2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

SAGE-ICP: Semantic Information-Assisted ICP

Jiaming Cui, Jiming Chen, and Liang Li

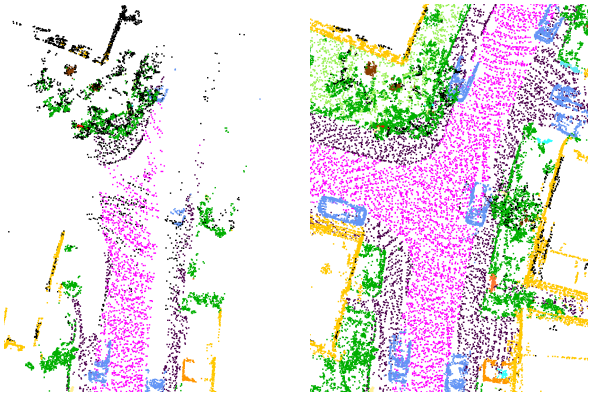
***Abstract*— Robust and accurate pose estimation in unknown environments is an essential part of robotic applications. We focus on LiDAR-based point-to-point ICP combined with effective semantic information. This paper proposes a novel semantic information-assisted ICP method named SAGE-ICP, which leverages semantics in odometry. The semantic information for the whole scan is timely and efficiently extracted by a 3D convolution network, and these point-wise labels are deeply involved in every part of the registration, including semantic voxel downsampling, data association, adaptive local map, and dynamic vehicle removal. Unlike previous semantic-aided approaches, the proposed method can improve localization accuracy in large-scale scenes even if the semantic information has certain errors. Experimental evaluations on KITTI and KITTI-360 show that our method outperforms the baseline methods, and improves accuracy while maintaining real-time performance, *i.e.*, runs faster than the sensor frame rate.**

I. Introduction

Real-time and accurate pose estimation is the basis for mobile robots that require navigation in unknown environments autonomously, where visual odometry (VO) [1], LiDAR odometry (LO) [2] and multi-sensor fusion odometry [3] have received much attention. Cameras are already widely used in commercialized autonomous driving systems due to their low cost, but they are susceptible to performance deterioration caused by illumination changes and scale uncertainty. LiDAR can obtain accurate distance measurements directly and is more robust to such changes. With the recent availability of long-range, low-cost, and high-resolution Solid-State-LiDAR, it has become more suitable and affordable for applying LiDAR-based localization systems in real large-scale scenes.

With the development of neural networks, semantic information of the mapped area can be acquired from LiDAR point clouds directly [4]–[7], which has been used to assist SLAM like generating high-precision semantic maps [8], improving pose estimation accuracy [9], [10] or participating in loop closure detection [11], [12]. However, it is difficult to be real-time due to the high time complexity of semantic segmentation and complex iterative pose optimization process. Moreover, the accuracy of most LiDAR-based semantic methods depends largely on point cloud semantic segmentation results [9], which means that lots of effort is required to label the data and train the model for different environments.

To tackle the above issues, we present SAGE-ICP, a novel semantic information-assisted ICP. Our work is inspired by the idea of KISS-ICP [13], which "keeps it small and simple". The original point-to-point ICP [14] requires correctly correlated point pairs and relatively accurate initial values which are not always possible. KISS-ICP improves the performance of ICP but the semantic information is not incorporated in the registration framework. In this paper, we make reasonable use of semantic information to find pairs of correlation points on top of the ICP framework. Nevertheless, whether the points from some objects are dynamic or static cannot be determined only by their semantic labels, *e.g.*, if the car is moving on the road or parked. Filtering out all the possible dynamic points may lead to worse performance of LO as pointed out in [8]. To deal with this problem, we use prior knowledge to distinguish between dynamic and static vehicles. Instead of gradually removing dynamic objects when updating the map, we remove the relevant point cloud directly before pose estimation, so they have less impact on localization accuracy. On the other hand, the raw point clouds are usually downsampled to reduce the computational time of LO. However, there may exist small objects containing useful information being filtered out in this manner. To solve this problem, different voxel sizes are used for downsampling in this paper and the local map has adaptive storage density for various semantic point clouds.



(a) Before (b) After

Fig. 1. An example of our adaptive voxel map. (a) and (b) show how our local map is updated before and after the mobile robot passes through the intersection. Points without semantic information (marked in black) are gradually replaced with semantic points.

