2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 1-5, 2023. Detroit, USA

SELVO: A Semantic-Enhanced Lidar-Visual Odometry

Kun Jiang[1,†], Shuang Gao[2,†], Xudong Zhang[3],
Jijunnan Li[3], Yandong Guo[4], Shijie Liu[1], Chunlai Li[1,5,‡], Jianyu Wang[1,5,‡]

***Abstract*— In the face of complex external environment, single sensor information can no longer meet the accuracy requirements of low-drift SLAM. In this paper, we focus on the fusion scheme of cameras and lidar, and explore the gain of semantic information to SLAM system. A Semantic-Enhanced Lidar-Visual Odometry (SELVO) is proposed to achieve pose estimation with high accuracy and robustness by applying semantics and utilizing strategies of initialization and sensor fusion. In loop closure detection thread, we propose a novel place recognition method based on semantic information to maintain the global consistency of the map. In the back-end, we design a joint optimization framework including visual odometry, lidar odometry and loop closure detection, and innovatively propose to recognize degraded scenes with semantic information. We have conducted a large number of experiments on KITTI [1] and KITTI-360 [2] dataset, and the results show that our system can achieve the high accuracy and competitive performance in comparison with state-of-the-art methods.**

I. INTRODUCTION

In the field of robot navigation, Simultaneous Localization and Mapping (SLAM) is a fundamental capability to obtain information about the surrounding environment and the 6 degrees of freedom (6-DoF) pose of the sensor. In the past few years, single-sensor-based SLAM methods, including lidar-based and vision-based, have been fully developed and adequately applied in areas such as Autonomous Driving, Robot Navigation, Augmented Reality (AR), etc.

The vision-based methods which use visual features on image sequence to regress the camera poses and reconstruct the feature point cloud maps, have developed relatively well in recent decades. In addition to a large number of traditional features extraction and matching methods, some learning-based methods have been proposed [3], [4], [5], [6]. However, considering the efficiency and generalization, learning-based features are more widely used for visual localization rather than visual tracking. None of these visual features can fundamentally solve the problems of weak textures, illumination changes, and viewing angle changes.

The lidar-based methods achieve mapping and localization by registering the captured point clouds. Some methods rely on structural features to achieve high accuracy in scenes with rich structural information [7], [8], [9]. However, in some degraded scenes, such as tunnels, promenades and open fields, these methods often fail because matching ambiguity. To avoid such circumstances, some researches utilize the coupling of Inertial Measurement Unit (IMU) data and point cloud features to achieve more stable effects [10], [11], [12]. There are also some methods to improve the odometry accuracy by fusing Lidar Odometry (LO) and Visual Odometry (VO) to supplement visual information when structural features are degraded [13], [14]. How to adjust the reliability of VO and LO measurements in different scenarios is still a sustainable research topic.

The semantics as additional information of point cloud besides the structured features, has an effective application in point cloud registration and loop closure detection. In this work, we propose a fusion odometry method of VO and LO with semantics. In detail, in the traditional point cloud feature matching stage, we use semantic information to filter out error matches to improve the relative pose accuracy between two adjacent lidar frames. Then we propose a novel Semantic Scan Context (SSC) calculation method for loop closure detection of lidar odometry. We directly employ ORB-SLAM2 [15], a mature visual SLAM system, to provide the relative pose between two adjacent camera frames. Finally, we design a fusion framework, including VO, LO and loop closure detection, to obtain a more accurate odometry.

In this paper, our main contributions can be summarized as following parts:

1. We design a Semantic-Enhanced Lidar-Visual Odometry fusion framework named SELVO, which includes VO, LO, and loop closure factors. We innovatively use semantic information to judge the degradation degree of the scene, so as to dynamically adjust the weights of different factors in the fusion framework.

2. We propose a novel SSC calculation method to avoid the failure of loop closure detection caused by excessive odometry drift, and narrow the range of candidate loop closure frames to reduce the registration computational burden.

3. We evaluate the final odometry accuracy on different scenes, including high way and city road. We also compare our proposed method with some state-of-the-art ones to prove the performance.

II. RELATED WORK

In this section, we introduce some recent representative work in lidar odometry, semantic lidar mapping and lidar-visaul odometry. All of them are closely related to our work

† These authors contributed equally to this work.
[1] K. Jiang, S. Liu, C. Li and J. Wang are with School of Physics and Optoelectronic Engineering, Hangzhou Institute for Advanced Study, University of Chinese Academy of Sciences, Hangzhou, China.
[2] S. Gao is with Li Auto Inc, Shanghai, China.
[3] X. Zhang and J. Li are with OPPO Research Institute, Shanghai, China.
[4] Y. Guo is with AI[2] Robotics, Shenzhen, China.
[5] C. Li and J. Wang are also with Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai, China.
‡ Both are the corresponding authors, e-mail: lichunlai@mail.sitp.ac.cn, jywang@mail.sitp.ac.cn.

