IEEE Robotics and Automation Letters (RAL) paper presented at
2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

BEVGM: A Visual Place Recognition Method With Bird's Eye View Graph Matching

Haochen Niu, Peilin Liu, *Senior Member, IEEE*, Xingwu Ji, Lantao Zhang, Rendong Ying, *Member, IEEE*, and Fei Wen, *Senior Member, IEEE*

***Abstract*—Visual place recognition (VPR) is an essential tool in robotics perception and navigation. Though much progress has been made recently, the performance of VPR is far from satisfactory in challenging scenarios, such as large appearance variations, reverse viewpoints, and heterogeneous data. This work aims to fully leverage semantic and spatial information to achieve more robust and accurate VPR in these challenging scenarios. To this end, we propose a novel bird's eye view (BEV) graph matching based pipeline, which represents a scene as a unified BEV graph that can better integrate appearance, semantics, and spatial structure of the scene. Following a coarse-to-fine hierarchical paradigm, we first search the top N candidates based on global descriptors. Then, we construct BEV graphs, and formulate the similarity measurement of a query-candidate pair as a quadratic assignment problem, for which an iterative solver taking geometric consistency into account is designed. Further, we propose a Shannon entropy based adaptive fusion strategy to fuse the similarity scores from the coarse and fine matching stages. Extensive evaluation across multiple datasets demonstrates the superiority of our method in various challenging scenarios.**

***Index Terms*—Visual place recognition, semantic scene understanding, localization, SLAM.**

I. Introduction

VISUAL place recognition (VPR) is a fundamental technology in robotic tasks aimed at recognizing revisited places [1], which serves as the cornerstone for loop closure detection in simultaneous localization and mapping (SLAM) [2]. At the core of VPR lies the representation of scenes and the similarity measurement between a pair of data samples. Many methods, such as classical visual bag-of-words (vBoW) [3] and learning-based encoders [4], [5], [6], have been proposed to extract global descriptors from images and measure place similarity based on the distance of these descriptors. Some methods further consider spatial information and adopt a coarse-to-fine hierarchical paradigm, utilizing local feature (e.g. point and region matching [7], [8], [9], [10], [11]) to rerank candidates obtained based on global descriptors. However, when the appearance and viewpoint changes, local feature may change dramatically, leading to degradation or even failure of these methods.

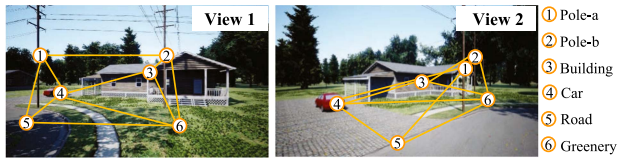


Fig. 1. Illustration that graphs of the same scene in two front views are inconsistent due to the ambiguities of the projection (only partial nodes and edges are drawn for clarity).

In contrast, instance-level semantic information and scene spatial layout are more stable when facing appearance and viewpoints changes. Recently, graph representation has garnered interest as a structure that can efficiently integrate these information. Typically, graph-based methods [12], [13], [14], [15] denote objects as nodes and utilize the 2D/3D distance between objects as edges to represent the scene, which have demonstrated superiority in indoor scenes. However, they overlook the background information and appearance of objects in the scene, resulting in suboptimal performance. Though some works take background information into account by clustering segmented background regions, they still represent scenes in the front view [16], [17], [18], in which the graph structure lacks consistency across different viewpoints, as illustrated in Fig. 1. Moreover, graph matching for similarity measurement is typically implemented relying solely on semantics and adjacency relations, neglecting appearance and/or geometric information.

In summary, existing methods suffer from the following limitations. 1) Representing a 3D scene as a 2D graph in the front view would lead to inconsistent spatial layout across different viewpoints. 2) Existing methods only use the adjacency topology information of semantics in matching but ignore the spatial geometric information. 3) Appearance information is either ignored, or processed separately rather than integrated in the graph matching stage, which limits the performance.

To address these limitations, we present BEVGM, a novel method for VPR based on bird's eye view (BEV) scene graph representation and matching, which benefits from the consistent spatial layout representation of BEV. BEV has demonstrated efficacy in autonomous driving tasks such as 3D prediction and planning recently [19], [20]. Specifically, for better information integration, we construct a unified BEV graph with semantics, appearance, and geometry information by processing objects as nodes and encoding background into edges. Besides, to measure similarity while considering all the information symbiotically,

Manuscript received 15 December 2023; accepted 29 March 2024. Date of publication 16 April 2024; date of current version 22 April 2024. This letter was recommended for publication by Associate Editor M. Popovic and Editor J. Civera upon evaluation of the reviewers' comments. This work was supported by the National Natural Science Foundation of China (No. 62276166) and the STI 2030-Major Projects (No. 2022ZD0208700). *(Corresponding author: Rendong Ying.)*

The authors are with Brain-Inspired Application Technology Center (BATC), Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: haochen_niu@sjtu.edu.cn; liupeilin@sjtu.edu.cn; jixingwu@sjtu.edu.cn; swagger@sjtu.edu.cn; rdying@sjtu.edu.cn; wenfei@sjtu.edu.cn).

Code is available at https://github.com/Haochen-Niu/BEVGM.

Digital Object Identifier 10.1109/LRA.2024.3389610

we transform a query-candidate pair of BEV graphs into an affinity matrix, and then formulate the similarity measurement as a quadratic assignment problem (QAP). Further, we design an iterative solver to solve the QAP based on reweighted random walks (RRW) [21] and neural constraint module LinSAT-Net [22], and fuse the coarse and fine similarity scores adaptively to get the final result.

The main contributions are as follows.

- A novel BEV graph matching based coarse-to-fine hierarchical pipeline for VPR, which uses a unified BEV scene graph representation to integrate high-level semantics, appearance, and 3D spatial information.
- A solver that incorporates a neural constraint model into the QAP solver and iteratively performs graph matching, which takes geometric consistency into account to achieve more robust matching.
- A Shannon entropy based adaptive fusion strategy fuses the similarity scores from the coarse and fine stages.
- Extensive experimental evaluation showcases the effectiveness of our method in various challenging scenarios.

II. Related Work

A. Descriptor-Based Methods

Descriptor-based VPR methods focus on effective extraction of local and global descriptors. While traditional methods typically aggregate handcrafted local descriptors into global descriptors [3], [23], trainable methods utilize end-to-end learning to directly extract global descriptors and become dominant recently [4], [5]. Besides, [24] takes semantic segmentation together with RGB as input to explicitly supplement semantic information during the model training.

