2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

SMORE-SLAM: Semantic Monocular SLAM with Scale Correction and Reverse Loop Utilization in Outdoor Environments

Yushi Chen[1], Fang Zhao[1], Yue Zhuge[2], Junxiong Liu[1], Jiaquan Yan[1] and Haiyong Luo[2]

***Abstract*— In large-scale outdoor environments, vehicles often encounter situations like retracing their path or turning around, leading to many reverse loop closures where the vehicles traverse previously covered paths from opposite viewpoints. Existing monocular SLAM methods, due to insufficient utilization of semantic information and neglect of leveraging reverse loop closures, result in significant scale drift and pose drift when confronted with such scenarios. In this paper, we introduce SMORE-SLAM, a semantic monocular SLAM with scale correction and reverse loop closure module. We constrain scale drift by harnessing semantic information across a wide spatial extent. Furthermore, we detect and correct reverse loop closures using semantic point cloud to reduce pose drift. Experimental results on the KITTI odometry dataset and the Oxford RobotCar dataset demonstrate the capability of our research in scale correction and reverse loop closure detection, enabling a reduction in trajectory errors of monocular SLAM.**

I. INTRODUCTION

Monocular Simultaneous Localization and Mapping (Monocular SLAM), requiring only a single camera as a sensor, possesses advantages such as low cost, flexibility, and portability, and finds wide applications in the field of visual localization and tracking for mobile platforms like vehicles and robots [1]–[7]. However, due to the absence of reliable measurement constraints and the accumulation of errors caused by incremental localization, monocular SLAM systems often encounter significant drift from both scale and pose, resulting in substantial errors, particularly in large-scale outdoor environments.

To address this issue, most monocular SLAM systems [1]–[4], [8] utilize loop closures to impose long-term constraints on keyframes and correct errors. Thus, every potential loop closure within the environment is of great importance. Research on loop closure detection and visual place recognition have seen rapid development [9]–[17].

Nevertheless, there are many instances where vehicles return along the same route or turning around during outdoor driving, resulting in loop closures which have completely opposite viewpoints. Existing loop closure detection methods have limitations in detecting those reverse loop closures, as illustrated in Fig. 1-a, rendering the system unable to exploit them for error correction.

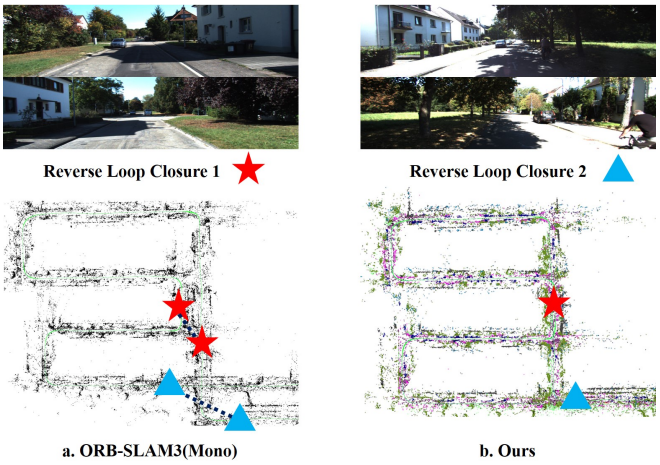


Fig. 1: Previous method is unable to correct the reverse loop closures indicated by the star and triangle symbols in the image, whereas our system successfully corrects them and the drifted scale. This capability significantly reduces the trajectory errors.

Some researchers introduce semantic information into monocular SLAM to solve drift problem. [18], [19] utilize the triangulated points of consecutive frames from regions of interest (ROI) to correct the drifted scale. [20], [21] employ object-level semantic information to graph optimization to prevent the drift. Nonetheless, ROI features and the semantics of specific objects may downgrade the robustness of SLAM. [22], [23] harness the point-level semantic information to reduce the accumulate errors in visual localization. However, a semantic map is required beforehand.

We propose a monocular SLAM system that achieves scale correction and reverse loop utilization based on semantic point cloud in outdoor environments. By fusing semantic and spatial information of local point cloud, we create semantic visual point cloud descriptors tailored for visual SLAM. The descriptors enable monocular SLAM to detect and correct loop closures with opposing viewpoint, as illustrated in Fig. 1-b. Moreover, we employ semantic-based scale correction, which utilizes semantic points from ground to rectify scale drift of monocular SLAM and maintain the scene representation capability of those descriptors.