2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) | 978-1-6654-9190-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/IROS55552.2023.10341419

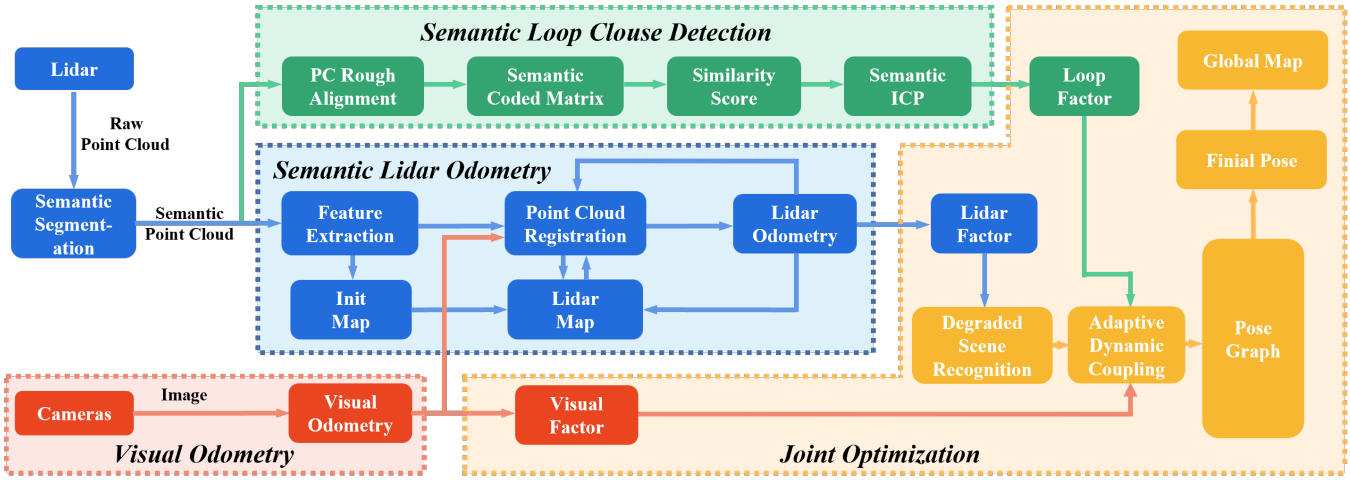


Fig. 1. **System Pipeline.** The data from one 3d Lidar and a pair of stereo cameras is passed into the system, with parallel execution of lidar odometry, visual odometry and loop closure detection threads. Then we jointly optimize the lidar factor, visual factor and loop factor to obtain a high-precision pose, finally update and register the semantic map with global consistency in real time.

in this paper.

A. Lidar Odometry

As one of the most classical lidar odometry framework, LOAM [7] classifies the point cloud features into two categories: edge points and planar points, for features matching and pose estimation. Based on this method, many derivative methods have emerged. Lego-LOAM [8] adds ground-optimized and uses a two-step Levenberg-Marquardt optimization method to to solve different components of the 6DoF transformation across consecutive scans. FLOAM [9] adopt a non-iterative two-stage distortion compensation method to reduce the computational cost, and achieve a fast lidar odometry and mapping. CT-ICP [16] combines continuity in the scan matching and discontinuity between scans, increasing the robustness of high frequency motions from the discontinuity. In addition, there are some IMU-assisted lidar odometry, such as fast-lio [10], [11] uses a tightly-coupled iterated extended Kalman filter to obtain the lidar scan poses, and LIOM [12] estimates the stable lidar poses by jointly minimizing the cost derived from lidar and IMU measurements.

B. Semamtic Lidar Mapping

Semantics are usually used for point cloud registration and loop closure detection in lidar Odometry. [17] propose a semantic ICP algorithm which uses joint semantic and geometric probabilities for finding the associations in the GICP-SE3 algorithm. SuMa++ [18] also uses semantic information provided by RangeNet++ [19] to filter dynamic objects and adds semantic constraints to the optimization problems as weights for scan registration. [20] designs parameterized semantic features (PSFs) and matching method. Inspired by the Intensity Scan Context (ISC), [21] proposes a Semantic Scan Context (SSC) computing method to recognize similar place for loop detection. Based on SSC, [22] proposes a semantic-aided lidar SLAM with loop closure named SA-LOAM to improve the localization accuracy.

C. Lidar-Visual Odometry

LIMO [23] utilizes lidar measurements to extract depth information for camera features tracking and pose estimation. Similar to our proposed method, LIOM uses LiDAR and camera as input, without IMU. The difference is that we use camera information to improve the lidar odometry accuracy, while it uses LiDAR information to improve the visual odometry accuracy. [13] achieves a tight coupling of visual and lidar measurements and removes the errors in assigning the depths of lidar to non-corresponding visual features. LVI-SAM [14] proposes a tightly coupled framework for lidar-visual-inertial odometry. The VO leverages LO estimation to facilitate initialization and the LO utilezes VO estimation for initial guesses to support scan-matching. [24] combines LIO subsystem for building geometric structure and VIO subsystem for rendering the map's texture to reconstruct dense and RGB-colored maps of the surrounding environment in real-time.

III. System Overview

The SELVO system pipeline we proposed is shown in Fig. 1. It consists of four parts: visual odometry, semantic lidar odometry, semantic loop closure detection and joint optimization.

Visual Odometry. Here, we refer directly to the ORB-SLAM2 [15] to obtain relative pose estimation of camera in Visual Odometry module.

Semantic Lidar Odometry. In order to improve the matching accuracy of the corresponding points in ***scan to map*** process, we introduce semantic constraints to filter error matches. In addition, at the beginning of the system, the relative pose obtained by visual odometry helps to compute the initial iterative value of lidar odometry, which can reduce

the time to reach convergence for lidar odometry, especially in high-speed scenarios.

Semantic Loop Closure Detection. We design a semantic topological point cloud to roughly align the loop closure candidate frames with the current frame, and screen out the best loop closure frame through the calculation of the semantic encoding matrix and similarity score. Then, semantic ICP is implemented for geometric verification.

Pose Graph Joint Optimization. Pose graph is constructed to jointly optimize the above three types of factors. For the purpose of the best optimization effect, We define a scene structured rate according to semantic distribution to determine degraded scenarios, and furthermore, design a cost function with adaptive weights for pose graph optimization.