As end-to-end global descriptor extraction methods suffer from losing the structure of scenes, reranking methods using point/region-level feature matching, in addition to global feature retrieval, have been proposed [7], [8], [9], [10]. LoST [7] clusters feature maps according to semantic labels and then stitches together the various categories of residual descriptors to form a local semantic tensor. And then they use the semantic information again for point-level spatial layout verification. TransVPR [9] incorporates a transformer module into the model for scene embedding and uses the output tokens of the transformer layers to realize patch-level descriptor matching. Furthermore, R2Former employs a pure transformer and a reranking strategy based on attention maps to achieve better results [10]. Though much progress has been made, these methods cannot achieve satisfactory performance in challenging scenarios yet.

B. Graph-Based Methods

An effective way to enhance the robustness and accuracy of VPR is to exploit structured high-level semantic information, e.g., using object-level semantic cues for scene representation. This can be achieved by object-oriented graph representation of the scenes. Typically, graph-based VPR methods represent objects as nodes. As for edges, [25], [26] use the co-visibility relationship between objects to construct graphs and match by cost-scaling push-relabel algorithm and spectral method, respectively. While some methods use 2D/3D distance to construct edges [12], [13], [14], [15], [16]. In the graph matching stage, X-View [16] uses RWD for scene graph embedding from semantic segmentation. Furthermore, [12] uses the directed neighbor walk descriptor as a complement to RWD to alleviate the problem of ambiguous instances in symmetric scenarios. [13] transfers RWD to semantic histogram to improve matching efficiency. Besides, [14] uses the edit distance between graphs to measure similarity. However, these methods typically only consider the adjacency relations between objects and ignore appearance and background information in the scene. Additionally, topological adjacency in the imaging plane is not consistent across viewpoint changes, which limits their performance in wide-angle scenarios.

To address these limitations, we consider this problem from BEV and propose a novel graph-based method for VPR, which enables better integration of appearance, semantics and structural information of the scene, and hence can achieve robust performance even in the case of extreme reverse views.

III. Method

A. System Overview

Our method implements VPR in a hierarchical paradigm, as shown in Fig. 2. Firstly, MR-NV [5] is used for coarse screening to obtain the top N candidates of a query with coarse scores. Then, semantic segmentation and depth information of the query are extracted. Afterwards, the foreground and background information are encoded into nodes and edges, respectively, to create a BEV topological graph representation of the scene (yellow dashed box). Subsequently, an affinity matrix from the query-candidate graph pair is constructed, with which we reformulate the similarity measurement into a QAP and solve it by an iterative solver to obtain the correspondence between nodes (blue dashed box). Finally, the hierarchical similarity scores are fused adaptively based on Shannon entropy and the candidates are reranked accordingly to get the final matching result (purple dashed box).

B. BEV Graph Construction

To better preserve the semantic and structural information of the scene, we partition the pixel-level semantic labels into foreground elements set Sem^f such as signs and buildings, and background elements set Sem^b such as road and greenery. Subsequently, we encode the foreground and background information into nodes and edges, respectively.

1) Node Representation: Firstly, we extract the foreground categories and apply morphological operations to remove small connected regions caused by noise. Secondly, clusters are obtained by applying DBSCAN [27] based on the depth and pixel position information of foreground segments. Then, we represent each cluster as a node $n_i = (x_i, y_i, c_i, \boldsymbol{f}_i)$, where x_i and y_i are the coordinates of the cluster center in the BEV, obtained through the projection of pixel coordinates from the image plane based on depth. c_i is the semantic category. $\boldsymbol{f}_i = \mathcal{F}(cluster_i)$, that the i-th cluster region of image $cluster_i$ is fed into the appearance embedding module MR-NV $\mathcal{F}(\cdot)$ to get appearance feature.

2) Edge Representation: We establish undirected edges between each pair of nodes. The distance l_{ij} between nodes n_i and n_j is regarded as the length property of the edge. Unlike existing methods that discard the background semantics or treat it as nodes as well, we project each pixel in the background into the BEV plane using inverse perspective mapping (IPM) [28] to aggregate more information in the graph. We set up a virtual BEV

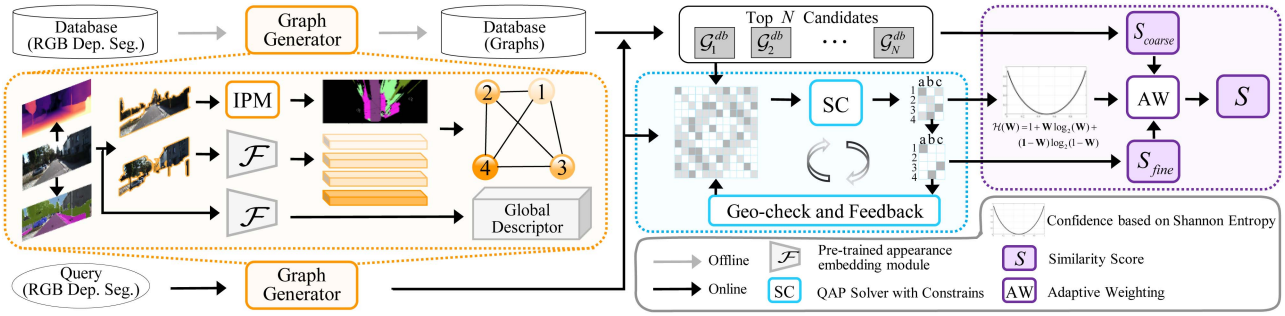


Fig. 2. Overview of the proposed BEVGM system (only partial nodes and edges are drawn for clarity).

camera with intrinsic matrix denoted by $\boldsymbol{K}_b$, which depends on the desired field of view and resolution. Let $\boldsymbol{P}_f = [x_f, y_f, z_f]^\top$ and $\boldsymbol{P}_b = [x_b, y_b, z_b]^\top$ denote the coordinates of a point in the real front view camera and virtual BEV camera coordinate systems, it follows that

$$\boldsymbol{P}_f = \boldsymbol{R}\boldsymbol{P}_b + \boldsymbol{t}, \tag{1}$$

where $\boldsymbol{R}$ denotes the rotation and $\boldsymbol{t}$ denotes the translation between two coordinate systems.

