Visual, Spatial, Geometric-Preserved Place Recognition for Cross-View and Cross-Modal Collaborative Perception

Peng Gao, Jing Liang, Yu Shen, Sanghyun Son and Ming C. Lin

Abstract— **Place recognition plays an important role in multi-robot collaborative perception, such as aerial-ground search and rescue, in order to identify the same place they have visited. Recently, approaches based on semantics showed the promising performance to address cross-view and cross-modal challenges in place recognition, which can be further categorized as graph-based and geometric-based methods. However, both methods have shortcomings, including ignoring geometric cues and affecting by large non-overlapped regions between observations. In this paper, we introduce a novel approach that integrates semantic graph matching and distance fields (DF) matching for cross-view and cross-modal place recognition. Our method uses a graph representation to encode visual-spatial cues of semantics and uses a set of *class-wise* DFs to encode geometric cues of a scene. Then, we formulate place recognition as a two-step matching problem. We first perform semantic graph matching to identify the correspondence of semantic objects. Then, we estimate the overlapped regions based on the identified correspondences and further align these regions to compute their geometric-based DF similarity. Finally, we integrate graph-based similarity and geometry-based DF similarity to match places. We evaluate our approach over two public benchmark datasets, including KITTI and AirSim. Compared with the previous methods, our approach achieves around 10% improvement in ground-ground place recognition in KITTI and 35% improvement in aerial-ground place recognition in AirSim.**

I. Introduction

Multi-robot systems have been widely studied over the past decades due to their scalability [1], parallelism [2] and reliability to failures [3], [4]. To enable efficient multi-robot collaboration, collaborative perception is an essential component to build a shared situational awareness of the surrounding environments by integrating individual perceptions. Collaborative perception has various real-world applications, such as collaborative multi-simultaneous localization and mapping (CSLAM) [5], [6], connected autonomous driving [7], [8], multi-robot delivery [9] and collaboratively search and rescue [10], [11].

Place recognition is a fundamental capability in multi-robot collaborative perception, with the goal of deciding if two robots are observing the same place. As shown in Figure 1, when unmanned ground vehicles (UGVs) and unmanned aerial vehicles (UAVs) collaboratively search an area, they need to recognize if they are observing the same place given their own observations before performing further operations, such as merging local maps, collaborative tracking and reasoning. However, place recognition in multi-robot systems is very challenging, as the multi-robot observations can be acquired by different sensors and the observations can appear quite different due to large perspective changes.

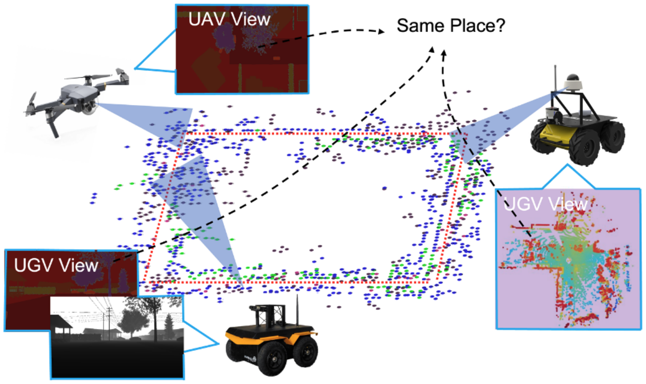


Fig. 1. An example scenario for place recognition in a robot team consisting of ground and aerial robots with different visual sensing capabilities. When three robots collaboratively search an area, before they merge their local maps (graph-based maps), they need to recognize the same place given their observations acquired by different sensors.

Given the importance of place recognition, a variety of studies have been developed. Traditional methods typically learn representations based on various sensing information, such as RGB images [12], [13] or LiDAR points [14], [15]. However, when a pair of observations have large perspective changes or are acquired from different sensors, these methods will lose effect. Recently, semantic-based approaches demonstrate promising performance to deal with perspective changes [16], [17] and sensing modality changes [18], [19]. They can be further divided into two groups, including graph-based methods based on the topology of semantic objects [16], [18], [17] and geometric-based methods based on fine geometry of semantics (e.g., shape, contour, density) [19], [20]. However, these methods still face several shortcomings. First, graph-based approaches simply abstract semantic objects as graph nodes, which ignore the important geometric cues, such as shape. Second, geometric-based approaches can not well address large non-overlapped regions due to perspective and scale changes.

In this work, we represent an observation as a semantic graph and a set of class-wise distance fields (DFs), thus encoding visual, spatial, and geometric cues of the observation. In the semantic graph, nodes denote objects with semantic attributes and edges denote the spatial distance between a pair of objects. In a DF of a specific class, each element is labeled with the distance to the closest pixel/point of that class. Given the graph representations, we perform a novel deep semantic

Peng Gao, Jing Liang, Yu Shen, Sanghyun Son, and Ming C. Lin are with the Department of Computer Science, University of Maryland, College Park, MD, USA. Email: {gaopeng, jingl, yushen, shh1295, lin}@umd.edu.

graph matching approach based on the geometric transformer to identify the correspondences of objects. The identified correspondences are further used to estimate the overlapped regions between a pair of DFs. Given the estimated overlapped region, we align two DFs by estimating their relative rotation (yaw angle) and compute the geometric similarity based on the aligned DFs. The final place recognition is performed by integrating graph-based similarity and geometric-based DF similarity.

The key contribution of this work is the introduction of *visual, spatial, and geometric preserved* place recognition for both ground-ground and aerial-ground multi-robot systems with different sensing capabilities. Specifically,

- We propose a novel representation that consistently represents an observation as a semantic graph and a set of class-wise DFs, which encodes visual, spatial, and geometric cues to improve expressiveness for place recognition. Our representations can be used in a multi-robot team with the same sensing modality or with *different modalities*.
- We propose an effective place recognition approach that integrates semantic graph matching and DF matching in a unified way. Our approach is able to not only perform ground-ground place recognition but also aerial-ground place recognition with *large perspective changes*.

