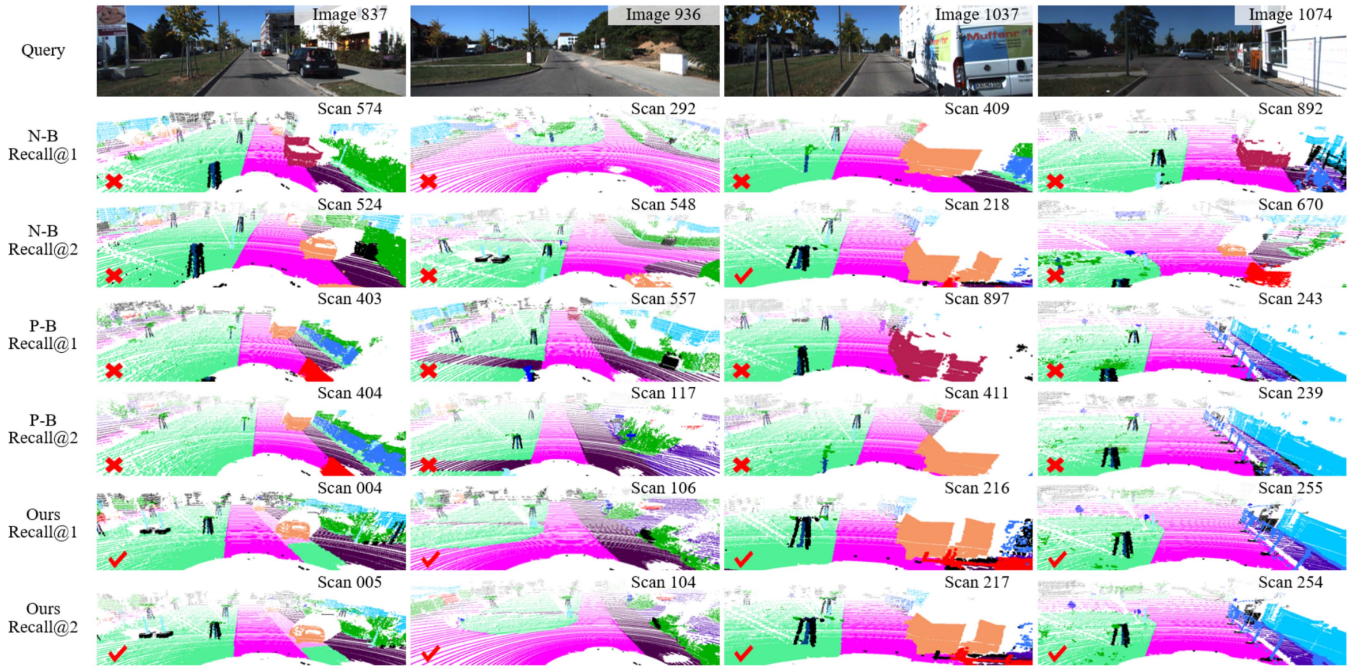


Fig. 9. Quantitative comparison between our C2L-PR method and baselines. N-B and P-B represent NetVLAD-baseline and PointNetVLAD-baseline, respectively. The first row shows query images, and rows 2-7 are the top 1 and top 2 cross-modal retrievals from different algorithms. ✔ means the correctly retrieved place while ✖ indicates incorrectness.

difficult than positive samples, that is, the larger β, the more challenges for the testing.

Table III reports the maximum F1 scores of different methods for positive/negative recognition under neg-100. We can see that our method significantly outperforms NetVLAD, PatchNetVLAD, CosPlace, $(LC)^2$, PointNetVLAD, i3dLoc, Semantic Scan Context, and I2P-Rec in terms of the mean value of maximum F1 scores, indicating that our superior performance in recognition accuracy. Our C2L-PR takes advantage of the handcrafted features and learning-based descriptors, thereby leading to better performance than the non-learning-based Semantic Scan Context. Furthermore, Fig. 10 qualitatively shows the Precision-Recall curves of our method and two baselines. The ideal curve (i.e., best performance) is that the values of precision and recall are approaching 1 at the same time. We can see that our method shows better Precision-Recall curves under both the neg-20 and neg-100 settings, which demonstrate our superiority over the baselines.

TABLE III
COMPARATIVE RESULTS IN TERMS OF F1 MAXIMUM SCORES (%)

Method	KIT-00	KIT-02	KIT-05	KIT-06	KIT-07	Mean
NetVLAD-baseline	37.8	22.9	25.2	17.9	47.0	28.7
PatchNetVLAD† [12]	37.2	19.2	31.0	19.3	21.2	25.6
CosPlace† [14]	29.6	10.2	17.3	8.7	40.7	21.3
$(LC)^2$ [56]	33.3	20.1	26.8	19.8	40.8	28.2
PointNetVLAD-baseline	50.4	45.3	26.2	15.3	15.0	30.4
i3dLoc [46]	49.3	37.7	38.9	18.0	37.0	36.2
Semantic SC‡ [26]	48.3	51.3	42.5	23.3	27.0	38.5
I2P-Rec [45]	48.5	56.3	42.4	24.0	30.2	40.3
C2L-PR (Ours)	**65.0**	**58.7**	**44.7**	**33.9**	**62.3**	**52.7**

KIT is Short for KITTI.
The best results are highlighted in bold.

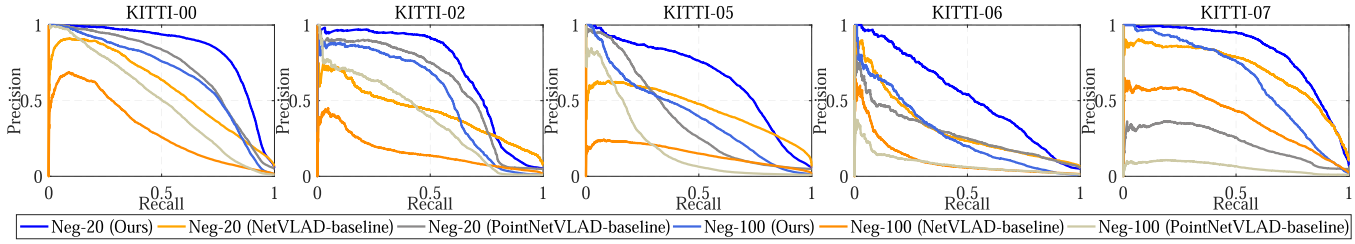


Fig. 10. Precision-Recall curves of our C2L-PR method and baselines on the KITTI dataset. Neg-20 means that the number of negatives is 20 times that of positives during testing, while neg-100 corresponds to 100 times. It shows that our PR curves are significantly better than those of NetVLAD and PointNetVLAD.

TABLE IV
GENERALIZATION EVALUATION IN TERMS OF RECALL@K ($K = 1, 5, 10$) (%) ON THE KITTI-360 DATASET

Method	KITTI360-00			KITTI360-02			KITTI360-04			KITTI360-05			KITTI360-06			KITTI360-09		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
NetVLAD-baseline	14.1	25.3	32.7	6.5	12.9	17.2	13.9	26.9	33.5	10.9	20.1	28.1	13.2	25.7	33.3	13.2	24.3	31.6
PointNetVLAD-baseline	4.1	9.2	12.4	0.5	1.1	1.6	1.8	3.1	4.9	3.9	5.8	7.2	6.1	10.0	13.0	9.3	14.1	18.5
C2L-PR (Ours)	**35.3**	**52.9**	**60.1**	**26.4**	**39.1**	**46.3**	**26.4**	**43.2**	**51.1**	**23.5**	**37.1**	**45.6**	**29.1**	**43.2**	**51.5**	**35.8**	**52.6**	**60.9**

The best results are highlighted in bold.