*This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDA28040500, the National Natural Science Foundation of China under Grant 62261042, the Key Research Projects of the Joint Research Fund for Beijing Natural Science Foundation and the Fengtai Rail Transit Frontier Research Joint Fund under Grant L221003, Beijing Natural Science Foundation under Grant 4222034, the Science and Technology Plan Project of Inner Mongolia Autonomous Region under Grant 2019GG328 and the Yibin City Introduction of High-Level Talent Project under Grant 2022YG03 (Corresponding author: Haiyong Luo, Fang Zhao)

[1]Beijing University of Posts and Telecommunications, Beijing, China {chenyushi, zfsse, liujunxiong, YanJiaguan}@bupt.edu.cn

[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China {zhugeyue21s, yhluo}@ict.ac.cn

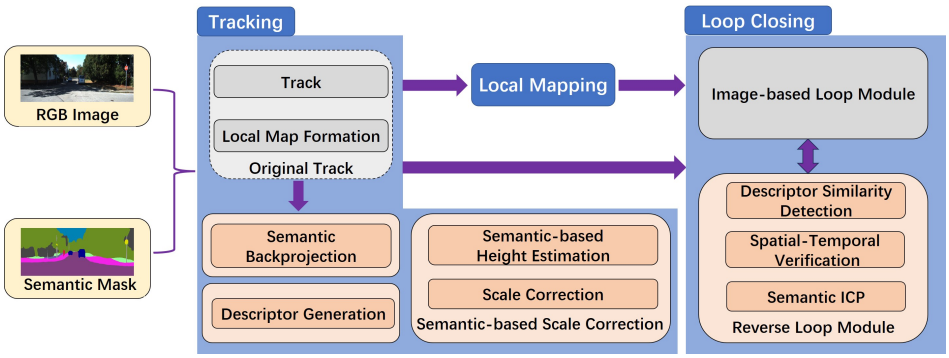


Fig. 2: The figure above illustrates all the steps of tracking and loop closure threads in SMORE-SLAM. The gray boxes represent the original steps of ORB-SLAM3, and the orange boxes denote our proposed modules in SMORE-SLAM, which will be elaborated in the following sections.

Experiments conducted on the KITTI odometry dataset [24] and the Oxford RobotCar dataset [25] validate the effectiveness of our proposed method, which successfully reduces trajectory errors through semantic-based scale correction and reverse loop closures. The primary contributions of this paper can be summarized as follows:

- a semantic monocular SLAM system with scale correction and utilization of reverse loop closures, which can effectively mitigate the drift errors.
- a semantic visual point cloud descriptor suitable for visual SLAM, which can be harnessed for detecting reverse loop closures.
- a semantic-based scale correction method which can restrain scale drift in monocular SLAM and maintain the representation capability of semantic visual point cloud descriptor.

II. RELATED WORK

A. Semantic Monocular SLAM

Monocular SLAM is a cost-effective solution for visual localization and tracking on vehicle platforms. Nevertheless, due to the lack of reliable constraints, monocular SLAM encounters challenges related to scale drift and pose drift, which can cumulatively result in substantial errors.

Loop closure detection is commonly employed to associate keyframes at the same locations, enabling corrections to achieve global consistency. Some researchers attempt to employ semantic information to rectify drift errors. There has been a considerable amount of work in this regard:

- **Semantics from ground:** [18] constrains scale drift of monocular structure from motion to some extent by integrating ground points and prior camera height. [19] further improves the points selection strategy and scale estimation methods. Both of them select points generated by adjacent frames from ROI. Unlike them, we map semantics for point cloud to identify ground points more accurately and select them from a much larger area for scale correction in monocular SLAM.
- **Object-level semantics:** [20], [21], [26] introduce object-level information into SLAM systems, incorporating object shapes and poses to constrain drift. Unlike them, we exploit more common outdoor environmental semantics for robustness rather than relying on specific objects.
- **Point-level semantics:** [27] employs semantic masks to mitigating the impact of dynamic objects on the tracking process. [22], [23] leverage the semantic segmentation results and geometrical structures of a prior map to bound drift errors of visual localization. In this paper, we only focus on solving the drift issue inherent to monocular SLAM through semantics and do not rely on prior maps.

B. Loop Closure Detection

Loop closure detection is a common way to bound the drift errors of SLAM system. A large amount of approaches have been proposed:

- **Image-based method:** In visual SLAM, loop closure detection commonly rely on image similarity. Approaches like [28], [29] represent image features as feature vectors and perform loop detection by comparing the similarity of these vectors. Some deep learning models for visual place recognition [9]–[14] extract descriptors from images to compute image similarity. However, they have almost neglected the issue of reverse loop closures. While LoSTX [30] leverage semantic information to recognize locations from opposing viewpoints, the substantial visual offset required poses challenges for integration into a visual SLAM system.
- **Point-based method:** Some researchers employ point cloud similarity for loop detection. [31] describes scenes using the height and spatial distribution of LiDAR point cloud and provides relative rotation information when calculating similarity. [32] introduces the approach in [31] into visual SLAM, enabling direct method SLAM to detect loop. However, [32] requires additional stereo data for scale optimization and still uses the low-level height features to encode point cloud descriptor. To enhance the representation capability of point cloud descriptor, some LiDAR-based methods attempt to utilize high-level features such as intensity [33] and

semantics [34], [35]. In this paper, we introduce image semantics into monocular SLAM, mapping it into point cloud. We leverage the semantic point cloud to construct descriptors and constrain drift of monocular SLAM.

- **Semantic-based method:** Some visual loop detection methods integrate with the semantic information of the environments. [15] utilizes object detection to detect loops under large viewpoint deviations. [16] constructs scene descriptors based on the detected object categories and their relative orientation for loop detection. [17] employs object landmarks and their inter-object topology to recognize loop closures. However, these methods are solely applicable to indoor scenes at the room-scale level. It is more challenging in large-scale outdoor scenes without feature-rich objects and topological structures. In this paper, we combine semantic segmentation models with visual point cloud to describe the surrounding environments and further mitigate the drift errors of monocular SLAM in outdoor scenarios.