Following the pinhole camera model, the homogeneous coordinates of a point in the two coordinate systems, denoted as $[u_f, v_f, 1]^\top$ and $[u_b, v_b, 1]^\top$, can be expressed as:

$$\left[u_f, v_f, 1\right]^\top = \boldsymbol{K}_f \boldsymbol{P}_f / z_f, \quad \left[u_b, v_b, 1\right]^\top = \boldsymbol{K}_b \boldsymbol{P}_b / z_b. \tag{2}$$

Considering the flat ground plane assumption, which holds approximately in urban road scenarios, a background point should satisfy $\boldsymbol{n}\boldsymbol{P}_b = h$, where h is the height of the BEV camera principal point and $\boldsymbol{n}$ is the ground normal vector. Combining (2) and substituting into (1) yields:

$$\left[u_b, v_b, 1\right]^\top = \frac{z_f}{z_b} \boldsymbol{K}_b \left(\boldsymbol{R} + \boldsymbol{t}\frac{1}{h}\boldsymbol{n}^\top\right)^{-1} \boldsymbol{K}_f^{-1} \left[u_f, v_f, 1\right]^\top. \tag{3}$$

After the projection, the background semantic segmentation in the BEV is obtained. Then, the semantic property can be calculated based on the distribution of the background area between nodes n_i and n_j as:

$$s_{ij} = [s_{ij,1}, s_{ij,2}, \ldots, s_{ij,k}, \cdots], s_{ij,k} = l_{ij,k}/l_{ij}, \tag{4}$$

where $l_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ is the total length of e_{ij} and $l_{ij,k}$ represents the length of category k in e_{ij}, $k \in Sem^b$. Finally, the edge information is defined as $e_{ij} = (l_{ij}, s_{ij})$.

Following these steps, we construct a BEV graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ for the input image with $\mathcal{N} = (n_1, n_2, n_3, \ldots)$, $\mathcal{E} = (e_{12}, e_{13}, \ldots e_{23}, e_{24}, \ldots)$.

C. Similarity Measurement

At the offline stage, BEV graphs are constructed for all samples in the database. And at the online stage, we construct a BEV graph of the query in the same way and then proceed with a coarse-to-fine matching procedure. Firstly, the top N candidates are selected based on coarse scores obtained from global descriptors. Next, we perform BEV graph matching on each query-candidate pair to get fine scores. Finally, the final result is obtained through the adaptive fusion of similarity scores from both the coarse and fine matching stages.

Specifically, let $\mathcal{G}^q = (\mathcal{N}^q, \mathcal{E}^q)$ denote the BEV graph of a query with a nodes and $\mathcal{G}^{db} = (\mathcal{N}^{db}, \mathcal{E}^{db})$ denote the BEV graph of a candidate with b nodes. The similarity is calculated as follows.

1) Similarity Definition: Firstly, we define the similarity between nodes $n_i^q \in \mathcal{N}^q$ and $n_j^{db} \in \mathcal{N}^{db}$ as:

$$Sim_n(n_i^q, n_j^{db}) = Sim_c(c_i, c_j) \cdot Sim_a(\boldsymbol{f}_i, \boldsymbol{f}_j), \tag{5}$$

where $Sim_c(\cdot)$ computes the category similarity as:

$$Sim_c(c_i, c_j) = \begin{cases} 1, & \text{if } c_i = c_j, \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

whilst $Sim_a(\cdot)$ computes the appearance similarity in terms of the cosine distance. Meanwhile, the similarity between edges $e_{ii'} \in \mathcal{E}^q$ and $e_{jj'} \in \mathcal{E}^{db}$ is defined as:

$$\begin{aligned} Sim_e(e_{ii'}, e_{jj'}) &= Sim_e((l_{ii'}, s_{ii'}), (l_{jj'}, s_{jj'})) \\ &= \lambda Sim_l(l_{ii'}, l_{jj'}) + (1-\lambda) Sim_s(s_{ii'}, s_{jj'}), \end{aligned} \tag{7}$$

where the length similarity $Sim_l(\cdot)$ is measured by the ratio of the length difference, with the function $exp(\cdot)$ guaranteeing that the output range is restricted into [0,1] and that the similarity is defined to be proportional to the computed score. And the semantic similarity $Sim_s(\cdot)$ is measured by the difference in semantic proportions, as follows:

$$Sim_l(l_{ii'}, l_{jj'}) = \exp\left(-|l_{ii'} - l_{jj'}|/l_{ii'}\right), \tag{8}$$

$$Sim_s(s_{ii'}, s_{jj'}) = -||s_{ii'} - s_{jj'}||_2. \tag{9}$$

Next, we define $\boldsymbol{W} \in \{0,1\}^{a \times b}$ as the node correspondence matrix of a graph pair. For nodes $n_i^q \in \mathcal{N}^q$ and $n_j^{db} \in \mathcal{N}^{db}$, $\boldsymbol{W}_{i,j} = 1$ if n_i^q matches with n_j^{db} and $\boldsymbol{W}_{i,j} = 0$ otherwise.

Based on the above definitions, the similarity between two BEV graphs can be expressed as:

$$\begin{aligned} S_{fine} = &\sum_{i=1}^{a}\sum_{j=1}^{b} Sim_n\left(n_i^q, n_j^{db}\right) \cdot \boldsymbol{W}_{i,j} \\ &+ \sum_{i=1}^{a}\sum_{\substack{i'=1 \\ i' \neq i}}^{a}\sum_{j=1}^{b}\sum_{\substack{j'=1 \\ j' \neq j}}^{b} Sim_e(e_{ii'}, e_{jj'}) \cdot \boldsymbol{W}_{i,j}\boldsymbol{W}_{i',j'} \end{aligned} \tag{10}$$

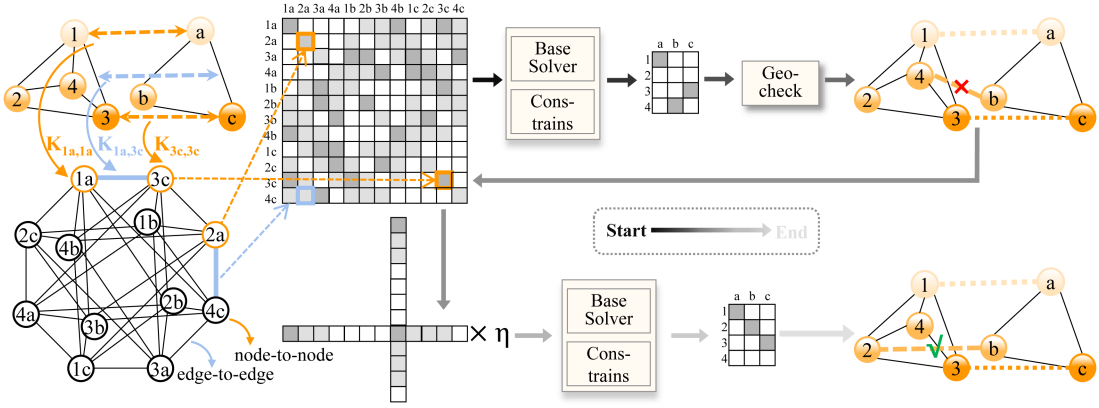


Fig. 3. Illustration of the iterative graph matching solver based on geometric consistency and negative feedback.