TABLE V
GENERALIZATION EVALUATION IN TERMS OF F1 MAXIMUM SCORES (%)

Method	K3-00	K3-02	K3-04	K3-05	K3-06	K3-09
NetVLAD-baseline	17.7	11.7	17.4	17.9	17.3	22.7
PointNetVLAD-baseline	8.8	2.6	3.7	5.9	8.5	18.7
C2L-PR (Ours)	**31.5**	**31.0**	**27.8**	**22.7**	**28.7**	**42.6**

K3-α denotes the α-th sequence of the KITTI-360 dataset.
The best result are highlighted in bold.

TABLE VI
COMPARATIVE RESULTS IN TERMS OF F1 MAXIMUM SCORES (%) ON THE OXFORD ROBOTCAR DATASET

Method	Oxford RobotCar				
	Neg-100	Neg-50	Neg-20	Neg-10	Neg-1
I2P-Rec [45]	4.6	7.7	14.9	23.9	70.3
PointNetVLAD-baseline	19.3	25.0	39.4	50.5	79.2
C2L-PR (Ours)	**43.9**	**54.6**	**68.7**	**74.0**	**89.4**

Neg-N represents that the number of Negatives is N times that of positives during testing.
The best results are highlighted in bold.

3) Generalization: To analyze the generalizability, we train our network on KITTI and then test it on KITTI-360 without retraining or fine-tuning. We use all the 6 sequences with loop closures in KITTI-360, and perform the experiments of place retrieval and positive/negative recognition. The quantitative results and comparison with baseline methods are shown in Tables IV and V. We can observe that our method has higher maximum F1 scores and Recall@K values, showing better generalization performance than the baselines. We conjecture the reason might be that our method has a handcrafted-feature module but the baseline methods do not, which helps understand unseen scenes based on prior knowledge. The prior knowledge here means that the scene can be represented using its semantic contour (the object feature) to a certain extent.

4) Robustness Across Different Datasets: To evaluate the robustness across different datasets, we test our C2L-PR with the Oxford RobotCar dataset. Since the dataset is recorded in U.K., its environment visual appearance is quite different from those in KITTI and KITTI-360, which are recorded in Germany. So, we use the KITTI dataset for pre-training and then fine-tune the network (excluding the modality alignment module) on the Oxford RobotCar dataset. The maximum F1 scores of C2L-PR are calculated and compared with I2P-Rec (which follows the *Align-to-2D* alignment strategy) and PointNetVLAD-baseline (which follows the *Align-to-3D* alignment strategy), as reported in Table VI.

We can find that despite the estimated monocular depth having scale variance between KITTI and Oxford RobotCar, our network still obtains high F1 scores. The reason for this success might be the ability of our C2L-PR to implicitly alleviate the impact of scale variance. Specifically, the scale issue of monocular depth is characterized by a geometric scaling ratio between the estimated depth and its actual depth. In our method, the extraction of place descriptors and the comparison of descriptor similarity are learnable. So, the network can implicitly mitigate the scale issue by training and encouraging the network to output a similarity score of 1 for descriptors from the same place, and 0 for those from different places. Besides, the F1 scores of our C2L-PR are significantly higher than those of I2P-Rec and PointNetVLAD-baseline, which indicates that our method can achieve cross-modal place recognition more robustly across different datasets with environment visual appearance diversity.

5) 2D-to-2D and 3D-to-3D Place Recognition: Besides our verified superior performance in 2D-to-3D cross-modal place recognition, we also test whether C2L-PR can be used as a single-modal method under 2D-to-2D and 3D-to-3D settings. Notably, since C2L-PR is a cross-modal method, some structural adjustments are necessary. Specifically, C2L-PR originally converts different modality inputs (*i.e.*, images and point clouds) into the same modality (*i.e.*, point clouds) with the proposed

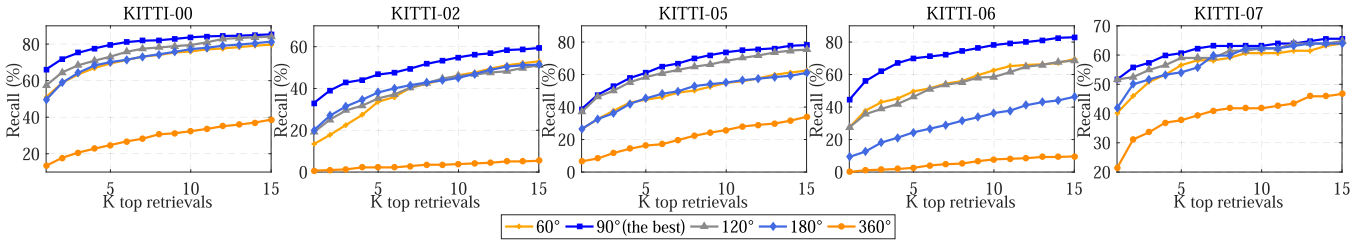


Fig. 11. Recall@K curves for different view angles of our point-cloud slicing strategy. 90° works best, while 360° without point-cloud slicing is the worst. This indicates that using slices instead of the whole point cloud has better performance, and around 90° is the suitable slice FOV on the KITTI dataset.

TABLE VII
F1 MAXIMUM SCORES (%) OF SINGLE-MODAL PLACE RECOGNITION PERFORMANCE UNDER 2D-TO-2D AND 3D-TO-3D SETTINGS

	NetVLAD	MixVPR	AnyLoc	PointNetVLAD	C2L-PR(Ours)
2D-to-2D	90.5	91.6	**92.2**	-	91.9
3D-to-3D	-	-	-	89.9	**96.0**

The best result are highlighted in bold.

TABLE VIII
ABLATION STUDY RESULTS (%) ON POINT-CLOUD SLICING STRATEGIES

Slicing Strategy	KIT-00	KIT-02	KIT-05	KIT-06	KIT-07	Mean
360°	24.8	10.5	14.4	2.9	22.1	14.9
Ours 180°	47.2	43.6	33.2	6.9	36.3	33.4
Naive 180°	47.1	38.0	26.4	6.8	31.8	30.0
Ours 120°	60.5	40.3	33.3	17.3	49.9	40.3
Naive 120°	59.1	42.6	29.9	11.1	46.6	37.9
Ours 90°	**65.0**	**58.7**	**44.7**	**33.9**	**62.3**	**52.7**
Naive 90°	64.1	48.1	42.9	33.4	59.9	49.7
Ours 60°	59.2	35.2	31.1	12.6	41.9	36.0
Naive 60°	52.0	33.8	30.1	11.6	35.2	32.5

KIT is the short for KITTI.
The best results are highlighted in bold.

TABLE IX
ABLATION STUDY RESULTS (%) ON LOSSES

Loss	KIT-00	KIT-02	KIT-05	KIT-06	KIT-07	Mean
$\mathcal{L}_{dis}$	8.1	7.3	14.4	4.7	27.2	12.3
$\mathcal{L}_{bce}$	61.6	54.0	42.9	27.4	27.0	42.6
$\mathcal{L}_{ce}$	19.8	31.1	15.3	12.5	15.7	18.9
$\mathcal{L}_{ce} + \mathcal{L}_{dis}$	64.1	57.1	39.1	23.1	39.0	44.5
$\mathcal{L}_{bce} + \mathcal{L}_{dis}$	63.3	52.1	41.3	33.8	52.7	48.6
$\mathcal{L}_{ce} + \mathcal{L}_{bce}$	**65.4**	57.1	42.6	29.7	56.1	50.2
$\mathcal{L}_{ce} + \mathcal{L}_{bce} + \mathcal{L}_{dis}$	65.0	**58.7**	**44.7**	**33.9**	**62.3**	**52.7**

Combining $\mathcal{L}_{bce}$, $\mathcal{L}_{ce}$, and $\mathcal{L}_{dis}$ leads to the best performance. KIT represents KITTI.
The best results are highlighted in bold.