III. METHOD

A. System Overview

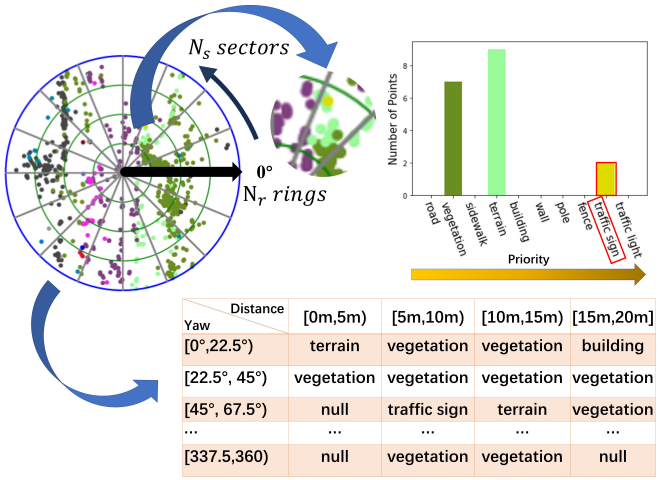
Fig. 2 presents an overview of our SMORE-SLAM system. We take original images and their semantic masks as input data. We backproject semantic information from feature points to point cloud and create semantic visual point cloud descriptor from the local map of current frame. A semantic-based scale correction is following to reduce the scale drift. In the loop closing thread, we design a reverse loop module as a complement to the original image-based loop module. For keyframes where no loop closure is detected in image-based loop module, we utilize the reverse loop module for further detection and validation. Relative rotation between current frame and loop candidates is provided during descriptor similarity detection. Subsequently, forward loop candidates are sent back to original loop module of [3] for verification and correction, while reverse loop candidates undergo spatial-temporal verification and validated loop closures are corrected.

B. Semantic Visual Point Cloud Descriptor

Inspired by [31], we construct semantic visual point cloud descriptor within monocular SLAM, enabling it to detect reverse loop closures. Compared to the LiDAR point cloud utilized in [31], the point cloud generated by monocular SLAM are sparse and inaccurate. To address this limitation in visual SLAM, we adopt the following approach to construct semantic visual point cloud descriptors.

Local Map Formation: We designate keyframes that have enough overlap with the current frame as local keyframes. The observed points from these keyframes constitute the current local map.

Semantic Backprojection: We perform semantic segmentation and ORB features detection on the images. For those features that obtain depth information through triangulation, we leverage the correspondence between pixels and local map points to label the point cloud with semantics, achieving the backprojection of semantic information.



Yaw \ Distance	[0m,5m)	[5m,10m)	[10m,15m)	[15m,20m]
[0°,22.5°)	terrain	vegetation	vegetation	building
[22.5°, 45°)	vegetation	vegetation	vegetation	vegetation
[45°, 67.5°)	null	traffic sign	terrain	vegetation
...	...	...	...	...
[337.5,360)	null	vegetation	vegetation	null

Fig. 3: Generation process of semantic visual point cloud descriptor. High-priority semantics provide values for corresponding areas. For areas without point cloud, a *null* value is used to represent them.

Descriptor Generation: For each input frame, we establish a polar coordinate system centered around the current camera within a horizontal circular area of radius R.

As depicted in Fig. 3, we divide the circular region into N_r rings based on distance to the center. Each ring area has the same width. On this foundation, we further partition the circular region into N_s sectors evenly. Consequently, the entire circular region is divided into $N_r \times N_s$ areas. The condition of each area can be represented as:

$$A_{ij} = \{l_k, r_k, \theta_k \mid \frac{(i-1)\cdot R}{N_r} \le r_k < \frac{i \cdot R}{N_r}, \frac{(j-1)\cdot 2\pi}{N_s} \le \theta_k < \frac{j \cdot 2\pi}{N_s}\} \quad (1)$$

where $A_{ij} \in \mathbb{R}^3$ represents the point cloud configuration within the area formed by the overlap of the i-th ring and the j-th sector. For points within a distance of R to the current camera from the local map, we determine their corresponding area based on the radius and angle of their projected points in the current polar coordinate system. l_k, r_k, and θ_k indicate the semantic label, polar radius, and polar angle of certain point k within a specific area respectively. To obtain a stable semantic point cloud, we select points that are observed by at least three keyframes and maintain semantic consistency.