The remainder of the paper is organized as follows. In Section II, we review existing methods for place recognition. In Section III, we introduce the proposed visual, spatial and geometric preserved place recognition approach. In Section IV, we present and discuss our experimental results in the scenarios of ground-ground and aerial-ground cases in KITTI and AirSim. Finally, we conclude the paper in Section V.

II. Related Work

Traditional methods for place recognition can be divided into two groups, including keypoint-based and region-based methods. The first group of methods focuses on using local features of key points in observations to perform place recognition, such as SIFT features [21], visual-spatial features [13], and super-point features of point clouds [22], [23]. The second group of methods focuses on using region-based holistic features to represent a scene, such as landmark-based graph [24], VLAD descriptor [25], HOG [26], GIST [27] and multi-modal VLAD features [28]. However, these methods can not work well in scenarios with large perspective changes, in which visual appearances of the same scene look quite different [29]. In addition, these methods can not work when a pair of observations are acquired by different sensors mounted on two robots.

To deal with the large-perspective challenge, some methods are proposed to learn view-invariant features based on Siamese network architectures [30], [31], [32], which are able to perform aerial-ground place recognition in geo-localization between ground-view observations and satellite maps. However, these methods can only work for specific scenarios with aerial-ground views, and they are not suitable to be deployed directly for ground-ground views [33].

Recently, several methods aim to study semantic representations and matching for cross-view and cross-modal place recognition. We further divide these methods into two categories, including graph-based and geometric-based methods. First, graph-based methods perform place recognition based on semantic graph representations, such as using semantic graph matching [16], [34], semantic histogram [17], bag of words [35], maximum clique [33], and semantic random walk [18]. Second, geometric-based methods focus on using fine geometric information of semantic objects for place recognition, such as shape, density, and contour of semantics. The existing methods include using truncated distance field (TDF) matching with manually defined scale factors to perform cross-view localization between RGB and LiDAR observations [19], learning view-invariant semantic scan representations [20], [36] and registering road shapes [37].

Even though semantic-based methods achieve promising performance, there are several shortcomings that have not been well addressed yet. First, graph-based representations are constructed by abstracting an object as a graph node, which ignores important geometric cues of semantics. Second, even though geometric-based approaches can well preserve geometric cues, they can not deal with large non-overlapped regions caused by large perspective and scale changes, which leads to strict limitations, such as requiring close viewpoints [20], manually selecting scale factors [19], or traveling a long distance to generate a unique road pattern [37] or collect enough number of static vehicles [33]. Our approach that integrates graph-based and geometric-based matching in a unified way can address these shortcomings for place recognition in multi-robot collaborative perception.

III. Approach

Notation. Matrices are denoted as boldface capital letters, e.g., $\mathbf{M} = \{\mathbf{M}_{i,j}\} \in \mathbb{R}^{n \times m}$. $\mathbf{M}_{i,j}$ denotes the element in the i-th row and the j-th column of $\mathbf{M}$. $\mathbf{M}_{i:j}$ denotes all the elements from the i-th column to the j-th column of $\mathbf{M}$. Vectors are denoted as boldface lowercase letters $\mathbf{v} \in \mathbb{R}^n$, and scalars are denoted as lowercase letters.

A. Problem Formulation

We consider three kinds of observations in this paper, including RGB images acquired by UAVs, RGB-D images, or LiDAR points acquired by UGVs, which are common sensor configurations in multi-robot teams. Our approach represents each observation consistently with a semantic graph representation and a set of class-wise distance field (DF) representations, as shown in Figure 2.

Specifically, given an observation with semantic labels obtained via semantic segmentation algorithms [38], [39], we first represent a place with a semantic graph $\mathcal{G} = (\mathcal{P}, \mathcal{E}, \mathcal{S})$ to encode the visual and spatial cues of the place. The node set $\mathcal{P} = \{\mathbf{p}_i, i = 1, \ldots, n\}$ represents the centroid locations of all semantic objects, with $\mathbf{p}_i$ encoding the centroid location of the i-th semantic object. We also define a semantic set $\mathcal{S} = \{\mathbf{s}_i, i = 1, \ldots, n\}$ where $\mathbf{s}_i \in \mathbb{R}^m$ is the one-hot feature vector to encode the visual semantics of objects in

$\mathcal{P}$ and m denotes the number of semantic classes. The edge set $\mathcal{E} = \{e_{i,j}, i, j = 1, 2, \ldots, n, i \neq j\}$ represents the connection between a pair of nodes, where $e_{i,j} = 1$ represents the connection between the i-th node $\mathbf{p}_i \in \mathcal{P}$ and the j-th node $\mathbf{p}_j \in \mathcal{P}$.

To encode geometric cues of observations, we further represent a place with a class-wise DF set $\mathcal{F} = \{\mathbf{F}_i\}^m, i = 1, 2, \ldots, m\}$, where $\mathbf{F}_i = \{f_{j,k}\}^{l \times w}$ denotes a DF belonging to the i-th class. Specifically, we first project all the observations to a top-down view and convert them from the Cartesian coordinate to the polar coordinate, which is computed as follows:

$$r = \sqrt{x^2 + y^2} \tag{1}$$

$$\rho = arctan(\frac{y}{x}) \tag{2}$$

where $[x, y]$ denotes the Cartesian coordinates and $[r, \rho]$ denotes the polar coordinates. Given the polarized observations, we compute DF as follows:

$$f_{j,k} = \arg\min_{j',k'} ||j - j', k - k'||^2 \text{ if } \phi(\mathcal{I}_{j',k'}) = i \tag{3}$$

where $\mathcal{I}_{j',k'}$ denotes the pixel/point at coordinate j, k in the the top-down projected views. $\phi(\mathcal{I}_{j',k'})$ denotes the semantic label of the pixel/point and $f_{j,k}$ denotes the distance from the coordinate j, k to the closest point of the i-th class.

In place recognition, observations observed by a pair of robots can be represented as $\mathcal{M} = \{\mathcal{G}, \mathcal{F}\}$ and $\mathcal{M}' = \{\mathcal{G}', \mathcal{F}'\}$ respectively. We formulate place recognition as a two-step matching problem, including the semantic graph matching with graphs $\mathcal{G}$ and $\mathcal{G}'$, as well as the DF matching with $\mathcal{F}$ and $\mathcal{F}'$. The objective is to compute a similarity score to determine whether these observations are recorded at the same place.