Align-to-3D strategy (see Section III-B and the modality alignment module in Fig. 2). Therefore, for the 3D-to-3D setting where inputs are only point clouds, we remove the modality alignment module and keep the remaining parts in C2L-PR. For the 2D-to-2D setting where inputs are only images, we use the *Align-to-3D* strategy to transform images into point clouds, thereby converting the 2D-to-2D task into a 3D-to-3D task.

We compute the single-modal place recognition accuracy of C2L-PR on the KITTI 00 sequence, and compare it with NetVLAD [22], PointNetVLAD [28], MixVPR [57], and AnyLoc [58], as shown in Table VII. It can be seen that the maximum F1 scores of our C2L-PR are higher than those of NetVLAD, MixVPR, and PointNetVLAD, and a little lower than AnyLoc. This demonstrates that (i) the adjusted C2L-PR has the ability to achieve single-modal place recognition; (ii) it has good recognition accuracy, with comparable performance to other single-modal methods.

G. Ablation Study

1) Ablation on Point-Cloud Slicing: We respectively use the whole LiDAR point cloud, and point-cloud slices with different FOV angles to test how point-cloud slicing influences the accuracy of cross-modal place recognition. The F1 max scores of the variants on the KITTI dataset are listed in Table VIII, where the results are classified into slicing strategies: the naive slicing strategy and our proposed strategy. The recall@K curves are shown in Fig. 11 for qualitative comparisons.

We draw three conclusions from the above results. First, using the whole LiDAR point cloud for cross-modal place recognition does not perform well. In contrast, using slices of the point cloud can significantly improve F1 max scores and Recall@K values. Second, reducing the FOV of point-cloud slices can achieve better place recognition, but not the smaller the better. On the KITTI dataset, a suitable FOV is around 90°. Third, our proposed strategy brings more gains to place recognition than the naive strategy, which is verified by its higher F1 max scores. This is due to our strategy guaranteeing a point-cloud slice whose FOV roughly covers the camera FOV. However, with the naive strategy, the network may fail to recognize the places if the camera FOV lies between two point-cloud slices. In this case, none of point-cloud slices is sufficiently similar to the semantic point cloud from the image.

Furthermore, in Fig. 12, we visualize place descriptors of the input image, the whole LiDAR point cloud, and the point-cloud slice at the same place. Fig. 12 shows that compared with the whole LiDAR point cloud, the point-cloud slice has a descriptor that is closer to the image. This is because the whole point cloud has many 3-D points outside the image FOV. These points have no corresponding pixels in the image, causing the descriptor of the whole point cloud to be different from the descriptor of the image. In contrast, the point-cloud slice has fewer points outside the image FOV, resulting in a more similar descriptor.

2) Ablation on Losses: We test different losses and their combinations to investigate their contributions. Table IX shows that binary cross-entropy loss $\mathcal{L}_{bce}$ is more effective than the cross entropy loss $\mathcal{L}_{ce}$ and the distance loss $\mathcal{L}_{dis}$, because it employs stronger supervision with $0/1$ labels for training, that is, directly judging whether the given two places come from the

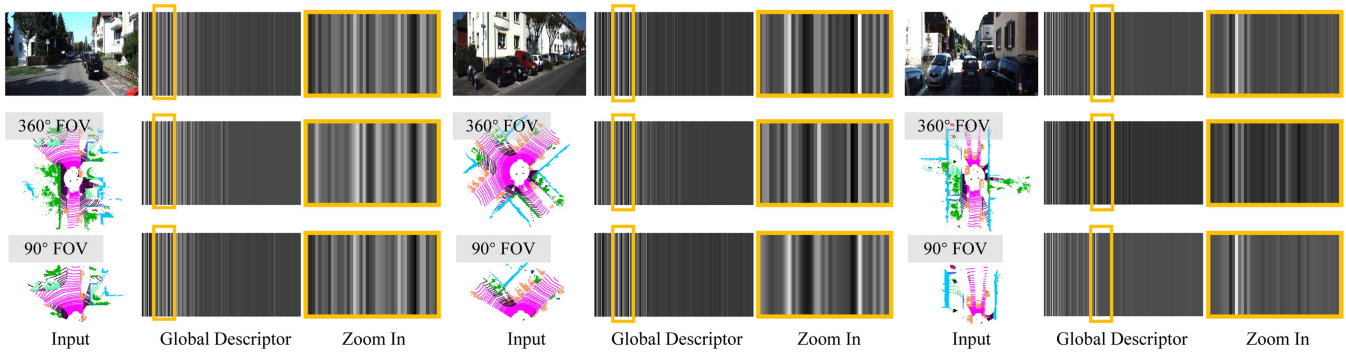


Fig. 12. Visualization of place descriptors. Given a place, we visualize its descriptor vector from a single image, a whole point cloud (360°), and a point-cloud slice (90°), respectively. In a descriptor vector, the value of each element is visualized in grayscale images. We locally zoom in the descriptor vector. It can be seen that point-cloud slices have descriptors that are closer to those from images than the corresponding whole point clouds.

TABLE X
TRAINING DATA ANALYSIS IN TERMS OF RECALL@K ($K = 1, 5, 10$) (%) ON THE KITTI DATASET

Training Data	KITTI 00			KITTI 02			KITTI 05			KITTI 06			KITTI 07		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Exact FOVs	62.1	78.2	83.0	26.7	44.8	52.4	32.6	45.5	55.5	43.3	62.2	74.4	50.0	55.7	58.2
Exact FOVs †	62.6	78.4	83.0	27.1	45.3	52.5	33.8	47.9	59.6	43.5	62.3	74.1	50.2	55.3	58.8
Exact FOVs ‡	59.6	73.2	76.1	24.3	41.9	49.5	29.5	43.3	51.0	42.2	58.6	70.1	48.7	53.4	55.3
C2L-PR (Ours)	**66.2**	**79.6**	**83.7**	**33.0**	**46.7**	**54.9**	**38.8**	**61.0**	**73.8**	**44.4**	**70.0**	**78.1**	**51.6**	**60.7**	**63.1**

†Represents data augmentation with a mask of 10% data. ‡denotes a mask of 20% data.
The best results are highlighted in bold.

same place or not. Notably, any combination of the two losses can obtain a higher mean F1 max score than that of using a single loss. Furthermore, jointly employing all three losses can achieve the best recognition performance.

Following the previous work [59], [60], [61], we construct binary place pairs and train the network using binary cross-entropy loss. Another common way to train the network is building triplet place groups and utilizing triplet loss as in the work [1], [62]. To compare these two training methods, we conduct an ablation study experiment of which the quantitative results are reported in Table XI. We can find that the binary training (using binary cross-entropy loss) and the triplet training (using triplet loss) have close average recognition accuracy. However, when combining the distance loss $\mathcal{L}_{dis}$ and the cross-entropy loss $\mathcal{L}_{ce}$, the accuracy of the binary training is improved, but the performance of the triplet training is degraded. This demonstrates that the binary cross-entropy loss is compatible with $\mathcal{L}_{dis}$ and $\mathcal{L}_{ce}$. Combining them together can achieve higher place recognition accuracy.

3) Ablation on Training Data: There are two types of FOV slice data for training the network. The first is exact-FOV slices, which are generated according to the poses of autonomous vehicles. The second is approximate-FOV slices created by employing our slicing strategy. The network is trained with these two types of slices, and their place recognition performance is compared, as reported in Table X. Furthermore, to mitigate possible overfitting of exact FOV, we implement data augmentation. Specifically, exact-FOV slices are randomly masked by setting 10% and 20% of the data to 0 respectively, corresponding to rows 4 and 5 in Table X.