We assign priorities to semantic labels within the environment based on a statistical analysis on the occurrences of semantic labels in the large-scale outdoor scenes dataset BDD100K [36]. Higher priority is assigned to labels with fewer occurrences. To prevent dynamic semantics from interfering with the descriptor's representation capability, we exclusively considered static semantic features (excluding vehicles and pedestrians). Each semantic label is represented by a unique semantic value. By harnessing the semantic features within each area, we can formulate a semantic

Fig. 4: Semantic-based height estimation. We harness the points with ground semantic in descriptor to estimate the camera height relative to the current ground plane.

visual point cloud descriptor $D \in \mathbb{R}^{N_r \times N_s}$ for each frame, represented as a 2D matrix that effectively characterizes the surrounding environments. We encode the element in row i and column j of descriptor matrix $D_j^i \in \mathbb{R}$ using:

$$D_j^i = \mathrm{f}(A_{ij}) \tag{2}$$

where the function f indicates acquiring the semantic value of point with the highest priority within certain area. A special value $null$ is used to represent an area without point cloud. The constructed descriptor will be used in III.C and III.D.

C. Semantic-based Scale Correction

Monocular SLAM is susceptible to scale drift over time due to the absence of precise depth information constraints. Therefore, we combine semantic information to perform scale correction with the following steps.

Semantic-based Height Estimation: As illustrated in Fig. 4, we extract semantic points labeled as "road" from the point cloud of current descriptor. We assume that the pitch angle of the camera is known and unchanged, allowing us to obtain the normal vector n of the current ground plane. For each ground point X within the descriptor, we can compute the corresponding height value h using:

$$n^T X = h \tag{3}$$

We compute the score of differences between the height values from different ground points using:

$$\eta = \max_p \left(\sum_{q \neq p} \exp\left(-\mu \left| h_p - h_q \right|_1\right) \right) \tag{4}$$

where we set μ as 50. The height h_p calculated for each ground points is compared to the heights h_q calculated from other ground points to assess their differences. A higher η value indicates smaller differences. We select the h_p with the highest η as the optimal camera height for current frame.

This strategy is similar to [18], with the difference that we do not select the points triangulated by the consecutive frames within ROI. Instead, we employ a broader extraction range in local map and ground points recognized through semantic segmentation to improve accuracy and robustness of our camera height estimation results.

Scale Correction: We figure out the scale factor with:

$$s = \frac{H}{h_p} \tag{5}$$

where H represents the true camera height. Comparing estimated height with the actual camera height allows us to determine the scale factor s. We harness s to correct the scale drift of camera pose and point cloud at each keyframe.

D. Loop Detection and Correction

In this section, we describe the process of detecting and correcting reverse loop closures using semantic visual point cloud descriptors. False loop closures can result in significant trajectory errors. To prevent the utilization of incorrect loop closures within the system, we consider following steps for reverse loop closures.

Image-based Loop Detection: In the loop closure detection thread, we first perform image-based loop closure detection. If a loop is detected for the current frame, validation and correction proceed as per the original flow. If no loop is found for the current frame, we switch to the reverse loop closure module. The reason for this approach is the presence of some unstable corner cases in our visual point cloud descriptor during forward loop closure detection, which will be analyzed in detail in the experimental section.

Descriptor Similarity Detection: We calculate the similarity between the semantic visual point cloud descriptor of current frame ${}_{\mathbf{cur}}D \in \mathbb{R}^{N_r \times N_s}$ and the descriptor of certain candidate frame ${}_{\mathbf{cand}}D \in \mathbb{R}^{N_r \times N_s}$ using:

$$\begin{aligned} \mathrm{F}({}_{\mathbf{cur}}D, {}_{\mathbf{cand}}D) = & \frac{\sum_{i=1}^{N_r}\sum_{j=1}^{N_s} \mathrm{g}({}_{\mathbf{cur}}D_j^i == {}_{\mathbf{cand}}D_j^i))}{N_r \times N_s - \tau({}_{\mathbf{cur}}D, {}_{\mathbf{cand}}D)} \\ & - \frac{\tau({}_{\mathbf{cur}}D, {}_{\mathbf{cand}}D)}{N_r \times N_s - \tau({}_{\mathbf{cur}}D, {}_{\mathbf{cand}}D)} \end{aligned} \tag{6}$$

$$\begin{aligned} \tau({}_{\mathbf{cur}}D, {}_{\mathbf{cand}}D) = \sum_{i=1}^{N_r}\sum_{j=1}^{N_s} \mathrm{g}({}_{\mathbf{cur}}D_j^i == \text{null} \\ \text{and } {}_{\mathbf{cand}}D_j^i == \text{null}) \end{aligned} \tag{7}$$

$$\mathrm{g}(x) = \begin{cases} 1 & \text{if } x \text{ is true} \\ 0 & \text{if } x \text{ is false} \end{cases} \tag{8}$$

Considering potential rotation between descriptors from different frames, we can perform column shift operation on the descriptors based on :

$$\mathrm{shift}(D, r) = \mathrm{CONCAT}_{x=0}^{N_s-1}(D_{(x+r)\%N_s}) \tag{9}$$

D represents a constructed semantic visual point cloud descriptor as III.B. r is the input rotation coefficient. $D_{(x+r)\%N_s} \in \mathbb{R}^{N_r}$ means retrieving the $(x+r)\%N_s$-th column of D. Function $\mathrm{CONCAT}_{x=0}^{N_s-1}$ concatenates those column vectors one by one to form a new descriptor matrix.

Performing an shift operation on the column vectors of descriptor D is equivalent to rotate the point cloud within the descriptor by $\frac{360r}{N_s}$ degrees. This operation helps us generate semantic visual point cloud descriptors under different rotational angles, as illustrated in Fig. 5.