B. *Deep Semantic Graph Matching*

We first formulate place recognition as a graph matching problem with graphs $\mathcal{G}$ and $\mathcal{G}'$. Given a graph representation $\mathcal{G}$, we encode visual and spatial cues of each object as $\mathcal{H} = \{\mathbf{h}_i\} = \psi(\mathcal{G})$, where $\mathbf{h}_i$ is the embedding vector of the i-th object and ψ is the geometric transformer network [40]. $\mathbf{h}_i$ explicitly encodes not only the i-th object's visual semantic cue but also the spatial cues, including distance and angle information. Formally, ψ is defined as :

$$\mathbf{q}_i^l = \mathbf{W}_q^l \mathbf{h}_i^l, \quad \mathbf{k}_i^l = \mathbf{W}_k^l \mathbf{h}_i^l, \quad \mathbf{v}_i^l = \mathbf{W}_v^l \mathbf{h}_i^l \tag{4}$$

where $\mathbf{q}_i^l$, $\mathbf{k}_i^l$ and $\mathbf{v}_i^l$ denote query, key and value at the l-th layer, $\mathbf{W}_q^l, \mathbf{W}_k^l, \mathbf{W}_v^l$ denote their associating trainable weights. $\mathbf{h}_i^l$ denotes the visual semantic embedding vector of the i-th object, where $\mathbf{h}_i^0 = \mathbf{s}_i^0$. The spatial information of objects is encoded as

$$\mathbf{r}_{i,j}^l = \mathbf{d}_{i,j}^l \mathbf{W}_d^l + \max_k \{\mathbf{a}_{i,j,k}^l \mathbf{W}_a^l\} \tag{5}$$

where $\mathbf{r}_{i,j}^l$ denotes the spatial embedding of the i-th object with respect to its j-th neighbor object, $\mathbf{d}_{i,j}^l$ denotes the distance between them, $\mathbf{a}_{i,j,k}^l$ denotes the angle of vertex i in the triangle constructed by the i-th, j-th and k-th objects.

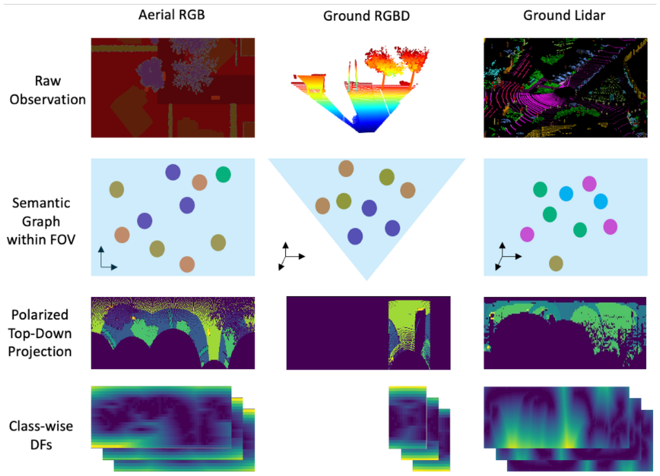


Fig. 2. Given different observations with semantic labels (first row), including an RGB image, an RGBD image pair, and a LiDAR point cloud, our approach represents them consistently with a semantic graph (second row) and a set of class-wise DFs (fourth row). DFs are generated by projecting observations to a top-down view and transforming them to polar coordinates (third row). As ground RGBD observations' field of view (FOV) is 60°, which is smaller than aerial RGB observations 360° and ground Lidar observations 360°, thus its polarized regions and DFs are much smaller than the other two's.

In the self-attention mechanism, we encode visual semantic and spatial features of objects given the self attention, which is computed as follows:

$$\alpha_{i,j}^l = \text{SoftMax}\left(\frac{(\mathbf{q}_i^l)^\top (\mathbf{k}_j^l + \mathbf{W}_r^l \mathbf{r}_{i,j})}{\sqrt{c^l}}\right) \tag{6}$$

where $\alpha_{i,j}^l$ is the self attention from object j to object i at layer l. To encode spatial relationships of objects, we add the spatial embedding $\mathbf{r}_{i,j}$ into the learning process, where W_r^l denotes its learnable parameter matrix. c^l is the dimensions of $\mathbf{r}_{i,j}$. This attention weight is obtained by comparing the query with its neighborhood keys and spatial attributes. The final attention is normalized by the SoftMax function. The object embedding vector weighted by self-attention is computed as $\mathbf{h}_i^{l+1} = \sum_{e_{i,j}=1} \alpha_{i,j}^l (\mathbf{v}_j^l)$.

In the cross-attention mechanism, we further encode visual-spatial features of potentially matched objects in the other observation. The cross attention is computed as follows:

$$\beta_{i,j}^l = \text{SoftMax}\left(\frac{(\mathbf{q}_i^l)^\top (\mathbf{k}_j^{'l})}{\sqrt{c^l}}\right) \tag{7}$$

where $\beta_{i,j}^l$ is the cross attention from the i-th object in graph $\mathcal{G}$ to the j-th object in graph $\mathcal{G}'$. The object embedding vector weighted by cross attentions is computed as $\mathbf{h}_i^{l+1} = \sum_{e_{i,j}=1} \beta_{i,j}^l (\mathbf{v}_j^{\prime l})$. The final object embedding vector is obtained via alternating self-attention and cross-attention multiple times on the visual-spatial attributes of objects.