Inherited from KISS-ICP, the theme of our work is "simplicity always gives enlightenment". Fig. 1 shows an example of our semantic mapping result. The main contributions are summarized as follows:

- We propose SAGE-ICP, a semantic information-assisted point-to-point ICP. It reserves the small and simple characteristics of KISS-ICP, and makes full use of se-

This work is supported by the National Natural Science Foundation of China (62088101/62203383). The authors are with the College of Control Science and Engineering, Zhejiang University, Hangzhou, 310027, China.

mantic information to improve accuracy while retaining a high odometry frame rate. The code is released at https://github.com/NeSC-IV/sage-icp.
- We propose an adaptive voxel map with higher storage density for some scarce and critical semantic information.
- We train the model on the SemanticKITTI [15], [16], and extensively evaluate our proposed approach on the KITTI Odometry Benchmark, KITTI road sequences [17] and KITTI-360 [18]. The results show that our method outperforms other methods and is robust to erroneous semantic segmentation results.

II. Related Work

A. Point Cloud Registration

The mainstream point cloud registration methods can be divided into two categories. The first is ICP-based methods [13], [14], [19]–[21], which require an initial guess on the transformation and optimize the objective function, *e.g.*, the sum of distances between pairs of associated points, iteratively. ICP-based methods require nearest neighbor (NN) search to find associated point pairs, which is time-consuming. The structure of KD-tree is widely used to speed up the correspondence point pairs association. FAST-LIO2 [22] proposes an *ikd-Tree* structure that enables incremental map updates, reducing the time for building KD-tree. Moreover, it incorporates IMU to provide an initial pose value and achieves a high convergence speed. Based on *ikd-Tree*, Bai *et al*. [23] propose a voxel-based LIO algorithm which further speeds up the registration process.

Another type of method is the feature-based approach. They first extract local features, such as planes, lines, and corner points, which are then used to estimate the transformation matrix. Zhang *et al*. [2] propose LOAM that aligns only planar and edge features to a sparse feature map, which reduces the number of points used for registration. Several other studies [24]–[26] have been built upon the LOAM framework with the goal of enhancing both accuracy and speed, resulting in advancements tailored to specific scenarios. However, these methods are sensitive to noise and incorrect matches. Moreover, careful and tedious parameter tuning is required depending on sensor resolution, experimental scenes, *etc*.

Recent new approaches focus on the runtime and accuracy of the system. CT-ICP [19] integrates motion undistortion into registration and performs well in several datasets. However, it requires a priori knowledge of the robot's motion model. In contrast, KISS-ICP [13] eliminates the need for sophisticated optimization techniques to address motion distortion, while achieving effective results with only the constant velocity model. This strategy allows it to run on various types of equipment without fine-tuning the system for a specific application.

B. Semantic-Assisted LiDAR SLAM

The development of point cloud semantic segmentation algorithms has made it possible to integrate semantic information in pure LiDAR-based SLAM, particularly for large-scale mapping and localization [8], [27]. SLOAM [27] is designed specifically for forest inventory. SuMa++ [8] filters dynamic objects from a surfel-based map, extending ICP with semantic constraints. Some recent studies have also utilized semantic information for loop closure detection [11], [28], [29]. RINet [28] addresses place recognition via a global descriptor combining semantic and geometric information. SA-LOAM [11] integrates a semantic-based loop closure detection method in LOAM, and PADLoC [29] uses transformer-based panoptic attention for deep loop closure detection. In our approach, we focus on utilizing semantic information in LiDAR odometry. It is important to note that pose graph optimization and loop closure detection can be easily integrated into our approach.

III. SAGE-ICP - Simplicity Always Gives Enlightenment

This paper aims to leverage semantic information in registration. Fig. 2 shows the pipeline of our approach. Firstly, point-wise semantic labels are assigned to the raw point cloud through a point cloud semantic segmentation network [5]. Then the semantic points belonging to vehicles are further processed to obtain instance segmentation results, while the potential dynamic vehicles are removed using prior knowledge. Subsequently, the point cloud is downsampled individually based on their respective semantic categories, thereby ensuring diverse semantic categories within the downsampled point cloud. The next step involves the alignment of the preprocessed point cloud with the local map. This alignment employs an adaptive threshold for data association [13], where the selection of associated points takes into account both the associativity of semantic labels and the Euclidean distance between points. Finally, the adaptive local voxel map is updated, enhancing its capability to store and update more critical semantic points. A comprehensive elaboration of these sequential procedures is provided in the subsequent sections.