We perform column shift operation on candidates for all angles as shown in:

$$\mathrm{E}({}_{\mathrm{cur}}D, {}_{\mathrm{cand}}D) = \max_{r^* \in [N_s-1]} \mathrm{F}({}_{\mathrm{cur}}D, \mathrm{shift}({}_{\mathrm{cand}}D, r^*)) \tag{10}$$

TABLE I: Absolute Trajectory Error of different monocular SLAM systems

Method	Sequences with Reverse Loop				Sequences with Forward Loop				Sequences without loop				
	02	08	Oxford1	Oxford2	00	05	06	07	01	03	04	09	10
ORB-SLAM3(Mono)	47.24	162.58	234.52	190.16	14.66	6.36	16.14	4.85	628.47	7.71	4.66	21.59	26.82
Song	28.55	26.67	56.75	89.37	8.87	8.94	11.29	4.73	380.57	4.90	4.48	13.24	11.83
Cube-SLAM	26.20	10.70	109.03	61.70	13.90	4.75	**6.98**	2.67	671.22	**3.79**	**1.10**	10.70	**8.37**
DSP-SLAM	21.64	12.40	70.38	62.51	12.30	**4.39**	7.88	4.73	574.00	5.77	2.89	10.81	9.98
Ours wo SSC	54.27	69.52	190.16	82.15	X	5.82	16.39	4.82	644.15	7.81	3.42	19.07	29.36
Ours wo RLC	20.33	14.22	30.65	30.40	8.85	5.31	8.99	**2.45**	351.00	4.50	2.33	**6.16**	9.04
Ours	**15.93**	**7.52**	**16.20**	**22.61**	**8.22**	5.54	8.90	2.53	**324.72**	4.40	2.15	6.66	9.10

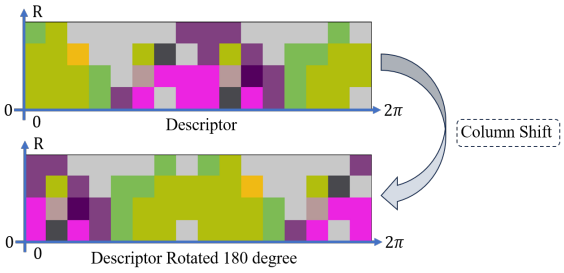


Fig. 5: Semantic visual point cloud descriptor and its 180-degree rotated counterpart.

where the maximum similarity score is chosen as the final similar score. The corresponding rotation coefficient r^* can provide a coarse-grained rotation relationship between two frames, which can be further utilized to calculate the relative rotation angles. Pairs with rotational angles falling within the forward loop range are sent back to image-based loop module for verification and correction. In the case of reverse loop closures, we process those pairs with following spatial-temporal verification.

Spatial-Temporal Verification: If a loop closure pair is identified, typically adjacent frames in the temporal sequence also share loop closure relationships. We calculate the average similarity in the temporal sequences for validation of reverse loop closures using:

$$\mathrm{T}({}_{\text{cur}}D, {}_{\text{cand}}D) = \frac{1}{N}\sum_{t=0}^{N}\mathrm{E}({}^{-t}_{\text{cur}}D, {}^{+t}_{\text{cand}}D) \tag{11}$$

where we set N as 3. ${}^{-t}_{\text{cur}}D$ represents the descriptor of the t-th frame before the current frame, while ${}^{+t}_{\text{cand}}D$ represents the descriptor of the t-th frame after the candidate frame. We use a threshold of 0.6 to result of T for further refinement of the previous candidates. Theoretically, our method can also detect lateral loop closures (often occurring at crossroads), but they are difficult to validate through the verification due to their brief spatial-temporal overlap. To minimize the occurrence of false positives, we opt to retain spatial-temporal verification and only focus on detecting reverse loop closures.

After the aforementioned verification, if there are candidate frames remaining, the one with the closest distance to the current frame is selected as the final loop closure frame.

Semantic ICP: We employ the ICP (Iterative Closest Point) algorithm for loop closure correction between current frame and loop frame. Rough alignment of point cloud in the current descriptor and candidate descriptor is achieved using the rotation coefficients r^* provided in (10), serving as an initial guess for ICP. Considering the inaccuracy in height associated with the visual point cloud, we only perform correction on the 3 degrees of freedom in the pose (horizontal coordinates and yaw angle). Moreover, we enforce semantic consistency and mutual proximity during point cloud registration for robust matching. During iterations, we optimize the distance error between the two sets of point cloud by:

$$R^*, t^* = \arg\min_{R,t}\frac{1}{2}\sum_{i=1}^{n}\|{}^{c}p_i - (R^{c}p'_i + t)\|_2^2 \tag{12}$$