Then the node and edge similarities can be transformed into an affinity matrix $\boldsymbol{K} \in \mathbb{R}^{ab\times ab}$, as illustrated in Fig. 3, where the diagonal elements represent $Sim_n(\cdot)$, while the off-diagonal elements represent $Sim_e(\cdot)$. Hence, (10) can be rewritten as $S_{fine} = vec(\boldsymbol{W}^\top)\boldsymbol{K}vec(\boldsymbol{W})$, where $vec(\cdot)$ denotes the vectorization operation. The graph match problem can be expressed in a QAP form as follows:

$$\max_{\boldsymbol{W}} vec(\boldsymbol{W}^\top)\boldsymbol{K}vec(\boldsymbol{W})$$
$$\text{s.t. } \boldsymbol{W}\in\{0,1\}^{a\times b}, \boldsymbol{W}\mathbf{1}_{b\times 1}\le\mathbf{1}_{a\times 1}, \boldsymbol{W}^\top\mathbf{1}_{a\times 1}\le\mathbf{1}_{b\times 1}, \quad (11)$$

The constraints only allow one-to-one correspondence between nodes in the two graphs.

2) Graph Matching: We employ an iterative solver to solve the graph matching problem (11). Building upon the base solver RRW and the linear constraint module LinSATNet, we implement an iterative solver by introducing negative feedback based on geometric consistency check, as shown in Fig. 3.

RRW treats the affinity matrix as the adjacency matrix of a graph, namely, association graph. Hence the problem is reformulated as node selection on the association graph using Markov random walk statistics. We use LinSATNet to enforce the constraints in (10), which encodes positive linear satisfiability into a learnable neural network. The bidirectional matching constraints can be reformulated into the following positive linear constraints, which serve as input to LinSATNet:

$$\boldsymbol{A}\cdot vec(\boldsymbol{W})\le\boldsymbol{p}, \quad (12)$$

where $\boldsymbol{p} = \begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}^\top_{1\times(a+b)}$ and

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A_1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{A_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{A_1} \\ \hdashline \boldsymbol{A_2} & \boldsymbol{A_2} & \boldsymbol{A_2} & \cdots & \boldsymbol{A_2} \end{bmatrix}_{(a+b)\times(a\cdot b)} \quad (13)$$

with $\boldsymbol{A_1} = \begin{bmatrix}1 & 1 & \cdots & 1\end{bmatrix}_{1\times a}$ and

$$\boldsymbol{A_2} = \begin{cases} \begin{bmatrix}\boldsymbol{I}_b & \mathbf{0}_{b\times(a-b)}\end{bmatrix}, & a\geqslant b, \\ \begin{bmatrix}\boldsymbol{I}_a \\ \mathbf{0}_{(b-a)\times a}\end{bmatrix}, & a<b. \end{cases}$$

The affinity matrix $\boldsymbol{K}$ actually incorporates only first-order node similarity and second-order edge similarity of the graph pair. An alternative way is to extend the affinity matrix to the third order by including angle information. Yet this would significantly increase the complexity to the optimization problem.

As a result, we separate the verification of spatial layout and employ an iterative algorithm to solve the problem (11). Because of the projection onto BEV, the geometric structural relationship between the two graphs is simplified to three degrees of freedom (DOF). Based on the node matching result, we aim to minimize the reprojection error to solve for the relative pose with RANSAC. For example, let (n_i^q, n_j^{db}) denotes a mismatched node-pair, which would have high reprojection error. Then, we propagate negative feedback from the initial matching results in the form of reducing the weights of corresponding elements in the affinity matrix. The updating formula is as follows:

$$\begin{aligned}\boldsymbol{K}_{iter+1} &\leftarrow \eta\cdot(\boldsymbol{K}_{iter})_{col:id,row:id}, \\ id &= j\cdot a + i,\end{aligned} \quad (14)$$

where $(\cdot)_{col:m,row:n}$ represents that the calculation involves elements corresponding to the m_{th} row and n_{th} column of this matrix. η is a negative feedback coefficient. In this way, we can repeat the above steps to iteratively solve and refine the matching matrix as shown in the right part of Fig. 3.

3) Adaptive Reranking: After obtaining the node correspondence matrix $\boldsymbol{W}$, we calculate the fine score as:

$$S_{fine} = vec(\hat{\boldsymbol{W}}^\top)\boldsymbol{K}vec(\hat{\boldsymbol{W}}), \quad (15)$$

where $\hat{\boldsymbol{W}}$ is obtained by the Hungarian algorithm [29] to convert the soft-matching matrix $\boldsymbol{W}$ into a hard-matching matrix. Then we design an adaptive fusion strategy to fuse the two-stage scores as:

$$S = (1+\kappa\mathcal{H}(\boldsymbol{W})S_{fine})\cdot(1+(1-\kappa\mathcal{H}(\boldsymbol{W}))S_{coarse}, \quad (16)$$

where S_{coarse} denotes the similarity score in the coarse matching stage, κ is a positive coefficient, and $\mathcal{H}(\cdot)$ signifies the confidence in S_{fine}, which is computed based on the Shannon entropy of $\boldsymbol{W}$:

$$\mathcal{H}(\boldsymbol{W}) = 1 + \boldsymbol{W}\log_2(\boldsymbol{W}) + (\mathbf{1}-\boldsymbol{W})\log_2(1-\boldsymbol{W}). \quad (17)$$

By reranking the candidates based on S, we obtain the best match with the highest S.

TABLE I
DATASET CHARACTERISTICS

Dataset	Reverse viewpoints	Appearance variations	Heterogeneous data
SYNTHIA (sim)	✓	✓	✗
AirSim (sim)	✓	✓	✓
KITTI (real)	✓	✗	✗
Oxford Robotcar (real)	✓	✗	✗

IV. EXPERIMENTAL RESULTS

We conduct a series of experiments to evaluate the proposed method on four datasets, the SYNTHIA dataset [30], the custom AirSim dataset [31] collected by us, the KITTI dataset [32] and the Oxford Robotcat dataset [33], with characteristics shown in Table I.