The following modules constitute our main innovations and contributions. Semantic LO, Semantic-enhanced Loop Closure Detection and Fusion Optimization framework are introduced in Sec.IV, Sec.V and Sec.VI respectively. Sec.VII describes the experiments in detail, and Sec.VIII is the conclusion of this work.

IV. Semantic Lidar Odometry

Feature-based lidar odometry methods [7], [9] inevitably cause mismatches in the process of point cloud registration. We introduce semantic constraints to improve the accuracy of pose estimation. On top of that, the results of visual odometry are utilized as initial guess for lidar odometry considering the robustness of the system.

A. Semantic Constraints

As depicted in Fig. 2, the different colors of the points indicate the different semantic labels. Fig. 2(a) and (b) reveal the process of solving the distances of point-to-edge and point-to-surface respectively. Facing different classes of points, the lines and planes formed by the N closest submap points which have the same label with the current scan point are given priority. The following formula embodies our idea:

$$R*, t* = \arg\min_{R,t} \left\{ \begin{array}{l} \sum\limits_{e} w^l d_{p2e}(E_k^i, E_m^j, \overrightarrow{P_m^e}, R, t) + \\ \sum\limits_{p} w^l d_{p2s}(S_k^i, S_m^j, \overrightarrow{N_m^s}, R, t) \end{array} \right\}, \tag{1}$$

where d_{p2e} and d_{p2s} denotes the distance of point-to-edge and point-to-surface respectively, E_k^i and S_k^i denotes the i edge point and surface point in the k scan. E_m^j corresponds to the closest point of E_k^i in edge submap, Similarly, S_m^j is the corresponding point of S_k^i in surface submap. $\overrightarrow{P_m^e}$ denotes the unit vector in the principal direction of the line consisted of the N closest corresponding points in edge submap. $\overrightarrow{N_m^s}$ means the normal vector of common surface fitted by the N closest corresponding points in surface submap. Usually N is set to 5. And w^l is defined as:

$$w^l = \left\{ \begin{array}{ll} 1 & if\ l_k^i = l_m^j \\ 0 & if\ l_k^i \neq l_m^j \end{array} \right., \tag{2}$$

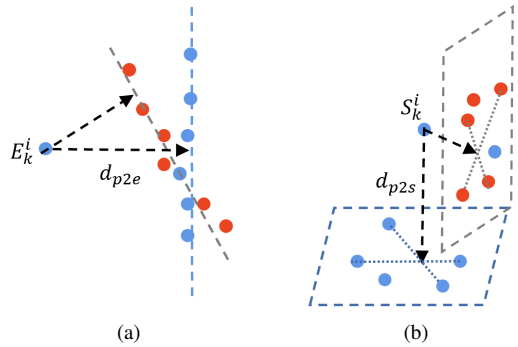


Fig. 2. **Finding the correspondence with semantic constraints.** Red and blue indicate points with different semantic labels. (a) and (b) show that the edge point E_k^i and surface point S_k^i find their corresponding line and plane, respectively. Both will give priority to structures that are consistent with their own labels. In (a), the line formed by the nearest red point to E_k^i is not be chosen, but the line fitted by the next closest points with the same label. The corresponding surface in (b) also follow this rule.

l_k^i denotes the semantic label of the i point in the k scan, l_m^j denotes the semantic label of point closest to l_k^i in submap. So, matches with different semantic values will be rounded off. And two types of distance can be expressed as Eq. 3 and Eq. 4:

$$d_{p2e}(\cdot) = \overrightarrow{P_m^e} \times \left(R \cdot E_k^i + t - E_m^j\right), \tag{3}$$

$$d_{p2s}(\cdot) = \overrightarrow{N_m^s} \cdot \left(R \cdot S_k^i + t - S_m^j\right), \tag{4}$$

B. Initial Guess

In ***scan to map*** process, the importance of the initial value of the iteration is self-evident, as it determines the accuracy of the final relative pose and the speed to reach convergence. In a pure lidar-based scheme, the initial pose guess of the k frame is usually set as Eq. 5:

$$T_k^{initial} = T_{k-1} T_k^{k-1} \approx T_{k-1} T_{k-1}^{k-2}, \tag{5}$$

where the relative pose T_k^{k-1} between the k frame and the $(k-1)$ frame is considered to be approximately equal to T_{k-1}^{k-2} since relative motion of little change in a short time. This suppose leads to a situation: there is no prior relative pose at the beginning of the system, thus the relative pose SE(3) of lidar is basically iterated from an identity matrix.

However, in high-speed scenarios such as highways, the initial motion between adjacent frames of lidar is so intense that it often requires a huge number of iterations to reach convergence or fall into local convergence, which brings a huge challenge to the subsequent performance. In our system, the initial guess of LO is obtained from VO, especially for the first frame, and it can effectively improve the robustness of the system in high-speed scenes.

V. Semantic-enhanced Loop Closure Detection

Loop closure detection plays a crucial role in building a large-scale map with global consistency. Inspired by [25], [21], we propose a novel semantic encoding matrix representation method, and the similarity score is performed to select the best loop closure frame. Before that, the angle and position deviation between the current frame and the candidate frames are corrected by point cloud rough alignment. Finally, Semantic ICP is implemented for geometric verification.

A. Point Cloud Roughly Aligned

Horizontal angle differences and translations between point cloud pairs have a significant influence on the accuracy of recognition. Point cloud pairs need to be pre-aligned in angle and translation before calculating similarity [21]. For a rough pre-alignment, calculating all points are not necessary. Therefore, only a few representative points are selected for the alignment process to reduce computation.