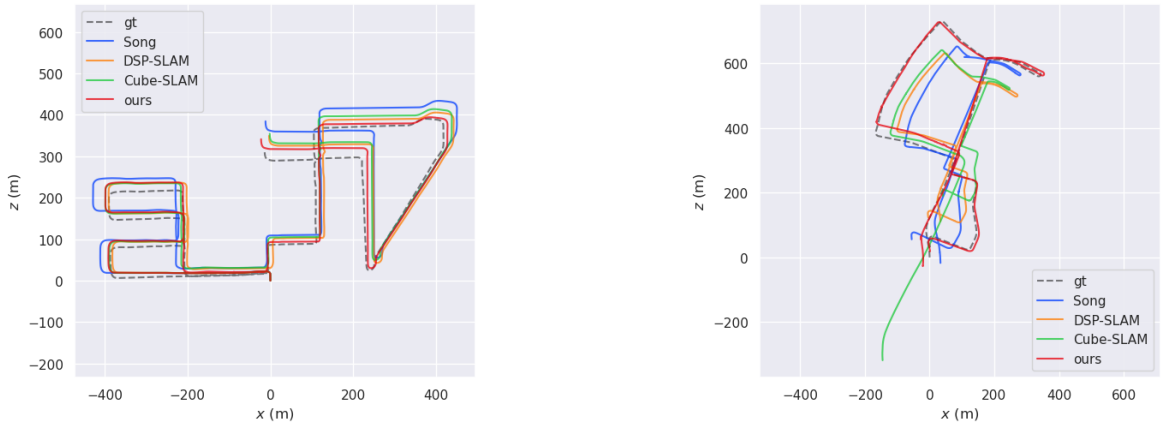
where R, t are relative rotation and translation between the two frames, n is the number of points in descriptor of current frame, p is the point from current descriptor, p' is the matched point of p from the loop descriptor. c represents the semantic category of certain point. If the optimization converges, we rectify the reverse loop closure with optimized pose; otherwise, we disregard this loop closure.

IV. EXPERIMENTATION AND ANALYSIS

A. Experimental Setup

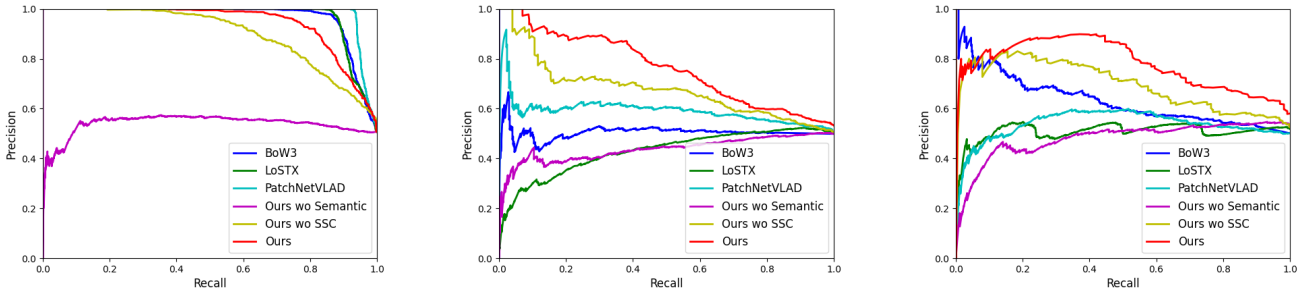
Datasets: Our experiments are performed on image sequences from the KITTI odometry dataset [24] and the Oxford RobotCar dataset [25]. The KITTI odometry dataset comprises 11 image sequences with ground truth poses, including sequences 00, 02, 05, 06, 07, and 08 that contain loop closures, with sequences 02 and 08 containing reverse loop closures. The Oxford RobotCar dataset consists of images captured along the long route under varying weather and road conditions. For our validation, we use all the KITTI odometry sequences and a subset of sequences with reverse loop closures from the sequences 2014-12-02-15-30-08 (referred to as oxford1) and 2014-12-05-15-42-07 (referred to as oxford2) of Oxford RobotCar dataset.

Other settings: We employ the PSPNet [37] trained on [38] and [36] dataset as our semantic segmentation model. Throughout the experiments, we set R=20, N_r=8, and N_s=16. All experiments are carried out on the same system with Intel i7-10750H @2.60GHz CPU, 16GB RAM and RTX 2070 GPU.



(a) KITTI 08 (b) Oxford1

Fig. 6: Trajectories of different monocular SLAM system on sequences with reverse loop closures.



(a) Forward (b) KITTI Reverse (c) Oxford Reverse

Fig. 7: P-R curves of visual loop detection methods on different sequences.

B. Trajectory Drift Evaluation

Baselines and Detail: To validate the effectiveness of our system in reducing trajectory drift, we perform the trajectory drift evaluation.

We compare our system with classic monocular SLAM [3](ORB-SLAM3(Mono)), ground-based monocular SLAM with method from [18](Song), semantic monocular SLAM [20], [21](Cube-SLAM and DSP-SLAM). We also conducted a comparison with our system without semantic-based scale correction (Ours wo SSC) and our system without a reverse loop closure module (Ours wo RLC) for ablation study.

Due to the scale ambiguity inherent in monocular SLAM, we followed the approach in Cube-SLAM code. Specifically, we utilized stereo matching on the first frame to obtain an initial scale, enabling the measurement of drift errors in monocular SLAM. The results of Cube-SLAM on the KITTI dataset are derived from their paper, other experimental results are reproduced based on their paper or code.