A. Real-time Semantic Segmentation

Cylinder3D [5] is a cylindrical partition and asymmetrical 3D convolution network which combines point-wise features and voxel-wise inference results together, outperforming range image-based 2D convolution methods [4], [8]. While the original Cylinder3D is hard to be real-time on a laptop for robotic applications, Hou et al. [7] propose a point-to-voxel knowledge distillation approach named PVD, significantly speeding up the Cylinder3D model and performing best in pure LiDAR semantic segmentation on the SemanticKITTI [15]. The inference time of the original Cylinder3D model is 170 ms and PVD is 76 ms, but the code of knowledge distillation is not yet open-sourced.

Building upon the original Cylinder3D model, we propose a novel method capable of achieving real-time semantic segmentation within the ROS2 framework [30] on a laptop (see Fig. 2). Our method consists of two key modifications to the point cloud segmentation procedure. First, we split the

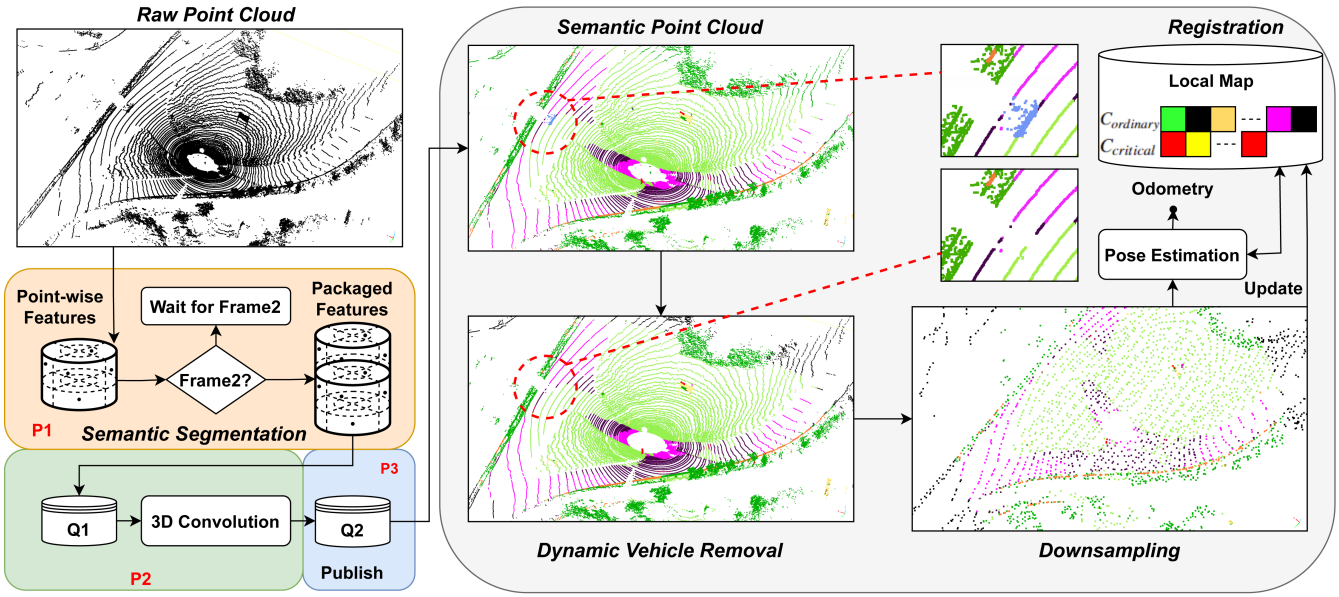


Fig. 2. Pipeline of our approach. We perform semantic segmentation in three separate processes to achieve real-time operation in the ROS2 environment.

entire semantic labels inference process into three distinct stages; Second, we merge pairs of successive frames to be fed into the neural network. The first frame of the raw point cloud is first rasterized by the cylindrical partition and point-wise features are extracted by MLP. To ensure the real-time performance of process 1 (P1 in Fig. 2), we randomly select a point in each voxel to extract features as the voxel-wise features. The second frame is treated in the same way and then packaged into queue 1 along with the first frame. Asymmetrical 3D convolution networks are then used to generate the voxel-wise outputs, while all points within the same voxel are considered to have identical labels (P2 in Fig. 2). Two frames of point clouds with semantic information are pushed back sequentially into queue 2, while process 3 publishes them in turn (P3 in Fig. 2). Since P2 can perform an inference in less than 200 ms, and the three processes are independent, P3 can publish a frame of semantic point cloud every 100 ms, matching the frequency of the raw point cloud. A detailed analysis of time consumption is shown in Table III.

B. Semantic Point Cloud Preprocessing

Our preprocessing of the semantic point cloud obtained from Section III-A consists of two parts: dynamic points removal and class-wise downsampling, to achieve faster convergence, higher robustness, and more accurate registration results.