TABLE XI
ABLATION ON THE TRIPLET TRAINING AND THE BINARY TRAINING (%)

Training	KIT-00	KIT-02	KIT-05	KIT-06	KIT-07	Mean
Triplet	62.0	50.2	41.1	29.5	30.9	42.7
Triplet†	54.5	53.2	39.3	19.1	27.9	38.8
Binary	61.6	54.0	42.9	27.4	27.0	42.6
Binary†	**65.0**	**58.7**	**44.7**	**33.9**	**62.3**	**52.7**

The Triplet training and the binary training respectively mean that The network is trained with the triplet loss and binary Cross-entropy loss. †denotes adding $\mathcal{L}_{dis}$ and $\mathcal{L}_{ce}$.
The best results are highlighted in bold.

It can be observed that a mask with an appropriate proportion only slightly improves network accuracy. If we continue to increase the mask proportion, the network performance would be degenerated. This indicates that the overfitting may not happen in exact-FOV slices. Furthermore, the recall values using exact-FOV slices or performing data augmentation are generally lower than those using approximate-FOV slices, demonstrating that training with approximate-FOV slices is better. The reason is that training with approximate-FOV slices ensures the consistency of FOV settings during training and testing. Particularly, in practical applications, vehicles often revisit the same place in different directions. This means that the FOVs of passing the same place twice are often approximated, but not exactly matched. If we use exact-FOV slices for training, the trained network would have the ability to identify exact-FOV slices, but may assign a low score to the target slice of which the FOV is just approximated but not perfectly matched, thus leading to place recognition mistakes. In contrast, we use

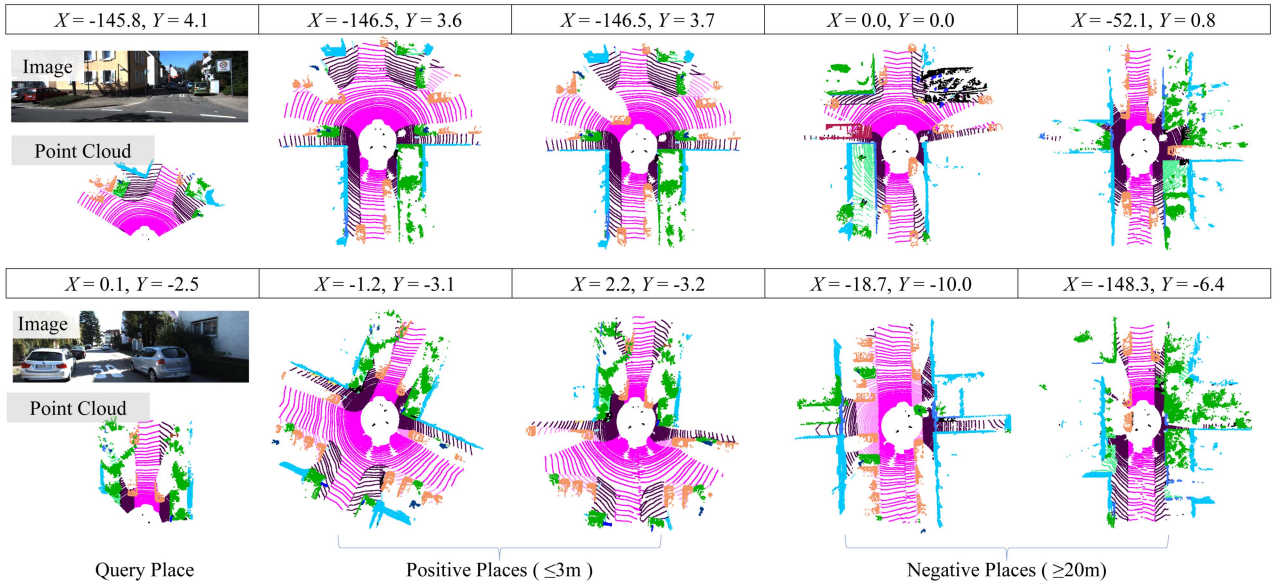


Fig. 13. Visualization of positive and negative samples. Given a place, its positive places are less than 3 m away, while the negative places are beyond 20 m. As can be seen, the negative places have more environmental differences from the query place than the positive places. The coordinates of places on the pre-built map are listed in the table above them.

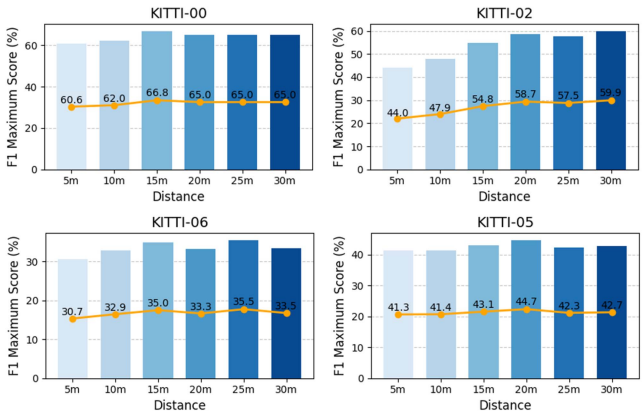


Fig. 14. Impact of distance thresholds of negative samples on cross-modal place recognition.

approximate-FOV slices for training to enable the network to recognize approximate FOVs, so that it guarantees the same FOV setting throughout the training and testing phases. That is, during training, the network learns to identify approximate-FOV slices, and during testing, the network also aims to recognize slices with similar FOVs.

4) Ablation on Threshold Parameters: We set the distance threshold for positive samples to 3 m and that for negative samples to 20 m. The visualization of positive and negative samples is shown in Fig. 13. We can see that there are more consistent environmental contents in positive places than in negative places. Our threshold choice follows [2], [26]. Notably, it can keep two places in a negative sample far away from each other to match the label value of 0. In other words, two places in a negative sample have a similarity score close to 0. This is consistent with the negative label of 0. Fig. 14 quantitatively reports the impact of different negative-sample distance thresholds on the network performance. The F1 maximum scores show that using a distance threshold of 15 m or above is better than using a threshold of 5 m or 10 m. This indicates that negative places need to be sufficiently far away. One of the recommended distance thresholds is 20 m.

Furthermore, we also test the network performance under other τ thresholds besides $\tau = 5$. Table XII shows that our method has higher recall scores than the baseline methods at $\tau = 1, 3, 10$, which illustrates the effectiveness of our network under different τ threshold choices.

5) Ablation on Two-Step Descriptor Extraction: The descriptor extraction is a two-step process. The first step utilizes a handcrafted extractor to detect the geometric and semantic contour of the scene as a handcrafted feature. The second step uses a learnable embedding sub-network to transform the handcrafted feature into a compact descriptor. We perform an ablative study to analyze the importance of combining geometric and semantic information as well as the effectiveness of two-step extraction. The qualitative experimental results are presented in Fig. 15. We can see that scene contours and visualized handcrafted features have richer content, and thus become more distinguishable when both geometry and semantics are considered. The quantitative results are reported in Table XIII, showing F1 max scores of using different descriptor variants on the KITTI dataset. Table XIII reveals that: 1) two-step descriptor extraction has higher cross-modal place recognition accuracy than one step; 2) adding semantic information leads to higher F1 max scores, indicating that semantics could be helpful to recognize places.