Metric: The evaluation metric used is the Absolute Trajectory Error (ATE) measured in meters. ATE quantifies the absolute positional difference between the estimated trajectory and the ground truth.

Result Analysis: The experimental results are shown in Table I. The lowest errors are marked in bold and the second lowest errors are underlined. "X" means system failed.

Compared to other methods, our system effectively reduces drift errors through scale correction and utilization of reverse loop closures in all the sequences with reverse loops. Partial visualized results are provided in Fig. 6.

As for other sequences, our method also achieves relatively low errors. Here, semantic-based scale correction plays a prominent role. The difference in error between our full system and system without reverse loop closure module is relatively small. KITTI 01 sequence depicts the vehicle swiftly traversing along a highway, a scenario prone to rapid scale drift within short time and lacking rich semantic information. While all monocular SLAM methods exhibit significant errors, our approach serves to mitigate the effects of drift to a certain extent with robust scale correction.

Although our system without semantic-based scale correction can reduce pose drift by detecting and correcting reverse loop closures, and can partially correct scale drift by incorporating essential graph optimization from ORB-SLAM, the impact of scale drift on the positions of surrounding point cloud affects the representation capability of semantic point cloud descriptors. As a result, the detection capability of our method decreases, leading to increased drift errors and tracking lost in a few sequences.

TABLE II: Comparison of visual loop detection methods on different sequences

Method	Forward	KITTI Reverse	Oxford Reverse
BoW3	0.965	0.505	0.632
LoSTX	<u>0.966</u>	0.428	0.508
PatchNetVLAD	**0.980**	0.590	0.534
Ours wo Semantic	0.458	0.437	0.479
Ours wo SSC	0.879	<u>0.678</u>	<u>0.694</u>
Ours	0.948	**0.763**	**0.776**

C. Loop Closure Evaluation

Baselines and Detail: To validate the effectiveness of our loop closure detection algorithm, we select positive and negative loop pairs based on ground truth poses from the KITTI and the Oxford RobotCar dataset. We follow the setting outlined in [34], considering sample pairs within a distance of 3 meters as positive samples and sample pairs beyond 20 meters as negative samples. We compare our descriptor similarity detection method with BoW3 [28], LoSTX [30] and PatchNetVLAD [12]. Additionally, we test the detection capability of descriptors using the maximum point height like original ScanContext (Ours wo Semantic) and descriptors without semantic-based scale correction (Ours wo SSC).

Metric: The evaluation metric used is PR-AUC. A higher value indicates better loop detection performance.

Result Analysis: The experimental results are presented in Table II and Fig. 7. The highest scores are marked in bold and the second highest scores are underlined. "Forward" consists of pairs from the KITTI 00 and 05, while "KITTI Reverse", "Oxford Reverse" are composed of pairs from the KITTI 08 and the Oxford RobotCar sequences respectively, which only contain reverse loops. Compared to other methods, our algorithm demonstrates a significant advantage in detecting reverse loop closures.

Since the local map does not contain point cloud information from the rear at the starting point, our descriptor tends to generate more $null$ entries in such conditions, which imposes some limitations on detecting related forward loop closures. This is why we retain image-based loop closure module in our system. Nevertheless, our method still achieves a competitive detection performance in "Forward" sequence.

We also observe a decrease in descriptor representation capability when semantic features are not utilized. This may be attributed to the sparsity of the visual point cloud and inaccuracies in the positions of the points. The detection capability of our method is also compromised when scale correction is not applied, which is consistent to the analysis in trajectory drift evaluation.

As mentioned in section II.B, LoSTX also focuses on addressing the challenge of reverse viewpoint loop closure detection. However, their performance in the above experiments was unsatisfactory, potentially due to its requirement for an adequate "visual offsets" to match landmarks in the images. We reselected reverse loop closure samples within 30 meters distance as positive samples and image pairs beyond 100 meters as negative samples from the KITTI dataset, resulting in the outcomes depicted in Fig 8.

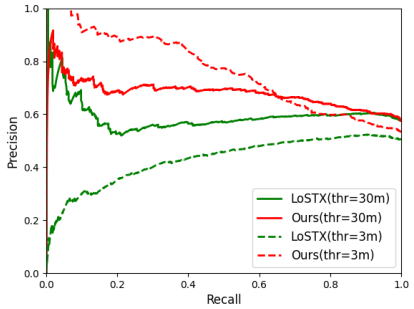


Fig. 8: The comparison between our method and LoSTX under the condition of visual offset.

TABLE III: Comparison of average run-time(ms)

Method	Tracking	Loop Detection	Loop Correction
ORB-SLAM3(Mono)	24.6	8.13	1097.7
Ours	32.9	15.6	1193.4

Taking visual offset into account, the results reveal an improvement in LoSTX performance, while our method shows a decline in performance. The decline in our method's performance is attributed to the widening distance, which may lead to changes in semantic layout in some environments.