Given the object embedding vectors, we compute the similarity between pairs of nodes in the graph $\mathcal{G}$ and the graph $\mathcal{G}'$ as follows:

$$\mathbf{S}_{i,j} = \exp(-||\mathbf{h}_i - \mathbf{h}_j^{'}||_2) \tag{8}$$

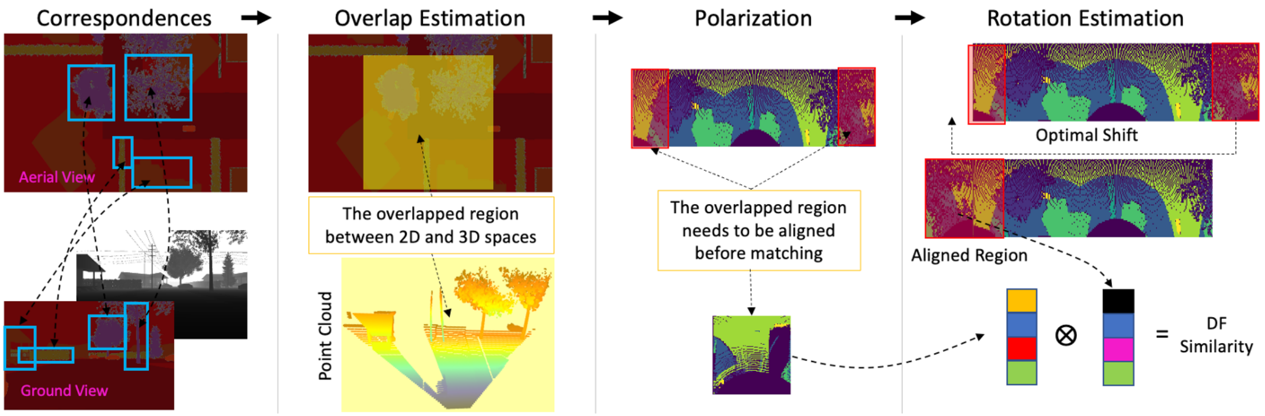


Fig. 3. Overview of our DF matching approach that consists of estimating overlapped regions based on identified correspondences, transforming RGB images or point clouds to polar coordinates, estimating rotation (yaw angle) between a pair of polarized observations, and computing the DF similarity based on the aligned observations. [The figures are best viewed in color].

where $\exp()$ denotes the exponential operator and $\mathbf{S} \in \mathbb{R}^{n\times n'}$ represents the similarity between the two graphs with n and n' objects respectively. As there are large perceptual noises and outliers existed in the observations, it is extremely hard to find one-to-one correspondences between $\mathcal{G}$ and $\mathcal{G}'$. Thus, we relax the one-to-one constraint to one-to-many constraint. Formally, we identify correspondences via selecting top M similarities in $\{\mathbf{Y}_{i,j}\}^{n\times n'} = \text{topM}(\mathbf{S})$, where $\mathbf{Y}$ denotes the correspondence matrix with $\mathbf{Y}_{i,j} = 1$ denoting the correspondence between the i-th object in $\mathcal{G}$ and the j-th object in $\mathcal{G}'$, otherwise $\mathbf{Y}_{i,j} = 0$. We use the circle loss to train our network [41], which is defined as:

$$L_{\mathcal{G}'\to\mathcal{G}} = \sum_{\mathbf{p}'_i\in\mathcal{P}'} \log[1+\sum_{\mathbf{p}_j\in\mathcal{P}^p}\exp(\gamma(D_{i,j}-\delta_p)^2) \tag{9}$$

$$\sum_{\mathbf{p}_k\in\mathcal{P}^n}\exp(\gamma(\delta_n - D_{i,k})^2)] \tag{10}$$

where $\mathcal{P} = \{\mathcal{P}^p, \mathcal{P}^n\}\mathcal{G}$ denotes the node set in graph $\mathcal{G}$. $\mathcal{P}^p$ denotes the positive nodes that have corresponding nodes in graph $\mathcal{G}'$. Similarly, $\mathcal{P}^n$ denotes the negative nodes that have no corresponding nodes in graph $\mathcal{G}'$. $\mathbf{D} = \{D_{i,j}\}^{n\times n'}$ denotes the distance matrix with $D_{i,j} = ||\mathbf{h}_i - \mathbf{h}_j||_2$ denoting the distance between a pair of feature vectors. $\delta_p = 0.2$ and $\delta_n = 1.4$ are two hyperparameters, which denote the positive and negative margins separately. $\gamma = 40$ denotes the scale factor. $L_{\mathcal{G}'\to\mathcal{G}}$ describes the loss given node set $\mathcal{P}'$ and $\mathcal{P} = \{\mathcal{P}^p, \mathcal{P}^n\}$. Similarly, we can compute the loss $L_{\mathcal{G}\to\mathcal{G}'}$ given node set $\mathcal{P}$ and $\mathcal{P}' = \{\mathcal{P}'^p, \mathcal{P}'^n\}$. The overall loss is defined as $L = (L_{\mathcal{G}\to\mathcal{G}'} + L_{\mathcal{G}'\to\mathcal{G}})/2$.

C. DF Matching

We further perform DF matching to encode geometric cues (e.g., shape, density) for place recognition. One traditional way is to directly calculate the distance between a pair of DF features [19]. Due to the large perspective changes in observations acquired by different robots, especially aerial-ground scenarios, the existence of non-overlapped regions will heavily affect the performance of place recognition.

1) Addressing Non-Overlapped Regions: To address the problem of non-overlapped regions, we estimate the overlapped regions by calculating the convex hull of all the nodes that have correspondences identified by semantic graph matching. Formally, the overlapped regions in a pair of observations are denoted as $\mathbf{A}$ and $\mathbf{A}'$ separately, which are computed as $\mathbf{A} = convexhull(\{\mathbf{p}_i\}^M)$ and $\mathbf{A}' = convexhull(\{\mathbf{p}'_j\}^M)$, where $convexhull$ denotes the function to compute convex hull given a list of points. The nodes have correspondences are denoted as $\{\mathbf{p}_i\}^M \in \mathcal{P}$ and $\{\mathbf{p}'_j\}^M \in \mathcal{P}'$ where M is the number of correspondences encoded in the constraint $\mathbf{Y}_{i,j} = 1$. We simplify the convex hull as a rectangle in this paper.