The performance is evaluated primarily in terms of the precision-recall curve (PRC). Meanwhile, for a more comprehensive analysis, we also employ the max F1-score (mF1), the area under the PRC (AUC) and recall at 100% precision (R@100P) as quantitative indicators [1], [34], [35]. As mentioned in [7], there exists visual offset between reverse viewpoints. Therefore, we define the intersection over union (IoU) of the camera's field of view as the criterion for determining the ground truth. In our experiments, a pair with an IoU greater than 0.3 is considered a correct match. The impact of threshold is further analyzed in Section IV-E. Note that our method focuses on place recognition and dose not perform 6-DOF pose estimation.

The compared methods include global descriptors-based, MR-NV [5], CoHOG [23] and NetVLAD [4], as well as reranking-based, R2Former [10], TransVPR [9], and LoST [7]. We denote LoST (resp. R2Former) the method without reranking and LoSTX (resp. R2FormerX) the full method, respectively. We utilize OneFormer [36] as the semantic segmentation module and Lite-mono [37] as the depth estimation module. For the learning-based baselines, as well as the learning-based modules employed in our method, we directly use their publicly released models without fine-tuning.

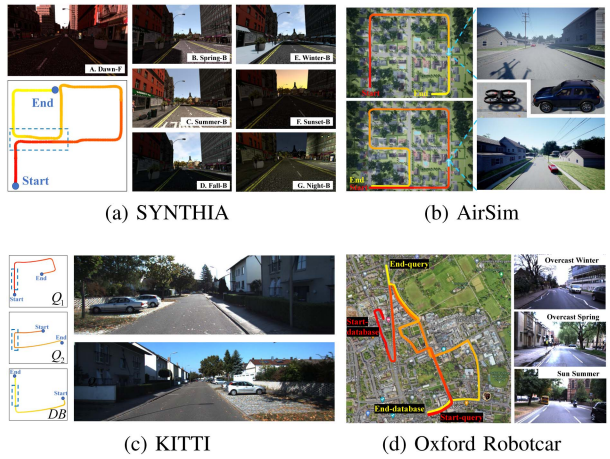
A. Results on the SYNTHIA Dataset

1) Dataset and Experiment Setup: The SYNTHIA dataset SEQS-02 is collected by a car with four cameras in a dynamic urban environment, travelling similar routes at different times of the year and day, as illustrated in Fig. 4(a).

To create test scenarios with large viewpoint and appearance variations, we use samples from the forward-view camera of the *Dawn* scene as database and samples from the backward-view camera of the *Spring*, *Summer*, *Fall*, *Winter*, *Sunset* and *Night* scenes as query sequences, similar to [7]. In this setting, only a small portion of the road exhibits query-database pairs with similar viewpoints (blue dashed box), while the remainder features large viewpoint difference.

2) Experiment Results: The quantitative results are shown in Table II, where bold and underline represent the best and second-best results, respectively. Clearly, our method achieves the best mF1 and AUC across all the six scenarios and significantly outperforms the compared methods, which can also be seen in Fig. 5. Regarding the metric R@100P, our method consistently attains substantial recall at 100% precision in various scenarios, with the highest average value.

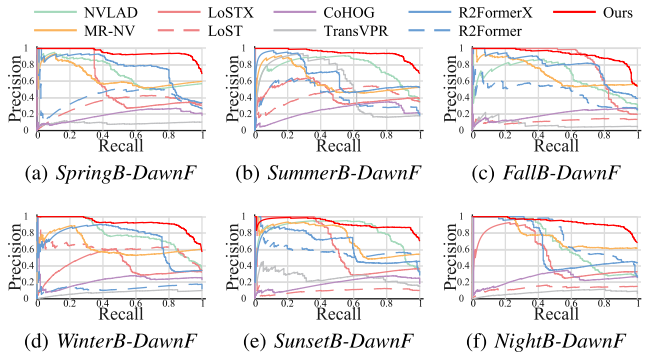
Furthermore, the R@100P of our method on the *SunsetB-DawnF* is only 1.1%, primarily due to the presence of extremely



(a) SYNTHIA (b) AirSim

(c) KITTI (d) Oxford Robotcar

Fig. 4. Illustration of datasets.



(a) *SpringB-DawnF* (b) *SummerB-DawnF* (c) *FallB-DawnF*

(d) *WinterB-DawnF* (e) *SunsetB-DawnF* (f) *NightB-DawnF*

Fig. 5. PRC results on the SYNTHIA dataset.

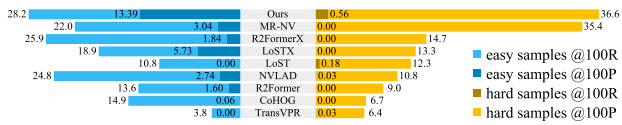


Fig. 6. The average ratio of true positive matches.

rare erroneous samples. It can be seen in Fig. 5(e) that the curve of our method experiences a downward spike followed by recovery. In contrast, though some methods achieve higher R@100P on one or two pairs, all of them exhibit a sharp drop in their curves, such as NetVLAD in Fig. 5(a), LoSTX in Fig. 5(c), and (e), and R2Former in Fig. 5(f). The reason is that these methods perform inadequately when faced with samples exhibiting large viewpoint difference. Consequently, as the threshold of similarity scores decreases, false positives (FP) increase rapidly, resulting in a sudden drop in precision. Besides, despite the limited count of true positives (TP), they can achieve a high recall at 100% precision due to minimal false negatives (FN).

To further illustrate this, Fig. 6 shows the average ratio of TP matches, categorizing matches within 30 degrees as 'easy' and others as 'hard'. Our method consistently attains the highest ratio of TP matches across all conditions.