Another ablation experiment is conducted to evaluate the necessity of our two-step descriptor design. We project the point cloud into a BEV image, followed by a NetVLAD layer to extract the CNN descriptor. This CNN descriptor is then compared with our two-step descriptor, as illustrated in Table XIII. It can

TABLE XII
EFFECTIVENESS ANALYSIS (%) OF OUR NETWORK UNDER DIFFERENT τ THRESHOLD CHOICES

KITTI Dataset	KITTI 00			KITTI 02			KITTI 05			KITTI 06			KITTI 07		
Recall@K	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
$\tau = 1$															
NetVLAD-baseline	6.8	18.0	26.3	2.9	12.1	21.8	7.0	20.8	33.3	5.1	19.0	26.6	61.7	80.9	91.5
PointNetVLAD-baseline	23.6	55.7	69.0	6.3	20.7	26.4	8.5	28.3	43.4	8.0	21.5	30.0	27.7	40.0	51.1
C2L-PR (Ours)	**34.9**	**68.0**	**80.9**	**13.2**	**37.9**	**51.7**	**22.8**	**40.9**	**56.4**	**13.9**	**38.0**	**51.5**	**74.5**	**87.2**	**93.6**
$\tau = 3$															
NetVLAD-baseline	18.2	32.2	41.2	8.7	23.4	34.4	16.2	37.6	49.4	15.7	32.8	43.7	52.4	73.2	**78.0**
PointNetVLAD-baseline	52.7	71.2	79.4	14.4	26.1	33.1	27.8	44.9	53.0	20.1	43.7	53.7	18.3	32.9	40.2
C2L-PR (Ours)	**60.6**	**80.3**	**84.9**	**23.7**	**43.8**	**53.5**	**35.4**	**55.1**	**69.0**	**32.8**	**59.0**	**69.8**	**61.0**	**74.4**	**78.0**
$\tau = 10$															
NetVLAD-baseline	26.0	37.5	46.2	19.9	37.5	51.3	28.7	48.0	59.6	27.0	47.1	60.9	39.1	51.6	56.5
PointNetVLAD-baseline	57.9	65.4	71.4	27.9	36.4	44.3	33.6	43.7	50.6	36.9	61.3	71.5	17.4	32.3	50.9
C2L-PR (Ours)	**61.0**	**72.8**	**76.9**	**39.6**	**49.6**	**56.6**	**43.5**	**59.7**	**69.0**	**55.8**	**77.7**	**85.0**	**60.2**	**67.7**	**70.8**

The best results are highlighted in bold.

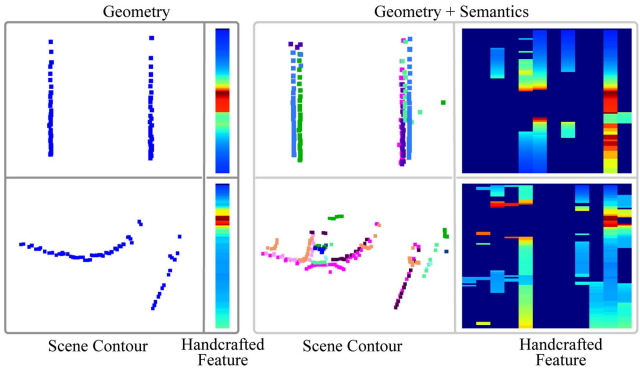


Fig. 15. Visualization of scene contours and handcrafted features. When combining geometric and semantic information, the scene contours and handcrafted features become more distinguishable.

TABLE XIII
ABLATION STUDY RESULTS (%) ON DESCRIPTOR EXTRACTION

Descriptor Extraction	CNN	One-step		Two-step	
		Geo.	Geo.+Sem.	Geo.	Geo.+Sem.
KITTI 00	49.2	35.9	46.8	36.0	**65.0**
KITTI 02	56.0	17.6	58.3	18.7	**58.7**
KITTI 05	41.7	28.1	42.0	28.7	**44.7**
KITTI 06	25.3	13.3	25.1	13.8	**33.9**
KITTI 07	36.6	12.9	23.9	16.3	**62.3**
Mean	41.8	21.6	39.2	22.7	**52.7**

Geo. and Sem. refer to geometric information and semantic information, respectively.
The best results are highlighted in bold.

be observed that the two-step descriptor achieves higher F1 max scores than the CNN descriptor, indicating higher place recognition accuracy. This demonstrates the necessity of the two-step descriptor design, which is better than the CNN descriptor design. We consider that the superiority of the two-step design comes from combining the advantages of handcrafted descriptor extraction and CNN descriptor extraction. The former explicitly perceives the geometric and semantic contour of the scene, while the latter further extracts a place descriptor in a data-driven manner.

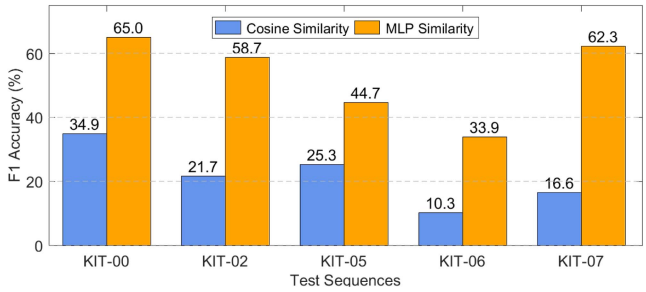


Fig. 16. Histogram of F1 scores for different similarity calculation methods. The cross-modal place recognition performance based on learnable MLP similarity is significantly higher than that based on hand-crafted cosine similarity.

TABLE XIV
PLACE RECOGNITION ACCURACY (%) USING VARIOUS MONOCULAR DEPTH ESTIMATES

	DPT	MS-DPT	Metric3D	iDisc	Ours
Accuracy	50.5	50.8	51.9	51.6	**52.7**

The best results are highlighted in bold.

6) Ablation on Similarity Measurement: Cosine similarity is a hand-crafted similarity measurement that evaluates the similarity between two descriptors by inner product. In contrast, MLP similarity is a learnable similarity measurement that employs a multilayer perceptron to infer descriptor similarity. Fig. 16 provides a quantitative comparison of cross-modal place recognition F1 scores using cosine similarity and MLP similarity, respectively. The testing sequences come from 5 suburban areas. It can be clearly observed that using MLP similarity leads to higher F1 scores than cosine similarity. This indicates that the learnable measurement can more effectively compare place similarity than the hand-crafted measurement.

7) Ablation on Depth Estimation: We evaluate the place recognition accuracy of the algorithm variants using four other monocular depth estimation methods, namely DPT [63], Metric3D [64], iDisc [65], and the improved version of DPT, MS-DPT [66]. The place recognition accuracy is evaluated on the KITTI dataset, and the mean F1 scores are reported in Table XIV.

TABLE XV
RECOGNITION ACCURACY (%) OF NETVLAD-BASELINE VARIANTS

Variant	KIT-00	KIT-02	KIT-05	KIT-06	KIT-07
Early Fusion	36.7	17.2	25.0	12.7	36.0
Defect Filling	23.5	16.9	23.9	17.8	41.3
NetVLAD-baseline	**37.8**	**22.9**	**25.2**	**17.9**	**47.0**

The early fusion combines original images and semantic maps as network input. Defect filling denotes that images projected from point clouds first undergo morphological operations to repair defects before being fed into the network.
The best results are highlighted in bold.

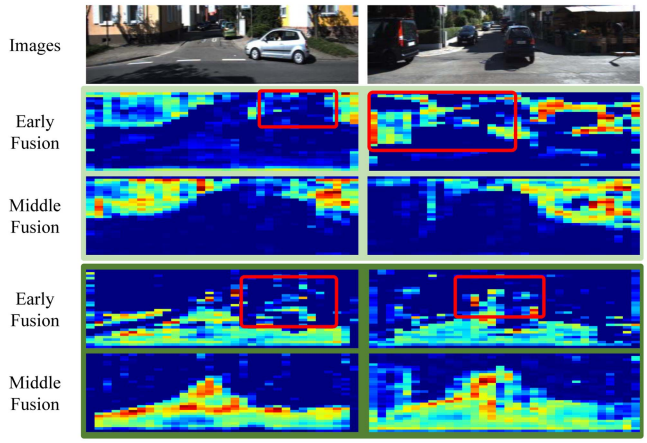


Fig. 17. Feature map visualization. Rows 2 and 3 visualize the activations of buildings. Rows 4 and 5 visualize the activations of road. The red boxes indicate errors. The early fusion directly merges the semantic map and the original image. The middle fusion integrates the semantic map with the CNN output of the original image.