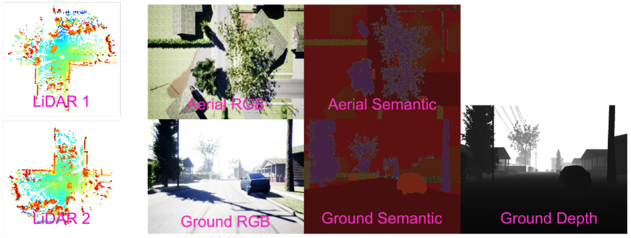
2) Addressing Rotation Changes: Given the estimated overlapped regions, we estimate the yaw angle (the motion direction of robots) between two robots' observations to align their DFs. Specifically, given a pair of polarized DFs $\mathcal{F}$ and $\mathcal{F}'$, the estimation of the yaw angle between them is defined as follows:

$$\mathbf{f}^\theta = vec([\mathbf{F}_{\theta:}, \mathbf{F}_{0:\theta}]) \tag{11}$$

where $[\mathbf{F}_{\theta:}, \mathbf{F}_{0:\theta}]$ denotes the shift operation on the polarized DFs. As $\mathbf{F}$ and $\mathbf{F}'$ are all in polar coordinates, rotating their observations in the yaw direction is equivalent to shifting their polarized DFs with the yaw angle θ. vec denotes the vectorization operation that converts a matrix to a vector by concatenating its rows. Finally, given a pair of DF features $\mathbf{f}^\theta$ and $\mathbf{f'}^\theta$, we estimate the optimal shift as follows:

$$\theta^* = \arg\max_\theta(\sum_i^m \frac{\mathbf{f}^\theta \mathbf{f'}^\theta}{|\mathbf{f}^\theta||\mathbf{f'}^\theta|}) \tag{12}$$

where θ^* denotes the optimal shift between a pair of DFs, which is computed by maximizing the overall DF similarity, where m denotes the number of semantic classes. The DF matching process is illustrated in Figure 3. Not only can our proposed DF matching deal with the inputs consisting of an RGB image and a point cloud, but it is also applicable to pairs of RGB images or pairs of point clouds.



(a) KITTI (b) AirSim

Fig. 4. Illustrations of the visual observations obtained by a pair of robots in KITTI and AirSim. KITTI covers the scenario of ground-ground robots with LiDAR sensors. AirSim covers the scenario of aerial-ground robots with RGB and RGBD cameras.

D. Place Recognition

Given the semantic graph matching and DF matching, the place recognition score is computed as follows:

$$score = \frac{\lambda}{K}\text{topK}(S_{i,j}) + \frac{(1-\lambda)}{m}\sum_{i}^{m}\frac{\mathbf{f}^{\theta *}\mathbf{f}'^{\theta *}}{|\mathbf{f}^{\theta} * ||\mathbf{f}'^{\theta *}|} \tag{13}$$

where λ is a hyperparameter that controls the weights of the graph-level similarity and the geometric-based DF similarity. The graph-level similarity is computed as the average of top K graph node similarities. The geometric DF similarity is computed as the average of m class-wise DF similarities. If one of $\mathbf{f}^{\theta *}$ and $\mathbf{f}'^{\theta *}$ is not existed, then $\frac{\mathbf{f}^{\theta *}\mathbf{f}'^{\theta *}}{|\mathbf{f}^{\theta} *||\mathbf{f}'^{\theta *}|} = 0$. If both of them are not existed, then $\frac{\mathbf{f}^{\theta *}\mathbf{f}'^{\theta *}}{|\mathbf{f}^{\theta} *||\mathbf{f}'^{\theta *}|} = 1$. Given the final similarity score that considers visual (e.g., semantics), spatial (e.g., object topology) and geometric cues (e.g., shape, density) of semantics, we perform robust place recognition by thresholding the similarity.

IV. Experiments

A. Experimental Setup

We employ two place recognition datasets, including a large-scale real-world dataset (KITTI) [42] and a simulated dataset (AirSim) [18] to benchmark our approach. Our experiments cover scenarios including ground-ground robots with LiDAR sensors and aerial-ground robots with RGB and RGBD sensors. Information on the benchmark datasets are presented in Table I.

TABLE I
Description of the Real-World KITTI and Simulated AirSim Datasets for Place Recognition.

Dataset	KITTI	AirSim
# Training Cases	91,826	10,148
# Validation Cases	40,531	1,000
# Testing Cases	91,674	2,000
Robot Type	Ground vs Ground	Aerial vs Ground
Sensors	LiDAR vs LiDAR	RGB vs RGBD
# Semantics	12	5

In the KITTI dataset, we generate over $200,000$ data instances. Each data instance contains a pair of point clouds. Following the recent method [16], we use RangeNet++ [39] to perform semantic segmentation on raw point clouds to detect these semantic objects. A total of 12 classes of objects is used to construct semantic graphs, including cars, trucks, other vehicles, sidewalks, other ground, buildings, fences, vegetation, truck, terrain, pole, and traffic signs. We use 3D positions of objects to generate the node set and the nearest neighbor search to generate the edge set. We use fences and vegetation to construct DFs. The ground-truth loop closure is obtained based on the ground-truth poses provided by the KITTI odometry dataset. We decide if two point clouds are positive or negative based on the Euclidean distance between them. If the distance is less than 10 m, then they are positive. If the distance is over 20 m, then they are negative. The ground-truth correspondences of objects are identified based on the unique ID of vehicles provided by the semantic KITTI dataset [43] and the ground-truth poses provided by the KITTI odometry dataset [42].

In the AirSim dataset, we generate over $10,000$ data instances. Each data instance contains one RGBD image pair acquired from a UGV and one RGB image acquired from a UAV. Following the recent method [18], we construct semantic graphs with 5 semantic objects, including buildings, fences, hedges, vegetation, and vehicles. In addition, we select fences, hedges, vegetation, and vehicles to construct DFs. For graph representations, we use 3D positions of objects and the nearest neighbor search to construct semantic graphs. In particular, 3D positions of objects observed by ground robots can be obtained directly from the depth images. For the aerial observations, we assume that the depth values of objects are the same (ignore the height of objects), which is the flight height of the UAV. For DF representations, we construct them using top-down projection of RGBD observations acquired from the ground view, and the RGB observations acquired from the aerial view directly. A pair of observations are decided to be positive when there are at least 10 correspondences between them. If the number of correspondences is 0, then the pair of observations is decided to be negative. The ground-truth correspondences are identified based on their ground-truth poses.