As shown in Fig. 3, in order to take into account the diversity of semantics, the points with the same semantics are clustered into the same class and a semantic topological graph is constructed by calculating the centroid of each cluster in the sector area. Based on this, we rotate the semantic topological point cloud in the horizontal direction and perform a 2D ICP to roughly align current point cloud P_1 and candidate point cloud P_2. The result of the rough alignment is expressed approximately as Eq. 6:

$$\mathrm{T}\left(P_2\right) \approx Trans \cdot \mathrm{T}\left(P_1\right), \tag{6}$$

where $\mathrm{T}\left(\cdot\right)$ denotes the mapping of original point cloud to topological point cloud, and $Trans$ is the rough transformation matrix. Therefore, the transformed point cloud $P_1^{'}$ of the current frame, i.e. $Trans \cdot P_1$, can be considered to be roughly aligned with the candidate frame P_2.

B. Semantic Encoding Matrix and Similarity Score

The idea of dividing point cloud into regular segments (bins) in our method has been explored in Scan Context [26], which only develops, however, spatial information such as height. Our semantic encoding matrix D is interpreted as a global descriptor, which can be obtained according to Fig. 4. Firstly, the point cloud is divided evenly into N_s sectors and N_r rings, resulting in $N_s * N_r$ unit areas. For a unit area B_{ij}, the encoding process is expressed as Eq. 7:

$$D\left(i,j\right) = E\left(B_{ij}\right) = \arg\max\left(Hist(B_{ij})\right), \tag{7}$$

where $i \in [1, N_r]$, $j \in [1, N_s]$, $E\left(\cdot\right)$ denotes a encoding function that we perform histogram statistics $Hist(\cdot)$ on semantic in unit area B_{ij} and pick out representative semantics which make up the largest share. Then the label of this representative semantic is assigned to $D\left(i,j\right)$.

Naturally, a pair of semantic encoding matrices $D_1^{'}$ and D_2 which have been roughly aligned can be obtained by means of the above. Note that the current point cloud P_1 first needs to be transformed to $P_1^{'}$ according to Eq. 6. We define the similarity score s between $D_1^{'}$ and D_2 as Eq. 8:

$$s = \frac{\sum_{i=1}^{N_r}\sum_{j=1}^{N_s}\Gamma\left(D_1^{'}(i,j), D_2(i,j)\right)}{N_r * N_s}, \tag{8}$$

where $\Gamma\left(\cdot\right)$ is used to determine if two elements are similar or not, it is defined by Eq. 9:

$$\Gamma\left(x,y\right) = \begin{cases} 1 & if\ |x-y| = 0 \\ 0 & otherwise \end{cases}, \tag{9}$$

C. Semantic ICP

It's necessary to perform geometric verification to rule out possible false loop closure frames. Here, we register two frames by semantic ICP, Similarly to our semantic LO as Eq. 10.

$$R^*, t^* = \underset{R,t}{\arg\min}(p_i - (Rp_i^{'} + t)), \tag{10}$$

p_i denotes the i point in current frame. $p_i^{'}$ is the nearest point to p_i, which also has the same semantic as p_i, the matches with different semantics is rejected. When loss of semantic ICP in Eq. 10 is too large, we then conclude that these two frames are not consistent, and the loop closure one is to be considered a failure.

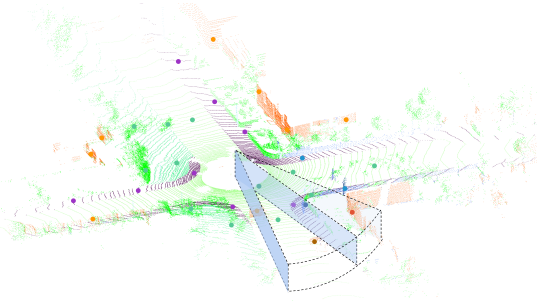


Fig. 3. **Semantic topological point cloud.** The point cloud is divided equally into sectors, and for each sector, the centroid of point cluster with same semantics is considered to be the topological point.

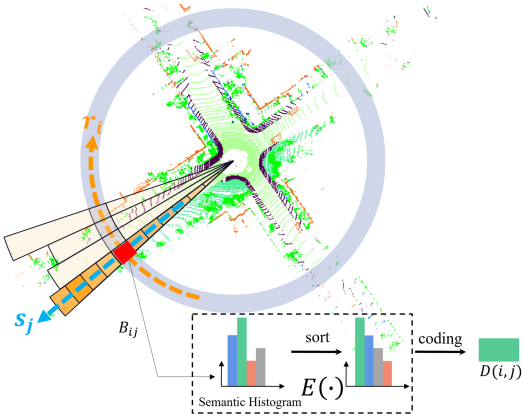
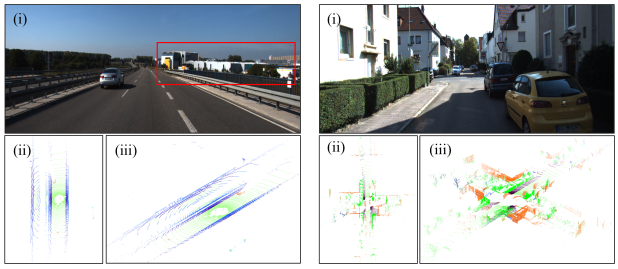


Fig. 4. **Semantic coding in scan context.** In the top view, the point cloud is divided into bins by uniform sector and ring. B_{ij} represents the intersection of r_i and s_j, and it is performed the semantic histogram statistics and sorted. Then the semantic with the largest proportion encodes this bin.



(a) Seq. 01 #490 (b) Seq. 00 #1345

Fig. 5. **Degradation and non degradation case in KITTI.** (i) is for camera view. (ii) and (iii) are lidar point cloud from top and side views. (a) is a degradation case in Seq. 01, where the red box shows a larger field of view than lidar. (b) is a non degradation case in Seq. 00.