3) Ablation Study: Table II provides the results of ablation studies of our method. 'Ours w/o CI' uses only the base solver

TABLE II
COMPARISON WITH SOTA METHODS ON THE SYNTHIA DATASET

Method	*SpringB-DawnF*			*SummerB-DawnF*			*FallB-DawnF*			*WinterB-DawnF*			*SunsetB-DawnF*			*NightB-DawnF*			*Average*		
	mF1	AUC	R@100P	mF1	AUC	R@100P	mF1	AUC	R@100P	mF1	AUC	R@100P	mF1	AUC	R@100P	mF1	AUC	R@100P	mF1	AUC	R@100P
NetVLAD [4]	0.727	0.708	0.003	0.765	0.770	-	0.664	0.666	0.006	0.741	0.796	0.303	0.802	0.834	-	0.598	0.684	0.222	0.716	0.743	0.089
LoST [7]	0.543	0.326	-	0.627	0.429	0.005	0.251	0.121	-	0.713	0.617	0.023	0.219	0.112	0.015	0.263	0.132	-	0.436	0.290	0.008
LoSTX [7]	0.514	0.578	0.343	0.569	0.450	-	0.790	0.829	**0.516**	0.538	0.385	-	0.587	0.649	**0.361**	0.521	0.537	-	0.587	0.571	0.203
CoHOG [23]	0.412	0.191	-	0.491	0.238	-	0.425	0.198	-	0.410	0.225	0.007	0.422	0.192	-	0.434	0.247	0.008	0.432	0.215	0.003
TransVPR [9]	0.188	0.107	0.015	0.621	0.566	-	0.178	0.084	-	0.196	0.078	-	0.403	0.269	-	0.204	0.081	-	0.298	0.198	-
MR-NV [5]	0.749	0.651	0.005	0.690	0.604	-	0.716	0.697	0.025	0.752	0.638	-	0.703	0.743	0.025	0.762	0.764	0.247	0.729	0.682	0.050
R2Former [10]	0.606	0.411	0.012	0.482	0.418	-	0.555	0.468	0.006	0.304	0.137	-	0.700	0.590	0.037	0.667	0.780	**0.313**	0.552	0.467	0.061
R2FormerX [10]	0.770	0.771	0.027	0.695	0.675	-	0.778	0.785	0.075	0.758	0.719	0.004	0.662	0.645	0.003	0.620	0.642	0.159	0.714	0.706	0.045
Ours w/o CI	0.817	0.786	0.083	0.822	0.778	-	0.711	0.720	0.092	0.810	0.670	-	0.820	0.792	0.009	0.811	0.798	-	0.799	0.757	0.031
Ours w/o AF	0.843	0.694	0.009	0.855	0.704	-	0.788	0.671	-	0.815	0.682	-	0.860	0.762	0.004	0.852	0.735	0.014	0.836	0.708	0.005
Ours w/o SL	0.912	0.913	0.014	0.853	0.892	0.004	0.865	0.918	0.242	0.784	0.819	0.003	0.866	0.908	0.185	0.865	0.915	0.299	0.858	0.894	0.125
Ours w WB	0.904	0.917	0.326	0.858	0.872	0.081	0.863	0.892	-	0.839	0.914	**0.307**	0.836	0.842	-	0.811	0.886	0.108	0.852	0.887	0.137
Ours w LAP	0.876	0.863	0.004	0.843	0.817	0.012	0.817	0.664	0.005	0.829	0.697	0.013	0.869	0.830	0.021	0.850	0.891	0.300	0.847	0.793	0.059
Ours	**0.923**	**0.951**	**0.347**	**0.901**	**0.949**	**0.206**	**0.876**	**0.930**	0.263	**0.876**	**0.926**	0.284	**0.904**	**0.921**	0.011	**0.897**	**0.938**	0.215	**0.896**	**0.936**	**0.221**

without adding constraints and iterative solving. 'Ours w/o AF' denotes the results without adaptive fusion. 'Ours w/o SL' does not use semantic labels, instead, it replaces them with the results of OneFormer. 'Ours w WB' shows the results with a weaker feature backbone, NetVLAD, instead of MR-NV. 'Ours w LAP' treats graph matching as a linear assignment problem (LAP), which is solved without considering edge similarity. It can be seen that each component contributes to the overall performance, and the combination of all components achieves the best performance. Note that though semantic segmentation with errors leads to a deterioration in AUC and mF1, our method still outperforms others, which demonstrates its robustness against semantic extraction errors. In addition, improving the accuracy of semantic segmentation helps improve the performance in terms of R@100P.

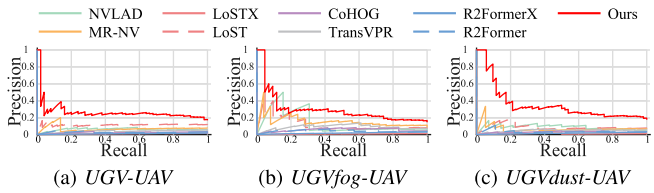
B. Results on the Custom AirSim Dataset

1) Dataset and Experiment Setup: Given the scarcity of datasets with opposing viewpoints, as mentioned in [2], we collected four sets of data in the AirSimNH simulation environment [31] utilizing ground vehicles and drones to create the custom AirSim dataset. As illustrated in Fig. 4(b), the lower and upper portions depict the trajectories collected by drone and ground vehicle, respectively. The data from the drone is used as database, referred to as *UAV*, while the data from the ground vehicle in different weather is used as queries, denoted as *UGV*, *UGVfog* and *UGVdust*. Semantic segmentation is obtained by OneFormer, whilst depth information is read from the simulator directly.

In this way, we construct a challenging dataset to simulate and assess the performance of VPR in the presence of heterogeneous data, extreme viewpoint variations and appearance changes. The dataset is availableat.[1]

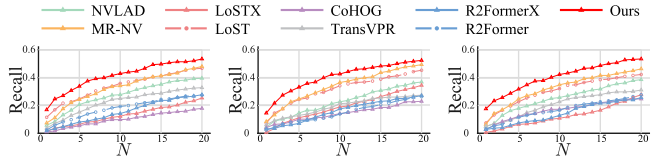
2) Experiment Results: The results depicted in Fig. 7 indicate that all the methods deteriorate dramatically in the considered highly challenging situations. Our method significantly outperform the compared ones and the advantage is more conspicuous at low recalls. For example, it achieves 100% precision at recalls of 2.3%, 5.0% and 6.3%, respectively.

Furthermore, in VPR applications, sequence data is typically used. Therefore, the presence of a correct match among the top N candidates can, to some extent, reflect the potential of obtaining correct matches leveraging sequence data. Considering the limited performance of all methods, we use Recall@N as a supplementary evaluation metric for further investigation, which is in line with existing works [1], [5], [9].



(a) *UGV-UAV* (b) *UGVfog-UAV* (c) *UGVdust-UAV*

Fig. 7. PRC reuslts on the custom AirSim dataset.



(a) *UGV-UAV* (b) *UGVfog-UAV* (c) *UGVdust-UAV*

Fig. 8. Recall@N results on the custom AirSim dataset.

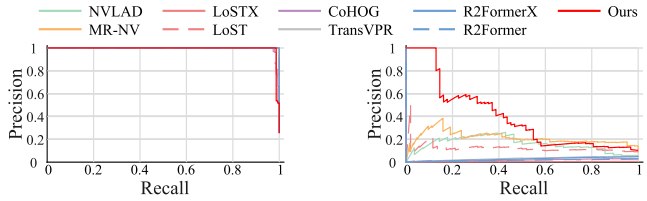
Fig. 9. Matching results on a typical example. Green and red boxes indicate correct and incorrect predictions.