In the implementation of our network ψ, we set the number of network layers to be $L = 6$ with 3 self-attention layers and 3 cross-attention layers alternatively. Each attention layer has m, 64, and 32 as their input, hidden, and output channels separately, where m is the number of semantic classes. We set $M = 10$ for topM as defined in Eq. (8) and set $K = 5$ for topK as defined in Eq. (13). In addition, we set $\lambda = 0.8$ as defined in Eq. (8). In all the experiments, we use ADMM as the optimization method with the learning rate setting to 0.0001 and weight decay setting to 0.00005.

For comparison, we first implement two baseline methods, including (**Ours-gm**) that only uses semantic graph matching and (**Ours-df**) that only uses DF matching. We also evaluate our full approach (**Ours**). In addition, we compare our methods with three previous methods, including one traditional approach, one graph-based approach and one geometric-based approach for place recognition.

- Point cloud vector of locally aggregated descriptors (**PointVlad**) [44] that is the traditional LiDAR point-based place recognition approach. As this method can not deal with RGB images, we just evaluate it in the KITTI dataset.

- Semantic graph matching (**SG**) [16] that recognizes places based on the similarity between a pair of semantic graphs.
- Cross-view geometric-based matching (**CGM**) [19] that uses top-down projected LiDAR points acquired by ground robots and aerial-view RGB image patches cropped from reference maps to perform learning-free TDF-based matching.

As we treat place recognition as a data retrieval process, we use the following metrics to evaluate place recognition performance.

- **Precision-recall curve** is used as the evaluation metric, which is a standard metric used in the place recognition literature [16]. Precision is defined as the ratio of the retrieved correct places over all the retrieved places. Recall is defined as the ratio of the retrieved correct places over the ground-truth correct places.
- Area under the curve (**AUC**) is a single-value evaluation metric to evaluate the overall performance of place recognition methods, which takes values in [0, 1] with a greater value indicating a better performance, and a value 1 indicating the perfect performance.

B. Results on the KITTI Dataset

The KITTI dataset totally contains 11 sequences obtained by a 64-ring LiDAR, as shown in Figure 4(a). We use sequences 00, 01, 03, and 05 for training, sequences 04, 06, and 07 for validation, and sequences 02 and 08 with loop closures for testing. As sequence 02 has the largest number of instances among all the sequences with loop closures, and sequence 08 has reverse loops, they are the most challenging sequences in the evaluation of place recognition.

Quantitative results are presented in Figure 5(a) and Figure 5(b) based on the precision-recall curve. We can see that **Ours-gm** outperforms graph-based method **SG** by explicitly considering the spatial relationships of objects in the learning process. **Ours-df** outperforms geometric-based method **CGM**, which indicates the importance of addressing non-overlapping regions between pairs of observations and estimating the rotation between them. Finally, our full approach outperforms the baseline method due to its capability of integrating visual, spatial, and geometric cues, as well as addressing non-overlapped regions for place recognition. In addition, we observe that the performance of **PointVlad** and **CGM** drops quickly in sequence 08 compared with it in sequence 02. It is because they can not deal with totally opposite-direction cases in sequence 08. **SG** performs much better as semantic graph matching is invariant to perspective changes. However, it still can not address non-overlapped regions when two observations are recorded far from each other. By addressing non-overlapped regions, our approach performs the best.

We also use a single-value evaluation metric AUC to quantitatively evaluate our approach and comparisons, as shown in Table II. It is observed that our approach obtains the score of 0.9357 and 0.8767 in sequences 02 and 08 separately, which significantly outperforms the second-best method [16]. The improvements are around **15**% and **8**% separately.

TABLE II

QUANTITATIVE RESULTS OF OUR APPROACH AND COMPARISONS WITH THREE PREVIOUS METHODS BASED ON AUC SCORE. OURS ACHIEVES THE HIGHEST SCORES ON ALL 3 DATASETS.

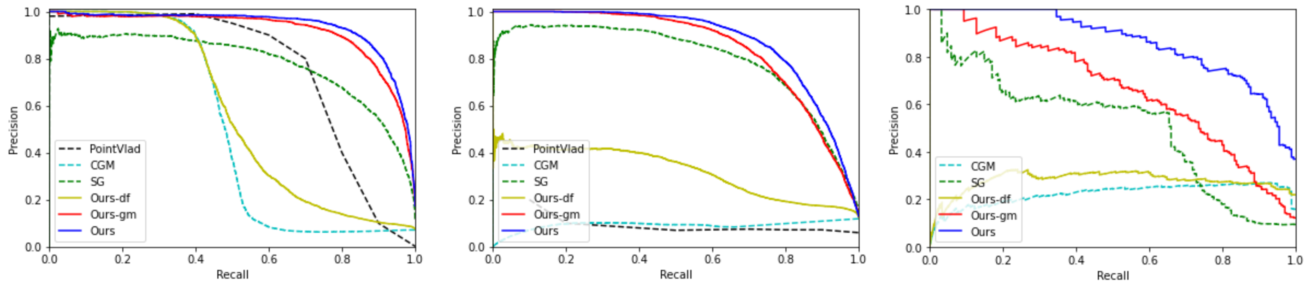
Method	KITTI-02	KITTI-08	AirSim
PointVlad [44]	0.7586	0.076	-
CGM [19]	0.5051	0.1014	0.2216
SG [16]	0.7807	0.7975	0.5054
Ours-df	0.5607	0.3221	0.2759
Ours-gm	0.9159	0.8464	0.6618
Ours-full	**0.9357**	**0.8767**	**0.8598**

C. Results on the Aerial-Ground AirSim Dataset

The AirSim dataset contains observation pairs acquired from a UAV and a UGV. The whole trajectory is 1km. In this dataset, we use RGBD images acquired by a UGV as the ground-view observations. We also use RGB images acquired by a UAV and the UAV flight height as the aerial-view observations, as shown in Figure 4(b). This dataset is very challenging due to the large perspective changes and sensing modality changes in aerial-ground observations.