VI. Fusion Optimization

In this section, an adaptive dynamic coupling lidar-visual optimization scheme is designed by judging the structural scenes are degraded or not, so that the two sensors can contribute the best accuracy in the scenes that satisfy their respective working conditions.

A. Lidar Degraded Scene Recognition Based on Semantic Information

The lidar degraded scene, such as KITTI Seq. 01 #490, is shown in Fig. 5 (a), the available structured information is only roads and repeated railings. Instead, the scene in KITTI Seq. 00 #1345 includes a lager number of structured point as Fig. 5 (b). In degraded scene, both feature extraction and matching are a challenge for lidar odometry. Fortunately, the scene understanding function of semantic point cloud enables us to discriminate these lidar degraded scenes.

Accordingly, we propose a semantic-based approach for degraded scene recognition. In general, degraded scenes often have fewer building points and a larger number of ground points. We can quantitatively measure the degree of scene degradation by calculating the structured rate σ as the following formula:

$$\sigma = e^{-(a-b)}, \tag{11}$$

where a and b are the percentage of ground points number and building points number respectively. And both calculations are processed by excluding outlier points. It stands to reason that when a is smaller and b is larger, the structured rate σ will be larger. To sum up, when the scene structured rate is lower than a threshold, we consider that this scene to be a degraded one and the lidar only has a low degree of confidence in the current scene. Here, the above parameters is given in Table. I.

B. Pose Graph Construction

The pose graph is constructed by introducing different sensor factors and loop closure factor, as shown in ***Joint Optimization*** module in Fig. 1. In the field of robot state estimation, this type of problem can be described as maximum posterior probability estimation, which is equivalent to solving a nonlinear least squares problem as Eq. 12.

TABLE I: Structural information in scenes of Fig. 5.

Case	Ground (a/%)	Building (b/%)	Structured rate (σ/%)
Seq. 00 #1345	29	23	94
Seq. 01 #490	76	0	47

$$\min \sum_{i \in \{ L,V,P \}} e_i^T \Sigma^{-1} e_i, \tag{12}$$

where L, V, P denotes all edges in pose graph including lidar factor, visual factor and loop closure factor. Both lidar factor and visual factor in pose graph can be naturally expressed as:

$$\Delta\xi_{i-1,i} = \ln \left(T_{i-1}^{-1} T_i\right)^{\vee}, \tag{13}$$

where $\Delta\xi_{i-1,i} \in \mathbb{R}^6$, it indicates the relative pose between adjacent frames $(i-1)$ and i.

For the loop closure factor here, we obtain the relative pose between the loop closure frame and the current frame in Section. V-C.

C. Adaptive Dynamic Coupling

The information matrix Σ in Eq. 12 reflects the confidence of sensors and loop closure constraints. It can be described in a more acceptable way just like the weight coefficient in Eq. 14.

$$\min \left\{ \begin{array}{c} \sum \phi_i * \left\| R_L \left(T_{i-1}^{'}, T_i^{'}, \Delta\xi_{i-1,i}^{L} \right) \right\|^2 + \\ \sum \varphi * \left\| R_V \left(T_{i-1}^{'}, T_i^{'}, \Delta\xi_{i-1,i}^{V} \right) \right\|^2 + \\ \sum \psi * \left\| R_P \left(T_i^{'}, T_j^{'}, \Delta\xi_{i,j}^{L} \right) \right\|^2 \end{array} \right\}, \tag{14}$$

$R_L(\cdot)$, $R_V(\cdot)$ and $R_P(\cdot)$ denote the residuals of the lidar, camera and loop closure edges respectively. $T^{'}$ means the lidar pose after fusion optimization. $\Delta\xi_{i-1,i}^{V}$ is camera's relative pose between adjacent frames, and $\Delta\xi_{i,j}^{L}$ is lidar's relative pose between the i and j frame, where j is the index of loop closure frame. ϕ_i, φ and ψ denote the weights of the three residuals.

Based on the degraded scene recognition in Section. VI-A, we design a multi-sensor adaptive dynamic coupling optimization scheme. Specifically, The sensors adaptively obtain optimization weights according to degraded scene recognition and structured rate in Eq. 15:

$$\left\{ \begin{array}{l} \phi_i = \left\{ \begin{array}{ll} \alpha * e^{-(a_i - b_i)} & if\ a_i > thre1 \cap b_i < thre2 \\ 1 & otherwise \end{array} \right. \\ \varphi = 1 \\ \psi = 1 \end{array} \right., \tag{15}$$

In the degraded scenes, we assign the weights to the lidar factor, which is positively correlated with the structured rate

TABLE II: Performance of proposed approach on different conditions.

Conditions			KITTI00	KITTI01	KITTI02	KITTI03	KITTI04	KITTI05	KITTI06	KITTI07	KITTI08	KITTI09	KITTI10
$\mathcal{S}$	$\mathcal{L}$	$\mathcal{C}$	$Avg\ T_{rel}$(%)/$Avg\ R_{rel}$(deg/100m)										
×	×	×	0.68/0.26	1.78/0.45	1.00/0.35	1.00/0.51	0.61/0.35	0.49/0.23	0.53/0.25	**0.40/0.24**	0.95/0.33	0.76/0.30	0.98/0.39
✓	×	×	0.62/0.26	1.68/0.43	0.75/0.28	0.85/0.49	0.58/0.27	0.42/0.24	0.50/0.24	0.54/0.31	0.85/0.28	**0.62/0.29**	0.81/0.40
✓	✓	×	**0.59/0.19**	1.68/0.43	0.74/0.26	0.85/0.49	0.58/0.27	**0.32/0.15**	0.52/0.28	0.45/0.25	**0.85/0.23**	0.74/0.26	0.81/0.40
✓	✓	✓	0.64/0.21	**0.71/0.20**	**0.71/0.23**	**0.85/0.30**	**0.49/0.13**	0.36/0.21	**0.38/0.18**	0.51/0.30	0.88/0.23	0.78/0.25	**0.61/0.24**

[1] $\mathcal{S}$, $\mathcal{L}$, $\mathcal{C}$ represent: semantic constraints, semantic loop closure detection and lidar-camera fusion. ✓and × mean with and not with conditions.
[2] $Avg\ T_{rel}$ and $Avg\ R_{rel}$ (the smaller the better) denote Average Translational Relative trajectory errors and Average Rotational Relative trajectory errors.