We can find that different depth estimation methods yield close F1 scores, indicating that the impact of using various state-of-the-art monocular depth estimation methods is limited for place recognition performance. Among them, the depth estimation method we use [51] outperforms the others.

H. Discussions on NetVLAD-Baseline

1) Semantic Map Fusion: Different embedding positions of semantic maps influence network performance. Early fusion first merges the original image and the semantic map together, and then feeds them into the feature extraction CNN. In contrast, middle fusion concatenates the semantic map and original image features, and then performs convolutional fusion. Table XV quantitatively shows that the place recognition accuracy of the middle fusion (done by NetVLAD-baseline) is higher than that of the early fusion. We think the reason could be that the semantic map comes from the decoupling of image features, thus it is reasonable to embed the semantic map at the feature level, which can guide more accurate feature generation. Fig. 17 intuitively visualizes some of the feature maps that are fed to the NetVLAD module. Rows 2 and 3 visualize the activations of buildings. Rows 4 and 5 visualize the activations of road. It can be seen that the middle fusion obtains accurate and smooth features, while the early fusion has more errors.

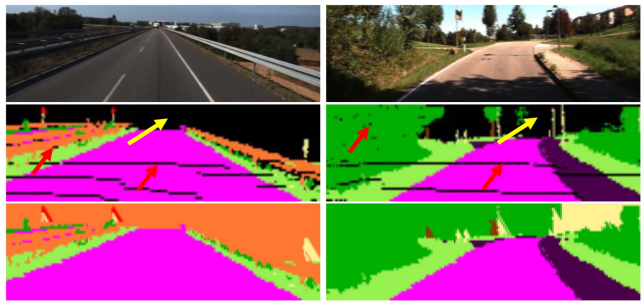


Fig. 18. Samples of morphological operation results. The second row shows the semantic maps projected from the sparse LiDAR data, suffering from defects such as gaps and holes. The third row shows the results after the morphological operations. The red and yellow arrows respectively point to defective regions that can and cannot be correctly repaired.

2) Defect Filling: Due to the sparsity of LiDAR point-cloud data, directly projecting LiDAR data onto 2-D images will result in some pixels without corresponding 3-D points, causing defects such as gaps and holes in the 2-D images. This is an inherent limitation of the *Align-to-2D* strategy. We apply morphological operations and feed the repaired images as input into the network. The quantitative results are displayed in Table XV. Besides, taking as an example the semantic maps, Fig. 18 qualitatively shows the samples of morphological operation results. We can see from Table XV that using the repaired images cannot improve the recognition accuracy of the baseline. This is because although the morphological operations are able to fill gaps and small holes correctly (pointed by the red arrows in Fig. 18), it cannot handle the holes whose contents are beyond the LiDAR sensing range (pointed by the yellow arrows in Fig. 18). In other words, we cannot accurately obtain the missing content away from the vehicle. So, defect filling through the morphological operations could not improve cross-modal place recognition performance of the baseline.

I. Robustness of Cross-Modal vs. Visual Methods

Robustness analysis of cross-modal and visual methods is performed on the Oxford RobotCar dataset. This dataset records the same driving route through Oxford multiple times, which undergoes significant appearance changes caused by varying illumination and weather, therefore being challenging. We evaluate the accuracy of cross-modal and visual place recognition under two scenarios: (1) when the vehicle first drives in the night and then passes the same place during the day, and (2) when the vehicle first travels in the rain and then revisits the same place on a sunny day. The F1 maximum scores of cross-modal and visual methods are reported in Table XVI. Table XVI shows that the cross-modal method achieves higher scores than the visual methods, indicating more accurate place recognition during environmental appearance changes. This is because LiDAR sensing in the cross-modal method is less sensitive to illumination and weather than images, allowing it to record accurate point clouds of environments under adverse conditions (such as rainfall and low light). Consequently, to a certain extent,

TABLE XVI
ROBUSTNESS COMPARISON OF VISUAL PLACE RECOGNITION VS. CROSS-MODAL PLACE RECOGNITION

	Visual		Cross-Modal
	NetVLAD	AnyLoc	C2L-PR (Ours)
Night → Day	59.5	67.2	**88.6**
Rain → Sun	81.3	83.6	**90.1**

The place recognition accuracy (%) is reported.
The best results are highlighted in bold.

TABLE XVII
ANALYSIS OF COMPUTATIONAL COST

Slices	FLOPs ↓	Memory ↓	Runtime ↓	Accuracy ↑
1	2.672G	2,672MB	0.064s	24.8%
4	5.420G	2,726MB	0.110s	47.2%
6	7.252G	2,766MB	0.142s	60.5%
8	9.081G	2,796MB	0.178s	65.0%
10	10.902G	2,828MB	0.206s	63.9%
12	12.748G	2,870MB	0.234s	59.2%

cross-modal place recognition is more robust than visual place recognition.

J. Efficiency of Camera-to-LiDAR Place Recognition

1) Computation Cost: We calculate the computation cost of the network on the KITTI 00 sequence, and evaluate its correlations with the number of point-cloud slices and place recognition accuracy. The computation cost includes memory usage, floating point operations (FLOPs), and running time. As indicated in Table XVII, more slices result in only a slight increase in memory usage. This slight increase is primarily due to the adjacent slices having a $1/2$ overlapping region, so that each 3-D point in the point cloud is processed twice, regardless of the number of slices. The slight memory increase mainly comes from caching more vectors to feed the orientation voting module. In summary, the number of slices has a limited impact on memory usage. As for FLOPs, although it changes obviously, the algorithm remains efficient, thanks to GPU hardware acceleration. It takes 0.234 seconds to compare an input place with 1,024 candidate places under a 12-slice setting. In contrast, there is a strong correlation between place recognition accuracy and the number of slices. 8 slices correspond to the highest accuracy because, at this time, the FOV of the slice approximates to that of the camera. On the one hand, if the slice FOV is too large, 3-D points beyond the camera FOV would have no matching pixels in the image. These unmatched points would become noises when comparing the similarity between the slice and the image. On the other hand, a slice with a small FOV cannot comprehensively describe the environment. Therefore, the recommended number of slices is 8, which achieves a good balance among memory usage, FLOPs, running time, and recognition accuracy.

2) Space and Time Complexity Comparison: We compare our space and time complexities with other methods. The space complexity involves FLOPs and GPU memory consumption, which indicate the scale of a method. The time complexity involves the running time. Table XVIII reports the quantitative results of these methods on the KITTI 00 sequence. It can be observed that (i) all methods have acceptable GPU consumption, allowing to allocate some GPU memory for other autonomous driving tasks. (ii) Our method has smaller FLOPs and uses less GPU memory, demonstrating lower space complexity. This is because we use lightweight 1D convolution to learn descriptor extraction instead of 2D convolution. (iii) The running time of our method outperform those of the NetVLAD-baseline, PointNetVLAD-baseline, and I2P-Rec. This is due to, on the one hand, compared with NetVLAD-baseline and I2P-Rec, our descriptor extraction uses 1D convolution instead of 2D convolution; and on the other hand, compared to PointNetVLAD-baseline, we only describe and represent the contour, rather than each point in the scene, thus reducing the time complexity.

TABLE XVIII
COMPARISON OF THE SPACE COMPLEXITY AND TIME COMPLEXITY

Slices	FLOPs ↓	Memory ↓	Runtime ↓	Accuracy ↑
1	2.672G	2,672MB	0.064s	24.8%
4	5.420G	2,726MB	0.110s	47.2%
6	7.252G	2,766MB	0.142s	60.5%
8	9.081G	2,796MB	0.178s	65.0%
10	10.902G	2,828MB	0.206s	63.9%
12	12.748G	2,870MB	0.234s	59.2%

The space complexity involves FLOPs and GPU memory consumption, which indicate the scale of a method. The time complexity involves the running time.