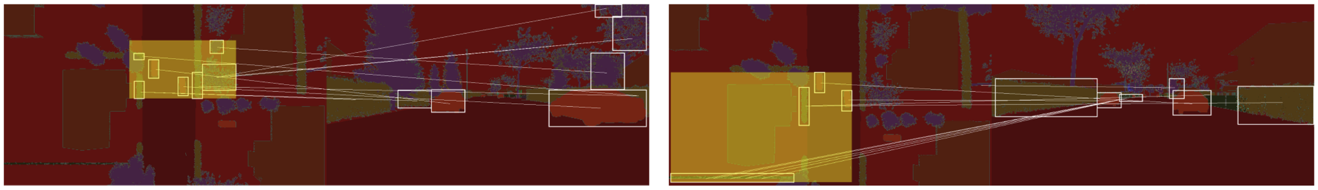
The quantitative results obtained by our method and comparisons are demonstrated in Figure 5(c). As the sensing modalities of observations are different, the traditional point feature-based approach **PointVlad** can not be used in this case. In addition, we can see that our baseline method **Ours-gm** significantly outperforms **SG** and **Ours-df** significantly outperforms **CGM**, which indicates the importance of explicitly learning visual-spatial cues in semantic graph matching and estimating overlapping regions to integrate geometric cues for place recognition. As shown in Table II. Our full approach achieves 35% improvements compared with the previous methods [16], [19] on AUC in the aerial-ground scenarios, which indicates the importance of integrating graph matching and geometric-based DF matching for place recognition, especially in the aerial-ground scenarios.

The qualitative results obtained by our full approach on the AirSim dataset are illustrated in Figure 6. The results show identified correspondences of objects, estimated overlapped regions, and matched places between aerial-ground observations. We observe that our approach can well identify the correspondences of objects in positive cases. Based on the correctly identified correspondences, our approach can significantly reduce the non-overlapped regions, as shown in Figure 6(a). For negative cases, the mismatched correspondences will generate two regions with large visual differences, thus significantly decreasing the matching score of negative cases, as shown in Figure 6(b). By correctly identifying correspondences, estimating overlapped regions, and integrating geometric cues into place recognition, our approach can perform place recognition well in both *cross-view* (aerial-ground) and *cross-modality* (RGBD-RGB) scenarios.



(a) KITTI-02 (b) KITTI-08 (c) AirSim

Fig. 5. Quantitative results on the KITTI dataset and AirSim dataset based on the precision-recall curve. Our methods are illustrated with solid curves and the others are shown in dash curves. Our method consistently achieves higher precision over the state-of-the-art methods on 3 datasets. [Best viewed in color].

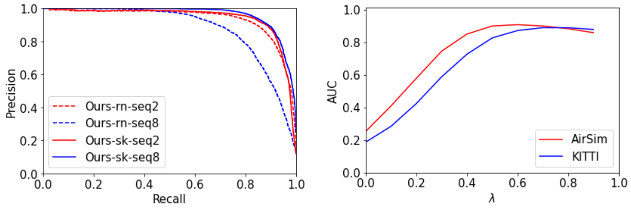


(a) Positive Case (b) Negative Case

Fig. 6. Qualitative results achieved by our approach in the AirSim dataset. The identified correspondences of objects are demonstrated in white bounding boxes on semantic segmentation images provided by both aerial (left) and ground (right) robots. The estimated overlapped regions are highlighted with yellow regions in the aerial observations. [The figures are best viewed in color].

We run our approach on a Linux machine with an i7 16-core CPU, 16G memory, and an RTX 2080 GPU. The average execution speed of our graph matching approach is 75Hz. Our full approach achieves 15Hz execution speed on KITTI and 6Hz on AirSim datasets.

D. *Discussion*



(a) Influence of SS (b) Analysis on λ

Fig. 7. Characteristics of our approach: (a) the influence caused by semantic segmentation (SS) and (b) analysis of hyperparameter λ.

1) Influence of Semantic Segmentation: The influence of semantic segmentation on our approach is shown in Figure 7(a). We compare the performance of our approach based on the semantic labels provided by RangeNet++ [39] (denoted as rn) and the ground-truth labels provided by Semantic KITTI (denoted as sk). We use sequences 02 and 08 to evaluate the influence. In sequence 02, it is observed that our approach achieves similar AUC scores based on RangeNet-provided labels or ground-truth labels, which are 0.9356 and 0.9420 separately. In sequence 08, the performance of our approach decreases from 0.9613 to 0.8767 when we change ground-truth labels with RangeNet-provided labels.

2) Hyperparameter: The analysis of hyperparameter λ as defined in Eq. (13) is shown in Figure 7(b). The hyperparameter λ is used to control the trade-off between graph matching similarity and DF similarity. We observe that our approach achieves the best performance when $\lambda \in [0.5, 0.8]$.

V. Conclusion

We propose a novel approach that integrates visual, spatial, and geometric cues to perform cross-view and cross-modal place recognition. Our approach consistently represents multi-modal observations, including RGB image, RGBD image pair, and LiDAR point cloud, as a semantic graph and a set of class-wise DFs. Given the cross-modal representations, our approach integrates semantic graph matching and DF matching in a unified way to perform place recognition, which can explicitly address non-overlapped regions between observations. Experimental results on two public benchmark datasets have shown that our approach obtains promising place recognition performance in both ground-ground and aerial-ground multi-robot systems.

Our approach has some limitations, offering possible future directions. First, the execution speed of our approach is affected by the size of the observations, especially in the generation of DFs. Downsampling techniques can be developed to reduce this size and further improve the runtime performance. Second, currently our approach assumes single-modal observations as input and can be extended to take multi-modal observations as inputs, such as UGVs with IMU and LiDAR, and UAVs with visual odometry and RGB camera, to improve robustness of place recognition.

References

[1] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

[2] G. Pini, A. Brutschy, M. Frison, A. Roli, M. Dorigo, and M. Birattari, "Task partitioning in swarms of robots: An adaptive method for strategy selection," *Swarm Intelligence*, vol. 5, pp. 283–304, 2011.

[3] P. Gao, R. Guo, H. Lu, and H. Z. Zhang, "Regularized graph matching for correspondence identification under uncertainty in collaborative perception," in *Robotics: Science and Systems*, 2021.