σ, and α is a constant that denotes the degradation factor. $thre1$ and $thre2$ are the threshold of a and b, which decide whether existing a degraded scene. Otherwise, the camera factor and the lidar factor contribute equally to the optimized pose.

VII. Experiments

A. Experiment Setup

We train and test our algorithm on KITTI dataset and KITTI-360 dataset, respectively. Both are equipped with a station wagon with a 64-line velodyne lidar, stereo cameras, and the IMU/GPS localization system. KITTI dataset includes 21 sequences (Seq. 00-10 provide groundtruth pose for training, Seq. 11-21 are used for evaluate accuracy in benchmark), covering urban, suburban, high-speed, wild and other scenes. KITTI-360 dataset recorded several suburbs of Karlsruhe, Germany, corresponding to over 320k images and 100k laser scans in a driving distance of 73.7km.

Our algorithm refers to the open source lidar odometry method FLOAM and ORB-SLAM2 for stereo cameras is chosen as visual odometry module. In the preprocessing stage of point clouds, we choose the current state-of-the-art semantic segmentation network RangeNet++ to obtain semantic information. Our computing platform includes Intel i7-8700 CPU @ 3.20GHz and Intel UHD Graphics 630 GPU. The objects of our experiments include LOAM, FLOAM, ISC-LOAM, ORB-SLAM2, etc. These open source framework all run on the same platform.

B. Ablation Studies

We designed several rigorous ablation experiments to demonstrate the effectiveness of our approach. Using FLOAM method as the baseline, we sequentially added semantic constraints, semantic loop closure detection and joint optimization. As shown in Table. II, we conducted our ablation experiments on KITTI dataset Seq. 00-10 which cover a variety of rich scenes. $Avg\ T_{rel}$ and $Avg\ R_{rel}$ are as the metric of algorithm.

*1) **Performance on Semantic Constraints**:* The first row of Table. II shows the accuracy of the original FLOAM, and the second row illustrates the performance change due to the addition of semantic constraints. Our method shows varying degrees of performance improvement in all scenes, and it also proves that semantic constraints can improve odometry accuracy and have a positive effect on reducing mismatching in ***scan to map*** process. The only exception is reflected in Seq. 07, where the baseline exhibits higher accuracy. Our analysis is that incorrect semantic segmentation results in an increase in the number of mismatches on Seq. 07, which causes a reduction in odometer accuracy.

TABLE III: Absolute Pose Error on KITTI Odometry with loop closure.

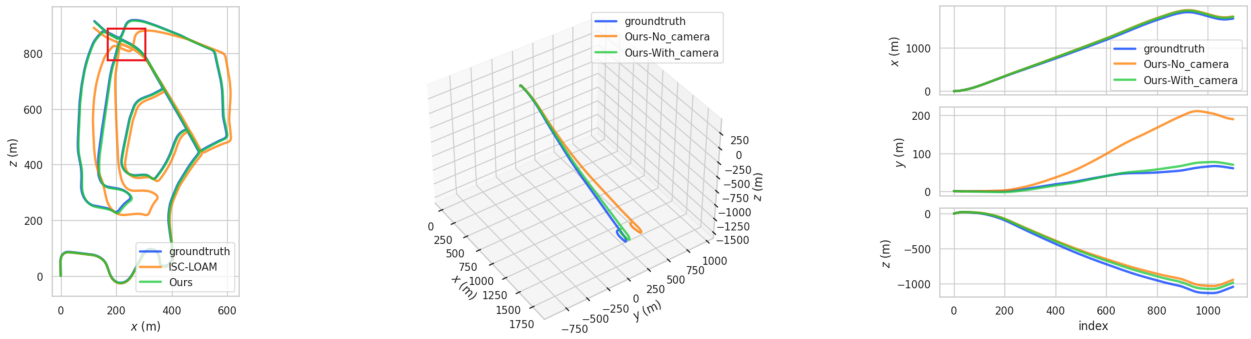
Sequence	**Ours-Odom** (m)	ISC-LOAM (m)	SA-LOAM (m)	**Ours-Loop** (m)
00	4.89	1.13	**0.99**	1.06
02	7.33	35.04	9.24	**4.03**
05	2.57	0.55	0.75	**0.44**
06	0.68	0.70	0.64	**0.63**
07	0.97	**0.36**	0.36	0.38
08	3.44	4.10	3.24	**2.75**
09	1.21	1.38	1.20	**1.13**
Average	3.01	6.18	2.34	**1.49**

*2) **Performance on Semantic Loop Closure Detection**:* The gain to the system from loop closure detection is reflected in the global accuracy. Here, we use the EVO [27] toolkit to evaluate the performance of loop closure detection by computing the Absolute Pose Error (APE). In Table. III, we selected all sequences with loop closure in KITTI dataset, including challenging cases such as rich loops and large-scale loops. Statistics show that compared to **Ours-Odom**, all APE of **Ours-Loop** have great improvement. Ours in global accuracy also has advantages over other methods with loop closure detection [28], [22], especially Seq. 02.