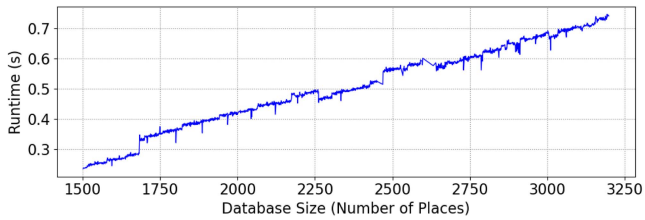


Fig. 19. Plot of algorithm runtime cost versus offline database size. We can see that the plot clearly indicates the running time cost is generally proportional to the database size.

3) Runtime and Pre-Built Map: We test the running time of our proposed method on a GeForce RTX 3060 GPU. We evaluate the relationship between the algorithm running time cost and the database size during place retrieval, as illustrated in Fig. 19. The offline database is a pre-built map composed of semantic point clouds. We can observe that the running time cost (mainly the place traversal time cost) is roughly proportional to the database size. This is because given a query place, all the candidate places in the database need to be traversed to retrieve the matching place. Consequently, a larger database entails more time. The map size we recommend could be less than 2,300 place samples (approximately a 2 KM vehicle motion trajectory), which takes a reasoning time cost of less than 0.5 s.

V. CONCLUSION AND FUTURE WORK

We proposed the C2L-PR network to achieve cross-modal place recognition using RGB images and LiDAR point clouds. A modality alignment module was proposed to reduce the modality gap by converting input data to a similar modality. Additionally, a FOV selection module was designed to alleviate the issue

of FOV inconsistency between the image and the point cloud. The experimental results demonstrate our superior performance in Recall@K values, F1 max scores, Precision-Recall curves, generalization capability, and various datasets. Furthermore, the ablation study and efficiency analysis substantiate the effectiveness of our network architecture, loss functions, training data, etc.

Our cross-modal place recognition method still has some limitations. For instance, for resource-limited vehicles, it is challenging to deploy huge maps. Besides, our place traversal time consumption is proportional to the database size. To overcome these limitations, we consider that studying map compression technology and efficient place recognition on the compressed map are promising and valuable directions for future research. Furthermore, in the future, we would like to improve our method for dynamic environments, such as complex urban cities. This would be more challenging because many moving objects may disturb our network. In addition, we will investigate viewpoint-free place recognition that can handle both horizontal and vertical FOV inconsistencies.

REFERENCES

[1] J. Ma, J. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen, "OverlapTransformer: An efficient and yaw-angle-invariant transformer network for LiDAR-based place recognition," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 6958–6965, Jul. 2022.

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

[3] W. Ma, S. Huang, and Y. Sun, "Triplet-graph: Global metric localization based on semantic triplet graph for autonomous vehicles," *IEEE Robot. Autom. Lett.*, vol. 9, no. 4, pp. 3155–3162, Apr. 2024.

[4] M. U. M. Bhutta, Y. Sun, D. Lau, and M. Liu, "Why-so-deep: Towards boosting previously trained models for visual place recognition," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1824–1831, Apr. 2022.

[5] W. Ma, H. Yin, L. Yao, Y. Sun, and Z. Su, "Evaluation of range sensing-based place recognition for long-term urban localization," *IEEE Trans. Intell. Veh.*, vol. 9, no. 5, pp. 4905–4916, 2024.

[6] S. Lowry et al., "Visual place recognition: A survey," *IEEE Trans. Robot.*, vol. 32, no. 1, pp. 1–19, Feb. 2016.

[7] S. Liu and J. Zhu, "Efficient map fusion for multiple implicit SLAM agents," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 852–865, Jan. 2024.

[8] Z. Zhou, X. Feng, S. Di, and X. Zhou, "A LiDAR mapping system for robot navigation in dynamic environments," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 852–865, Jan. 2024.

[9] X. Tian, Z. Zhu, J. Zhao, G. Tian, and C. Ye, "DL-SLOT: Tightly-coupled dynamic LiDAR SLAM and 3D object tracking based on collaborative graph optimization," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 1017–1027, Jan. 2024.

[10] W. Zhao, H. Sun, X. Zhang, and Y. Xiong, "Visual SLAM combining lines and structural regularities: Towards robust localization," *IEEE Trans. Intell. Veh.*, vol. 9, no. 6, pp. 5047–5064, Jun. 2024.

[11] T. Pivoňka and L. Přeučil, "On model-free re-ranking for visual place recognition with deep learned local features," *IEEE Trans. Intell. Veh.*, vol. 9, no. 12, pp. 7900–7911, Dec. 2024.

[12] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-NetVLAD: Multi-scale fusion of locally-global descriptors for place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14141–14152.

[13] R. Wang, Y. Shen, W. Zuo, S. Zhou, and N. Zheng, "TransVPR: Transformer-based place recognition with multi-level attention aggregation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 13648–13657.

[14] G. Berton, C. Masone, and B. Caputo, "Rethinking visual geo-localization for large-scale applications," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 4878–4888.

[15] D. Xiao, S. Li, and Z. Xuanyuan, "Semantic loop closure detection for intelligent vehicles using panoramas," *IEEE Trans. Intell. Veh.*, vol. 8, no. 10, pp. 4395–4405, Oct. 2023.

[16] H. Xu, H. Liu, S. Meng, and Y. Sun, "A novel place recognition network using visual sequences and LiDAR point clouds for autonomous vehicles," in *Proc. IEEE Int. Conf. Intell. Transp. Syst.*, 2023, pp. 2862–2867.

[17] L. Chen et al., "HVP-net: A hybrid voxel-and point-wise network for place recognition," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 395–406, Jan. 2024.

[18] K. Żywanowski, A. Banaszczyk, M. R. Nowicki, and J. Komorowski, "MinkLoc3D-SI: 3D LiDAR place recognition with sparse convolutions, spherical coordinates, and intensity," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1079–1086, Apr. 2022.

[19] L. Li et al., "RINet: Efficient 3D LiDAR-based place recognition using rotation invariant neural network," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4321–4328, Apr. 2022.

[20] Z. Cao, C. Wang, J. Li, J. Yang, and S. Wang, "LiDAR place recognition based on range image and column-shift-invariant attention," *IEEE Trans. Intell. Veh.*, early access, Mar. 18, 2024, doi: 10.1109/TIV.2024.3376002.

[21] P. Shi, Y. Xiao, W. Chen, J. Li, and Y. Zhang, "A new horizon: Employing map clustering similarity for LiDAR-based place recognition," *IEEE Trans. Intell. Veh.*, vol. 9, no. 10, pp. 5995–6005, Oct. 2024.

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

[23] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, and X. Giró-i Nieto, "Bags of local convolutional features for scalable instance search," in *Proc. Int. Conf. Multimedia Retrieval*, 2016, pp. 327–331.

[24] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "End-to-end learning of deep visual representations for image retrieval," *Int. J. Comput. Vis.*, vol. 124, no. 2, pp. 237–254, 2017.

[25] C. Toft et al., "Long-term visual localization revisited," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 4, pp. 2074–2088, Apr. 2022.

[26] L. Li et al., "SSC: Semantic scan context for large-scale place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 2092–2099.

[27] J.-U. Im, S.-W. Ki, and J.-H. Won, "Omni point: 3D LiDAR-based feature extraction method for place recognition and point registration," *IEEE Trans. Intell. Veh.*, vol. 9, no. 8, pp. 5255–5271, Aug. 2024.

[28] 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.