The results are shown in Fig. 8. It is clear that our method outperforms the compared ones across different N values and shows a significant improvement. Additionally, unlike the results in Section IV-A, we observe that the reranking step actually has a detrimental effect on the performance of compared methods, as shown by the resutls of LoSTX and R2FormerX in Fig. 8. The two methods perform reranking based on feature points/regions in the front view. Consequently, they are less robust compared to their global descriptors in these challenging scenarios. Fig. 9 shows the results on a typical example by LoST, MR-NV and our method.

C. Results on the KITTI Dataset

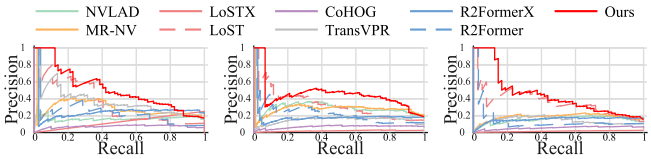
1) Dataset and Experiment Setup: To validate the effectiveness of our method in real-world urban scenarios, we conduct experiments on the *Sequence 16* of the KITTI dataset [32] for its large viewpoint variations. As shown in Fig. 4(c), we divide

[1]https://drive.google.com/drive/folders/1dOXKuRIGF7pDoZFKZbOJkE_4LVqRksIR?usp=sharing



(a) Q_1-*DB* (b) Q_2-*DB*

Fig. 10. PRC results on the KITTI dataset.



(a) *Overcast Autumn* (b) *Overcast Spring* (c) *Sun Summer*

Fig. 11. PRC results on the Oxford Robotcar dataset.

this sequence into three subsets, Q_1, Q_2 as queries, and *DB* as the database. The Q_1-*DB* pair includes repeated segments with similar viewpoints, while the Q_2-*DB* pair includes reverse viewpoints. We only use RGB as input, while semantic segmentation and depth estimation are obtained using OneFormer and Lite-Mono, respectively.

2) Experiment Results: Fig. 10(a) shows that when passing through the same place in the same direction, all methods perform well. However, for the difficult Q_2-*DB* pair, their performance deteriorates dramatically, consistent with previous results. In contrast, our method shows the least performance degradation, achieving a R@100P of 12.9%. Furthermore, our method yields an mF1 of 0.434, about 34.0% higher than the second-best method NetVLAD (0.324), and has an AUC of 0.399, about 83.0% higher than MR-NV (0.218).

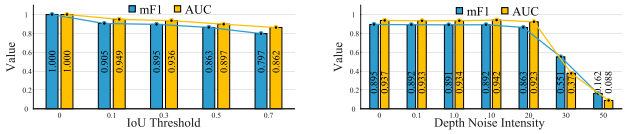
D. Results on the Oxford Robotcar Dataset

1) Dataset and Experiment Setup: The Oxford Robotcar dataset is collected by traversing a route through central Oxford multiple times. Similar to [38], we sample three sequences from the initial four km of the full traverse, *Overcast Autumn* (2014-12-09-13-21-02), *Overcast Spring* (2015-05-19-14-06-38) and *Sun Summer* (2015-07-29-13-09-26), and divide each one into query-database pair, as illustrated in Fig. 4(d). Each pair consists of two segments with large viewpoint difference. Other settings are the same as Section IV-C1.

2) Experiment Results: Fig. 11 shows that our method outperforms other methods, achieving R@100P of 12.3%, 6.7% and 12.5% on three pairs, respectively. Additionally, our method yields an average mF1 of 0.499, 17.1% higher than the second-best method LoST (0.426), and has an average AUC of 0.470, 33.1% higher than LoST (0.353). This demonstrates the superiority of our method in challenging real-world scenarios.

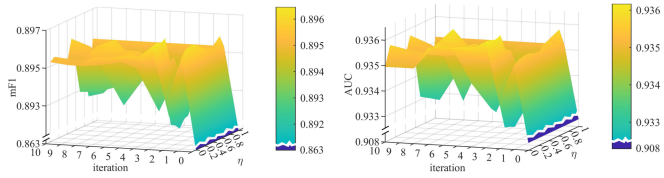
E. Sensitivity Analysis

1) Impact of IoU Threshold: In our experiments, the ground truth of a data pair depends on IoU threshold, as mentioned in Section IV. To evaluate the impact of IoU threshold, we conduct experiment on the SYNTHIA dataset with different IoU



(a) Impact of IoU threshold (b) Impact of depth noise

Fig. 12. Impact of IoU threshold and depth noise.



(a) Impact on mF1 (b) Impact on AUC

Fig. 13. Impact of solver parameters.

TABLE III
RUNTIME OF DIFFERENT SYSTEM COMPONENTS

Average Per Component	Global Description	Coarse Matching	Graph Construction	Graph Matching (top$N$=20)	Adaptive Reranking
Time (ms)	17.2	0.5	256.6	333.1	1.2

thresholds. The results are shown in Fig. 12(a). A high threshold implies more stringent criteria for determining whether two images belong to the same location, resulting in a decrease in the algorithm's performance. The experimental results are consistent with this, and it is smooth with respect to the different IoU. Considering the camera's field of view and scene characteristics, we set the IoU threshold to 0.3.

2) Impact of Depth Noise: In this experiment, we add Gaussian noise to the depth map to simulate the noise in the depth estimation on the SYNTHIA Dataset. Fig. 12(b) shows that the performance of our method monotonically decreases with increasing noise, and when the noise reaches a high level, there is a sharp drop in performance. This suggests that our method is not sensitive to depth noise within a certain range.

3) Impact of Solver Parameters: We assess the influence of the solver parameters on the SYNTHIA dataset. As depicted in Fig. 13, both mF1 and AUC demonstrate similar trends in response to variations of parameters. Iterative solving significantly enhances performance. The iterative algorithm converges within only a few iterations. Additionally, the negative feedback coefficient η has a relatively minor impact on performance.

F. Runtime Analysis

We conduct all experiments on a computer with a 3.70 GHz Intel i9-10900 K CPU and GeForce GTX 3090Ti GPU. Table III shows the average runtime per compomet on the *SpringB-DawnF* sequence.