Nonetheless, our method still maintains a certain level of performance due to the presence of similar semantic layouts along some nearby driving paths, which is the reason why we employ spatio-temporal verification to select the optimal loop closure frame. The characteristic of LoSTX may enable it to identify similar scenes earlier, but it is not conducive to integration into visual SLAM systems. If loop closures are found at a large distance, distant points tend to be used for loop correction. In visual SLAM, distant points are sparser and less accurate in position. We attempted to integrate LoSTX into the SLAM system, but it consistently resulted in system failed upon loop closure discovery.

D. Time Evaluation

We present our computational cost of Tracking (semantic segmentation, original track, descriptor generation, semantic-based scale correction), Loop Detection(original detection and verification, descriptor similarity detection and spatial-temporal verification) and Loop Correction(original forward loop correction, semantic ICP, original optimization) in Table III. Despite the minor increased time for each component, our system still maintains real-time performance due to the utilization of multithreading and keyframes.

V. CONCLUSION

We have presented a monocular visual SLAM approach that leverages semantic information for scale correction and reverse loop closure utilization. Experiments on the KITTI odometry dataset and the Oxford RobotCar dataset

are conducted to evaluate the accuracy of trajectory and loop closure detection, confirming the effectiveness of the proposed method. We aim to address the shortcomings of semantic visual SLAM through this intriguing work and propel its advancement. We also find our semantic visual point cloud descriptor relies on scale stability during test. In the future, we will endeavor to extend our method to SLAM systems with stable scale, such as those employing stereo-camera, multi-camera or visual-inertial configurations.

REFERENCES

[1] 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.

[2] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[3] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.

[4] M. Ferrera, A. Eudes, J. Moras, M. Sanfourche, and G. Le Besnerais, "Ov ^{2} slam: A fully online and versatile visual slam for real-time applications," *IEEE robotics and automation letters*, vol. 6, no. 2, pp. 1399–1406, 2021.

[5] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[6] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[7] C. Forster, M. Pizzoli, and D. Scaramuzza, "Svo: Fast semi-direct monocular visual odometry," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 15–22.

[8] X. Gao, R. Wang, N. Demmel, and D. Cremers, "Ldso: Direct sparse odometry with loop closure," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2198–2204.

[9] N. Merrill and G. Huang, "Lightweight unsupervised deep loop closure," in *Proc. of Robotics: Science and Systems (RSS)*, Pittsburgh, PA, June 26-30, 2018.

[10] Nathaniel Merrill and Guoquan Huang, "CALC2.0: Combining appearance, semantic and geometric information for robust and efficient visual loop closure," in *2019 International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 2019.

[11] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[12] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 141–14 152.

[13] S. Garg and M. Milford, "Seqnet: Learning descriptors for sequence-based hierarchical place recognition," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4305–4312, 2021.

[14] G. Berton, G. Trivigno, B. Caputo, and C. Masone, "Eigenplaces: Training viewpoint robust models for visual place recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 080–11 090.

[15] J. Li, K. Koreitem, D. Meger, and G. Dudek, "View-invariant loop closure with oriented semantic landmarks," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 7943–7949.

[16] B. Zhou, Y. Meng, and F. Kai, "Object-based loop closure with directional histogram descriptor," in *2022 IEEE 18th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2022, pp. 1346–1351.

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

[18] S. Song, M. Chandraker, and C. C. Guest, "High accuracy monocular sfm and scale correction for autonomous driving," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 730–743, 2015.

[19] D. Zhou, Y. Dai, and H. Li, "Ground-plane-based absolute scale estimation for monocular visual odometry," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 2, pp. 791–802, 2019.

[20] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[21] J. Wang, M. Rünz, and L. Agapito, "Dsp-slam: Object oriented slam with deep shape priors," in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1362–1371.

[22] S. Liang, Y. Zhang, R. Tian, D. Zhu, L. Yang, and Z. Cao, "Semloc: Accurate and robust visual localization with semantic and structural constraints from prior maps," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4135–4141.

[23] M. Herb, M. Lemberger, M. M. Schmitt, A. Kurz, T. Weiherer, N. Navab, and F. Tombari, "Semantic image alignment for vehicle localization," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1124–1131.

[24] 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.

[25] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[26] Y. Wu, Y. Zhang, D. Zhu, Y. Feng, S. Coleman, and D. Kerr, "Eao-slam: Monocular semi-dense object slam based on ensemble data association," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 4966–4973.

[27] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[28] Sivic and Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings ninth IEEE international conference on computer vision*. IEEE, 2003, pp. 1470–1477.

[29] E. Garcia-Fidalgo and A. Ortiz, "ibow-lcd: An appearance-based loop-closure detection approach using incremental bags of binary words," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3051–3057, 2018.

[30] S. Garg, N. Suenderhauf, and M. Milford, "Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics," *Proceedings of Robotics: Science and Systems XIV*, 2018.

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

[32] J. Mo, M. J. Islam, and J. Sattar, "Fast direct stereo visual slam," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 778–785, 2021.

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

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

[35] Li, Lin and Kong, Xin and Zhao, Xiangrui and Huang, Tianxin and Li, Wanlong and Wen, Feng and Zhang, Hongbo and Liu, Yong, "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.

[36] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2636–2645.

[37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881–2890.

[38] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.