Fig. 6 (a) shows our method correctly recognizes the loop closure frame and the accumulated error is compensated well in red box, which is due to the correct judgment of the loop closure frame by our semantic loop closure detection and good semantic ICP registration results.

*3) **Performence on Joint Optimization**:* In our method, the addition of the camera brings two changes: (1). as the initial guess of LO (2). VO factor is added to the pose graph for joint optimization. These changes make an outstanding contribution to the generalizability of our algorithm. As shown in Fig. 7, the case regarding VO as the initial guess performs much better than the case without VO on the numbers of iteration, where Seq. 01, 12, 17 are high-speed scenes.

In the last row of Table. II, it is worth noting that the Seq. 01 has a good improvement in accuracy compared to the first three. The ground truth, ours-nocamera, and ours-withcamera trajectories are shown in Fig. 6 (b) and (c), along with a comparison of the values on coordinate axis. There are a lot of lidar degraded scenes in Seq. 01, which is why the lidar has a huge bias in the y-axis. At this point, we trust the cameras more since the pixels in red box of Fig. 5 ensure that the visual odometry has enough features for a relatively



(a) trajectories of KITTI Seq. 02 (b) trajectories of KITTI Seq. 01 (c) xyz view of KITTI Seq. 01

Fig. 6. Comparison of trajectories.

TABLE IV: Accuracy Evaluation and Comparison on KITTI Odometry dataset.

Approach	00* Urban	01 Highway	02* Urban	03 Country	04 Country	05* Country	06* Urban	07* Urban	08* Urban	09* Urban	10 Country	00-10 Mean
LOAM [29]	0.78/-	1.43/-	0.92/-	0.86/-	0.71/-	0.57/-	0.65/-	0.63/-	1.12/-	0.77/-	0.79/-	0.84/-
PSF-LO [20]	**0.64**/-	1.32/-	0.87/-	0.75/-	0.66/-	0.45/-	0.47/-	0.46/-	0.94/-	0.56/-	**0.54**/-	0.74/-
SuMa++ [18]	**0.64**/0.22	1.60/0.46	1.00/0.37	**0.67**/0.46	0.37/0.26	0.40/0.20	0.46/0.21	**0.34/0.19**	1.10/0.35	**0.47/0.23**	0.66/0.28	0.70/0.29
ORB-SLAM2 [15]	1.22/0.23	1.08/0.21	1.40/0.24	2.81/**0.16**	1.20/**0.11**	0.69/**0.16**	0.97/0.23	0.69/0.26	1.16/0.28	1.15/0.26	0.97/0.34	1.21/**0.23**
ISC-LOAM [25]	0.68/0.31	2.29/0.55	4.04/1.23	1.25/0.55	1.20/0.36	0.51/0.26	0.60/0.34	0.55/0.39	1.16/0.41	0.73/0.30	1.71/0.49	1.34/0.47
Ours	**0.64/0.21**	**0.71/0.20**	**0.71/0.23**	0.85/0.30	**0.49**/0.13	**0.36**/0.21	**0.38/0.18**	0.51/0.30	**0.88/0.23**	0.78/0.25	0.61/**0.24**	**0.63/0.23**

[1] All errors are represented as $Avg\ T_{rel}(\%)/Avg\ R_{rel}$(deg/100m).
[2] *: with loop closure, -: unknown.
[3] The result of LOAM, PSF-LO, SuMa++ are obtained from their raw paper [29], [20], [18]. ORB-SLAM2 and ISC-LOAM are running on the same hardware platform as our approach.

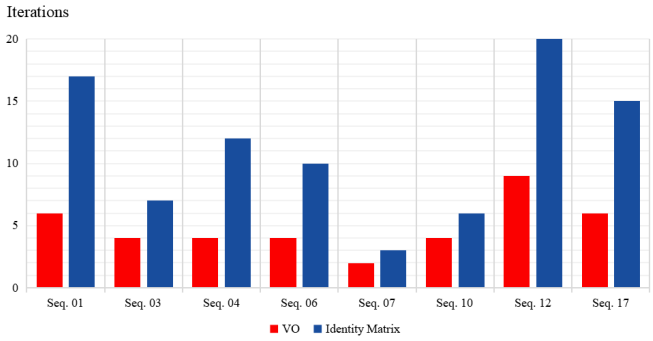


Fig. 7. **Number comparison of iterations between case1: VO as initial guess and case2: Identity Matrix as initial guess.** Note that the conditions for termination of iterations include: (a). iterating for 30 times and (b). translation error is less than 0.01m, otherwise reaching convergence.

accurate pose estimation.

Sequences such as 09 show slight performance degradation due to the fact that visual sensors also have corner case like degradation scenes in some time periods. However, we chose to give equal confidence to both in the times when there are no lidar degraded scenes. Since our current work focuses only on the lidar degraded scene correction, reduction of accuracy is within the error tolerance and do not affect the generalization performance of our algorithm. We believe that better accuracy can be achieved by adjusting the confidence levels of both for general scenes.

TABLE V: Accuracy Evaluation and Comparison on KITTI-360 Semantic SLAM trajectory evaluation.

Approach	test0	test1	test2	test3	Average
FLOAM [9]	0.92	1.17	6.11	8.10	4.08
ISC-LOAM [25]	0.35	1.17	4.76	**1.10**	1.85
ORB-SLAM2 [15]	1.53	2.22	**2.12**	1.79	1.92
SUMA++ [18]	2.27	2.87	4.62	2.77	3.13
Ours-Odom	0.58	1.95	3.51	6.81	3.21
Ours-Loop	**0.34**	**0.92**	3.15	1.91	**1.58**

[1] All errors are represented as APE(m).
[2] All test sequences are with loop closure.