V. CONCLUSION

A novel VPR method representing scenes as semantic topological graphs in the BEV has been proposed. It follows a hierarchical paradigm and leverages geometric consistency check based feedback to match a query with candidates iteratively. Further, an adaptive fusion strategy for effectively fusing the

similarity scores from the coarse and fine stages has been designed. Experiments on both simulated and real-world datasets, focusing on urban road scenes with significant appearance variations and viewpoint difference, demonstrated that the proposed method outperforms existing ones in challenging scenarios.

However, there are still some limitations. In scenarios where the flat world assumption used in IPM does not hold, such as jungles or uneven forest paths, our method would deteriorate. Moreover, the generalization of semantic segmentation models remains an issue, especially when facing large scene changes. In future work, we plan to enhance the generalization capability of our method.

REFERENCES

[1] K. A. Tsintotas, L. Bampis, and A. Gasteratos, "The revisiting problem in simultaneous localization and mapping: A survey on visual loop closure detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 19929–19953, Nov. 2022.

[2] M. Zaffar et al., "Vpr-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change," *Int. J. Comput. Vis.*, vol. 129, no. 7, pp. 2136–2174, 2021.

[3] D. Galvez-López and J. D. Tardos, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[4] 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.

[5] A. Khaliq, M. Milford, and S. Garg, "Multires-netvlad: Augmenting place recognition training with low-resolution imagery," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 3882–3889, Apr. 2022.

[6] C. Liu, H. Liu, L. Zhang, H. Zeng, L. Luo, and B. Fan, "Learning task-aligned local features for visual localization," *IEEE Robot. Automat. Lett.*, vol. 8, no. 6, pp. 3366–3373, Jun. 2023.

[7] S. Garg, N. Suenderhauf, and M. Milford, "Lost? Appearance-invariant place recognition for opposite viewpoints using visual semantics," in *Proc. Robot.: Sci. Syst.*, 2018, pp. 1–11.

[8] H. Yue, J. Miao, Y. Yu, W. Chen, and C. Wen, "Robust loop closure detection based on bag of superpoints and graph verification," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2019, pp. 3787–3793.

[9] 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. 13638–13647.

[10] S. Zhu, L. Yang, C. Chen, M. Shah, X. Shen, and H. Wang, "r^2 former: Unified retrieval and reranking transformer for place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 19370–19380.

[11] J. Ma, X. Ye, H. Zhou, X. Mei, and F. Fan, "Loop-closure detection using local relative orientation matching," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7896–7909, Jul. 2022.

[12] C. Liu and S. Shen, "Towards view-invariant and accurate loop detection based on scene graph," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2023, pp. 2127–2133.

[13] 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 Robot. Automat. Lett.*, vol. 6, no. 4, pp. 8349–8356, Oct. 2021.

[14] S. Lin, J. Wang, M. Xu, H. Zhao, and Z. Chen, "Topology aware object-level semantic mapping towards more robust loop closure," *IEEE Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7041–7048, Oct. 2021.

[15] X. Ji, P. Liu, H. Niu, X. Chen, R. Ying, and F. Wen, "Object slam based on spatial layout and semantic consistency," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 2528812.

[16] A. Gawel, C. D. Don, R. Siegwart, J. Nieto, and C. Cadena, "X-view: Graph-based semantic multi-view localization," *IEEE Robot. Automat. Lett.*, vol. 3, no. 3, pp. 1687–1694, Jul. 2018.

[17] P. Neubert, S. Schubert, K. Schlegel, and P. Protzel, "Vector semantic representations as descriptors for visual place recognition," in *Proc. Robot.: Sci. Syst.*, 2021, pp. 1–11.

[18] P. Hou, J. Chen, J. Nie, Y. Liu, and J. Zhao, "Forest: A lightweight semantic image descriptor for robust visual place recognition," *IEEE Robot. Automat. Lett.*, vol. 7, no. 4, pp. 12531–12538, Oct. 2022.

[19] Z. Li et al., "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," in *Proc. Eur. Conf. Comput. Vis.*, 2022, vol. 13669, pp. 1–18.

[20] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 194–210.

[21] M. Cho, J. Lee, and K. M. Lee, "Reweighted random walks for graph matching," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 492–505.

[22] R. Wang, Y. Zhang, Z. Guo, T. Chen, X. Yang, and J. Yan, "Linsatnet: The positive linear satisfiability neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 36605–36625.

[23] M. Zaffar, S. Ehsan, M. Milford, and K. McDonald-Maier, "CoHOG: A light-weight, compute-efficient, and training-free visual place recognition technique for changing environments," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 1835–1842, Apr. 2020.

[24] Y. Shen, S. Zhou, J. Fu, R. Wang, S. Chen, and N. Zheng, "StructVPR: Distill structural knowledge with weighting samples for visual place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 11217–11226.

[25] Z. Qian, J. Fu, and J. Xiao, "Towards accurate loop closure detection in semantic SLAM with 3D semantic covisibility graphs," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 2455–2462, Apr. 2022.

[26] J. Yu and S. Shen, "Semanticloop: Loop closure with 3D semantic graph matching," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 568–575, Feb. 2023.

[27] M. Ester et al., "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. ACM SIGKDD Conf. Knowl. Discov. Data Mining*, 1996, vol. 96, no. 34, pp. 226–231.

[28] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *Proc. IEEE Intell. Veh. Symp.*, 2007, pp. 315–320.

[29] J. Munkres, "Algorithms for the assignment and transportation problems," *J. Soc. Ind. Appl. Math.*, vol. 5, no. 1, pp. 32–38, 1957.

[30] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3234–3243.

[31] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSIM: High-fidelity visual and physical simulation for autonomous vehicles," in *Proc. Field Serv. Robot.*, 2017, pp. 621–635.

[32] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[33] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.

[34] S. Hausler, A. Jacobson, and M. Milford, "Multi-process fusion: Visual place recognition using multiple image processing methods," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1924–1931, Apr. 2019.

[35] C. Masone and B. Caputo, "A survey on deep visual place recognition," *IEEE Access*, vol. 9, pp. 19516–19547, 2021.

[36] J. Jain, J. Li, M. Chiu, A. Hassani, N. Orlov, and H. Shi, "Oneformer: One transformer to rule universal image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 2989–2998.

[37] N. Zhang, F. Nex, G. Vosselman, and N. Kerle, "Lite-mono: A lightweight CNN and transformer architecture for self-supervised monocular depth estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 18537–18546.

[38] S. Garg, N. Suenderhauf, and M. Milford, "Semantic–geometric visual place recognition: a new perspective for reconciling opposing views," *Int. J. Robot. Res.*, vol. 41, no. 6, pp. 573–598, 2022.