[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, pp. 1231–1237, 2013.

[30] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3292–3310, Mar. 2023.

[31] 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.

[32] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: A statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, no. 1, pp. 43–52, 2010.

[33] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1999, pp. 1150–1157.

[34] 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.

[35] R. Arandjelovic and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1578–1585.

[36] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3304–3311.

[37] B. Wang et al., "P2-Net: Joint description and detection of local features for pixel and point matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 16004–16013.

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

[39] 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.

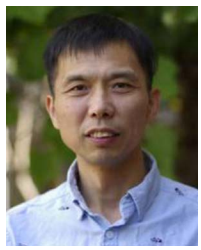
[40] A. Oertel, T. Cieslewski, and D. Scaramuzza, "Augmenting visual place recognition with structural cues," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5534–5541, Oct. 2020.

[41] H. Lai, P. Yin, and S. Scherer, "AdaFusion: Visual-LiDAR fusion with adaptive weights for place recognition," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 12038–12045, Oct. 2022.

[42] L. Bernreiter, L. Ott, J. Nieto, R. Siegwart, and C. Cadena, "Spherical multi-modal place recognition for heterogeneous sensor systems," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 1743–1750.
[43] S. Wang et al., "DistilVPR: Cross-modal knowledge distillation for visual place recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2024, pp. 10377–10385.
[44] W. Xie et al., "ModaLink: Unifying modalities for efficient image-to-pointcloud place recognition," 2024, *arXiv:2403.18762*.
[45] S. Zheng et al., "I2P-Rec: Recognizing images on large-scale point cloud maps through bird's eye view projections," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2023, pp. 1395–1400.
[46] P. Yin, X. Lingyun, J. Zhang, H. Choset, and S. Scherer, "i3dLoc: Image-to-range cross-domain localization robust to inconsistent environmental conditions," in *Proc. 17th Robot. Sci. Syst.*, 2021. [Online]. Available: https://roboticsconference.org/2021/program/papers/027/index.html
[47] N. C. Mithun, K. Sikka, H.-P. Chiu, S. Samarasekera, and R. Kumar, "RGB2LiDAR: Towards solving large-scale cross-modal visual localization," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 934–954.
[48] N. Samano, M. Zhou, and A. Calway, "You are here: Geolocation by embedding maps and images," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 502–518.
[49] D. Cattaneo, M. Vaghi, S. Fontana, A. L. Ballardini, and D. G. Sorrenti, "Global visual localization in LiDAR-maps through shared 2D-3D embedding space," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 4365–4371.
[50] X. Chen et al., "OverlapNet: Loop closing for LiDAR-based SLAM," in *Proc. Robot. Sci. Syst.*, 2020. [Online]. Available: https://roboticsconference.org/2020/program/papers/9.html
[51] W. Yuan, X. Gu, Z. Dai, S. Zhu, and P. Tan, "New CRFs: Neural window fully-connected CRFs for monocular depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 3906–3915.
[52] Y. Zhu et al., "Improving semantic segmentation via video propagation and label relaxation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8856–8865.
[53] J. Behley et al., "Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI dataset," *Int. J. Robot. Res.*, vol. 40, no. 8/9, pp. 959–967, 2021.
[54] T. Mijit, E. Firkat, X. Yuan, Y. Liang, J. Zhu, and A. Hamdulla, "LR-seg: A ground segmentation method for low-resolution LiDAR point clouds," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 347–356, Jan. 2024.
[55] Z. Jiang et al., "Efficient and unbiased safety test for autonomous driving systems," *IEEE Trans. Intell. Veh.*, vol. 8, no. 5, pp. 3336–3348, May 2023.
[56] A. J. Lee, S. Song, H. Lim, W. Lee, and H. Myung, "$(LC)^2$: LiDAR-camera loop constraints for cross-modal place recognition," *IEEE Robot. Autom. Lett.*, vol. 8, no. 6, pp. 3589–3596, Jun. 2023.
[57] A. Ali-Bey, B. Chaib-Draa, and P. Giguere, "MixVPR: Feature mixing for visual place recognition," in *Proc. IEEE Winter Appl. Comput. Vis.*, 2023, pp. 2998–3007.
[58] N. Keetha et al., "AnyLoc: Towards universal visual place recognition," *IEEE Robot. Automat. Lett.*, vol. 9, no. 2, pp. 1286–1293, Feb. 2024.
[59] X. Kong et al., "Semantic graph based place recognition for 3D point clouds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 8216–8223.
[60] S. Pramanick, E. M. Nowara, J. Gleason, C. D. Castillo, and R. Chellappa, "Where in the world is this image? Transformer-based geo-localization in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 196–215.
[61] H. Zhang, X. Chen, H. Jing, Y. Zheng, Y. Wu, and C. Jin, "ETR: An efficient transformer for re-ranking in visual place recognition," in *Proc. IEEE Winter Appl. Comput. Vis.*, 2023, pp. 5665–5674.
[62] D. Cattaneo, M. Vaghi, and A. Valada, "LCDNet: Deep loop closure detection and point cloud registration for LiDAR SLAM," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2074–2093, Aug. 2022.
[63] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 12179–12188.
[64] W. Yin et al., "Metric3D: Towards zero-shot metric 3D prediction from a single image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 9043–9053.
[65] L. Piccinelli, C. Sakaridis, and F. Yu, "iDisc: Internal discretization for monocular depth estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 21477–21487.
[66] J. Song and S. J. Lee, "Knowledge distillation of multi-scale dense prediction transformer for self-supervised depth estimation," *Sci. Rep.*, vol. 13, no. 1, 2023, Art. no. 18939.

Huaiyuan Xu received the B.S. degree, M.S. degree, and Ph.D. degree from Tianjin University, Tianjin, China, from 2010 to 2022. He is currently a Postdoctoral Research Fellow with The Hong Kong Polytechnic University, Hong Kong, China. His research interests include robotic perception, computer vision, and deep learning.

Huaping Liu (Senior Member, IEEE) received the Ph.D. degree from Tsinghua University, Beijing, China, in 2004. He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include robot perception and learning. He was the recipient of the National Science Fund for Distinguished Young Scholars. He was the Area Chair of robotics science and systems for several times. He is the Senior Editor of *International Journal of Robotics Research*.



Shoudong Huang (Senior Member, IEEE) received the bachelor's and master's degrees in mathematics, and the Ph.D. degree in automatic control from Northeastern University, Shenyang, China, in 1987, 1990, and 1998, respectively. He is currently a Professor with Robotics Institute, University of Technology Sydney, Ultimo, NSW, Australia. His research interests include nonlinear control systems and mobile robots simultaneous localization and mapping, exploration, and navigation. He was appointed as an Associate Editor for the leading journal in robotics, IEEE TRANSACTIONS ON ROBOTICS, from 2019 to 2022. He was also appointed as an editor for one of the two main large robotics conferences, IEEE/RSJ International Conference on Intelligent Robots and Systems, from 2020 to 2022. Hewas the Publication Chair for the top robotics conference, Robotics: Science and Systems 2022.

Yuxiang Sun (Member, IEEE) received the bachelor's degree from the Hefei University of Technology, Hefei, China, in 2009, the master's degree from the University of Science and Technology of China, Hefei, in 2012, and the Ph.D. degree from The Chinese University of Hong Kong, Shatin, Hong Kong, in 2017. He is currently an Assistant Professor with the Department of Mechanical Engineering, City University of Hong Kong, Hong Kong. His current research interests include Robotics and AI, autonomous systems, mobile robots, autonomous driving, robotic perception and control, autonomous navigation. He is an Associate Editor of IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE ROBOTICS AND AUTOMATION LETTERS, IEEE International Conference on Robotics and Automation, and IEEE/RSJ International Conference on Intelligent Robots and Systems.