C. *Kitti and Kitti-360 Benchmark*

In Table. IV, we compare our complete algorithm with the current state-of-the-art SLAM algorithm on the KITTI benchmark and use $Avg\ T_{rel}$ and $Avg\ R_{rel}$ to evaluate the performance. Our algorithm, on average, can achieve 0.63% in translational error and 0.23/100m in rotational error for training set, 0.83% in translational error and 0.24/100m in rotational error for testing set. Compared to other methods, we obtain the best performance over most of the sequences in the training set and show the best robustness.

We also evaluate our algorithm in KITTI-360 Semantic SLAM trajectory evaluation benchmark for verifying the generalization ability to diffierent data sources. Diffierent from KITTI benchmark, KITTI-360 pays more attention to the global accuracy of trajectory, and adopts the standard Absolute Pose Error (APE) as metrics for evaluating pose estimation. In Table. V, we shows errors in four testing sequences about several methods. Note that all parameters in our approach are pre-trained on KITTI dataset and all

parameters are kept constant. Compared to other methods, **Ours-Loop** shows the best performance in average APE. In addition, all testing sequences are with loop closure, we note the great improvement in test3 as its longest mileage and the most number of loops. However, test2 has a large number of repeat loops, few of which actually work, resulting in insignificant improvement. The number of working loops can actually be increased by adjusting the parameters of the loop closure candidate frames.

VIII. Conclusion

In this paper, we propose a Semantic-Enhanced Lidar-Visual Odometry fusion framework named SELVO and embed a novel semantic loop closure detection into it. We introduce semantic constraints into lidar odometry to improve registration accuracy. In order to overcome the poor performance of lidar in degraded scenes, we first introduce the concept of degradation rate based on semantic distribution. The robustness and generalisation of our approach are ensured by jointly optimising the poses of the camera and lidar. A novel semantic loop closure detection method has been improved to further enhance our global accuracy. We have verified the leading performance of our method on a large number of datasets, even for some challenging scenes. In future work, we will further investigate multi-sensor collaborative SLAM and explore the application of semantic information on point cloud feature extraction, we also keep focusing on analysis of richer semantic information on degraded scene recognition.

References

[1] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[2] Yiyi Liao, Jun Xie, and Andreas Geiger. Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[3] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 224–236, 2018.

[4] Jerome Revaud, Philippe Weinzaepfel, César De Souza, Noe Pion, Gabriela Csurka, Yohann Cabon, and Martin Humenberger. R2d2: repeatable and reliable detector and descriptor. *arXiv preprint arXiv:1906.06195*, 2019.

[5] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020.

[6] Jiaming Sun, Zehong Shen, Yuang Wang, Hujun Bao, and Xiaowei Zhou. Loftr: Detector-free local feature matching with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8922–8931, 2021.

[7] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014.

[8] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765. IEEE, 2018.

[9] Han Wang, Chen Wang, Chun-Lin Chen, and Lihua Xie. F-loam : Fast lidar odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4390–4396, 2021.

[10] Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

[11] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 2022.

[12] Haoyang Ye, Yuying Chen, and Ming Liu. Tightly coupled 3d lidar inertial odometry and mapping. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3144–3150. IEEE, 2019.

[13] Youngwoo Seo and Chih-Chung Chou. A tight coupling of vision-lidar measurements for an effective odometry. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1118–1123. IEEE, 2019.

[14] Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 5692–5698. IEEE, 2021.

[15] Raul Mur-Artal and Juan D Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[16] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and François Goulette. Ct-icp: Real-time elastic lidar odometry with loop closure. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 5580–5586. IEEE, 2022.

[17] Steven A Parkison, Lu Gan, Maani Ghaffari Jadidi, and Ryan M Eustice. Semantic iterative closest point through expectation-maximization. In *BMVC*, page 280, 2018.

[18] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. Suma++: Efficient lidar-based semantic slam. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537. IEEE, 2019.

[19] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet ++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220, 2019.

[20] Guibin Chen, Bosheng Wang, Xiaoliang Wang, Huanjun Deng, Bing Wang, and Shuo Zhang. Psf-lo: Parameterized semantic features based lidar odometry. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5056–5062. IEEE, 2021.

[21] Lin Li, Xin Kong, Xiangrui Zhao, Tianxin Huang, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. Ssc: Semantic scan context for large-scale place recognition. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2092–2099. IEEE, 2021.

[22] Lin Li, Xin Kong, Xiangrui Zhao, Wanlong Li, Feng Wen, Hongbo Zhang, and Yong Liu. Sa-loam: Semantic-aided lidar slam with loop closure. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7627–7634. IEEE, 2021.

[23] Johannes Graeter, Alexander Wilczynski, and Martin Lauer. Limo: Lidar-monocular visual odometry. In *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 7872–7879. IEEE, 2018.

[24] Jiarong Lin and Fu Zhang. R 3 live: A robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 10672–10678. IEEE, 2022.

[25] Han Wang, Chen Wang, and Lihua Xie. Intensity-slam: Intensity assisted localization and mapping for large scale environment. *IEEE Robotics and Automation Letters*, 6(2):1715–1721, 2021.

[26] Giseop Kim and Ayoung 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)*, pages 4802–4809. IEEE, 2018.

[27] Michael Grupp. evo: Python package for the evaluation of odometry and slam. `https://github.com/MichaelGrupp/evo`, 2017.

[28] Han Wang, Chen Wang, and Lihua Xie. Intensity scan context: Coding intensity and geometry relations for loop closure detection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2095–2101. IEEE, 2020.

[29] Ji Zhang and Sanjiv Singh. Low-drift and real-time lidar odometry and mapping. *Autonomous Robots*, 41(2):401–416, 2017.