[4] J. D. Bjerknes and A. F. Winfield, "On fault tolerance and scalability of swarm robotic systems," in *Distributed Autonomous Robotic Systems*, 2013, pp. 431–444.

[5] P. Schmuck and M. Chli, "CCM-SLAM: Robust and efficient centralized collaborative monocular simultaneous localization and mapping for robotic teams," *Journal of Field Robotics*, vol. 36, no. 4, pp. 763–781, 2019.

[6] ——, "Multi-UAV collaborative monocular SLAM," *IEEE International Conference on Robotics and Automation*, 2017.

[7] P. Gao and H. Zhang, "Bayesian deep graph matching for correspondence identification in collaborative perception," in *Robotics: Science and Systems*, 2021.

[8] S. Wei, D. Yu, C. L. Guo, L. Dan, and W. W. Shu, "Survey of connected automated vehicle perception mode: from autonomy to interaction," *Intelligent Transportation Systems*, vol. 13, no. 3, pp. 495–505, 2018.

[9] P. Yin, S. Zhao, I. Cisneros, A. Abuduweili, G. Huang, M. Milford, C. Liu, H. Choset, and S. Scherer, "General Place Recognition Survey: Towards the real-world autonomy age," *ArXiv*, 2022.

[10] J. P. Queralta, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. Westerlund, "Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.

[11] B. Reily, J. G. Rogers, and C. Reardon, "Balancing mission and comprehensibility in multi-robot systems for disaster response," in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2021.

[12] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[13] P. Gao and H. Zhang, "Long-term loop closure detection through visual-spatial information preserving multi-order graph matching," in *AAAI Conference on Artificial Intelligence*, 2020.

[14] S. Siva, Z. Nahman, and H. Zhang, "Voxel-based representation learning for place recognition based on 3d point clouds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[15] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9, 2014, pp. 1–9.

[16] X. Kong, X. Yang, G. Zhai, X. Zhao, X. Zeng, M. Wang, Y. Liu, W. Li, and F. Wen, "Semantic graph based place recognition for 3D point clouds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020.

[17] X. Guo, J. Hu, J. Chen, F. Deng, and T. L. Lam, "Semantic histogram-based graph matching for real-time multi-robot global localization in large scale environment," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8349–8356, 2021.

[18] A. Gawel, C. Del Don, R. Siegwart, J. Nieto, and C. Cadena, "X-View: Graph-based semantic multi-view localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1687–1694, 2018.

[19] I. D. Miller, A. Cowley, R. Konkimalla, S. S. Shivakumar, T. Nguyen, T. Smith, C. J. Taylor, and V. Kumar, "Any way you look at it: Semantic crossview localization and mapping with lidar," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2397–2404, 2021.

[20] L. Li, X. Kong, X. Zhao, T. Huang, W. Li, F. Wen, H. Zhang, and Y. Liu, "SSC: Semantic scan context for large-scale place recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021.

[21] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[22] H. Thomas, C. R. Qi, J.-E. Deschaud, B. Marcotegui, F. Goulette, and L. J. Guibas, "Kpconv: Flexible and deformable convolution for point clouds," in *IEEE International Conference on Computer Vision*, 2019.

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2017.

[24] P. Gao and H. Zhang, "Long-term place recognition through worst-case graph matching to integrate landmark appearances and spatial relationships," in *2020 IEEE International Conference on Robotics and Automation*, 2020.

[25] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[26] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss, "Robust visual robot localization across seasons using network flows," in *AAAI Conference on Artificial Intelligence*, 2014.

[27] Y. Latif, G. Huang, J. J. Leonard, and J. Neira, "An online sparsity-cognizant loop-closure algorithm for visual navigation," in *Robotics: Science and Systems*, 2014.

[28] P. Yin, L. Xu, J. Zhang, and H. Choset, "Fusionvlad: A multi-view deep fusion networks for viewpoint-free 3d place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2304–2310, 2021.

[29] M. Zaffar, A. Khaliq, S. Ehsan, M. Milford, K. Alexis, and K. McDonald-Maier, "Are state-of-the-art visual place recognition techniques any good for aerial robotics?" *IEEE International Conference on Robotics and Automation workshop*, 2019.

[30] X. Zhang, W. Sultani, and S. Wshah, "Cross-view image sequence geo-localization," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023.

[31] R. Rodrigues and M. Tani, "Are these from the same place? seeing the unseen in cross-view image geo-localization," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021.

[32] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," *Advances in Neural Information Processing Systems*, 2019.

[33] J. Ankenbauer, K. Fathian, and J. P. How, "View-invariant localization using semantic objects in changing environments," *arXiv preprint arXiv:2209.14426*, 2022.

[34] J. Yu and S. Shen, "SemanticLoop: Loop closure with 3d semantic graph matching," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 568–575, 2022.

[35] Z. Yuan, K. Xu, X. Zhou, B. Deng, and Y. Ma, "SVG-Loop: Semantic–visual–geometric information-based loop closure detection," *Remote Sensing*, vol. 13, no. 17, p. 3520, 2021.

[36] L. Li, X. Kong, X. Zhao, T. Huang, W. Li, F. Wen, H. Zhang, and Y. Liu, "RINet: Efficient 3d lidar-based place recognition using rotation invariant neural network," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4321–4328, 2022.

[37] Z. Ye, C. Bao, X. Liu, H. Bao, Z. Cui, and G. Zhang, "Crossview mapping with graph-based geolocalization on city-scale street maps," in *IEEE International Conference on Robotics and Automation*, 2022.

[38] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," in *IEEE/CVF Winter Conference on Applications of Computer Vision*, 2018.

[39] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019.

[40] Z. Qin, H. Yu, C. Wang, Y. Guo, Y. Peng, and K. Xu, "Geometric transformer for fast and robust point cloud registration," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[41] Y. Sun, C. Cheng, Y. Zhang, C. Zhang, L. Zheng, Z. Wang, and Y. Wei, "Circle loss: A unified perspective of pair similarity optimization," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[43] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A dataset for semantic scene understanding of lidar sequences," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

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