Removing Dynamic Instance. Real-world scenes often have a large number of moving objects. Traditional geometric feature-based approaches suffer from environmental changes, causing significant registration errors. SuMa++ [8] filters dynamic surfels by checking semantic consistency when a new frame is updated to the map, and has shown that directly removing all dynamic semantic categories leads to a degeneration in odometry accuracy. As a result, it does not remove moving objects before the point cloud registration, which means that dynamics still affect pose optimization to some extent. Since static vehicles tend to have regular positions in real scenarios (*e.g.*, on parking lots, by the sidewalk), we propose a simple yet effective approach to distinguish them from dynamic ones. Fig. 2 shows an example before and after filtering dynamics. We first categorize semantic points into two distinct groups: dynamic $\mathbf{C_d}$ and static $\mathbf{C_s}$. Subsequently, we utilize Euclidean point cloud clustering from the PCL library [31] to effectively distinguish various instances within $\mathbf{C_d}$ denoted as $\mathbf{I_d}$. Then we deal with each instance $i_d^k \in \mathbf{I_d}$ individually. For each instance, we search for the points $\mathbf{P_r^k}$ within a certain range d_r around i_d^k on the KD-tree constructed from all the static points within $\mathbf{C_s}$. We then tally the count of semantic points $\mathbf{P_{ic}^k}$ of specific categories (*e.g.*, parking lots, sidewalk). If the value of θ_{ic} surpasses a designated threshold θ_0, we suppose that the instance i_d^k is stationary:

$$\theta_{ic} = \frac{|\mathbf{P_{ic}^k}|}{|\mathbf{P_r^k}|} \tag{1}$$

where $|\mathbf{P^k}|$ is the number of points in $\mathbf{P^k}$.

Semantic Subsampling. Extracting keypoints from dense raw point clouds is an essential step for feature-based scan registration [2], [24]–[26], which usually has many parameters to be tuned for different scenes. Since semantic information is an advanced environmental feature, complex geometric feature extraction can be avoided. We adopt point cloud subsampling approach from KISS-ICP [13] to maintain one point per voxel in the original coordinates. Moreover, we further use different voxel grid sizes for different categories, to prevent some small but critical points (*e.g.*, road signs, lane lines) inevitably from being filtered out.

C. Semantic Assisted Association

Data association is a prerequisite for iterative pose optimization, *i.e.*, finding the correct point associations. Most ICP-based methods adopt the standard nearest neighbor approach to handle this task, while semantic methods [8]–[10] directly associate two points with the same label, or give a weight coefficient according to whether the semantic label of the associated point pair is consistent.

As the semantic segmentation result cannot be 100% correct, our strategy is to integrate semantic labels into the nearest neighbor search and avoid associating points from different categories. For each source point in the preprocessed point cloud $p_s^k \in \mathbf{P_{pre}}$, we firstly calculate its voxel coordinates based on motion prediction $T_{pred,t}$, and collect all points $\mathbf{P_{ls}^k}$ within that voxel and neighboring voxels from the local map. The semantic-Euclidean distance $d_{se,j}$ of each point $p_{ls,j}^k \in \mathbf{P_{ls}^k}$ from the source point is defined as

$$d_{se,j} = \gamma ||p_{ls,j}^k - p_s^k||_2 \tag{2}$$

where semantic correlation coefficient γ is given by

$$\gamma = \begin{cases} \gamma_0, & \text{if } l(p_{ls,j}^k) = l(p_s^k) \text{ or } l(p_{ls,j}^k) \cdot l(p_s^k) = 0 \\ 1, & \text{otherwise} \end{cases} \tag{3}$$

where $l(p_{ls,j}^k)$, $l(p_s^k)$ are semantic labels of $p_{ls,j}^k$ and p_s^k, respectively, and γ_0 is a constant in range $(0,1)$. $l(\cdot) = 0$ indicates that the point has no semantic category. Since the inference error of Cylinder3D [5] increases with distance, we manually remove labels from point cloud 50 meters away from the center of the LiDAR. Then we get a target point $p_{ls,\bar{j}}^k \in \mathbf{P_{ls}^k}$ with the smallest semantic-Euclidean distance $d_{se,\bar{j}}$ and store this point pair in the collection $\mathscr{C}(\tau_t)$ if their Euclidean distance is below adaptive threshold τ_t computed by constant velocity model

$$\begin{aligned} \tau_t &= 3\sigma_t, \\ \sigma_t &= \sqrt{\frac{1}{|\mathbf{M}_t|} \sum_{i \in \mathbf{M}_t} \delta(\Delta T_i)^2}, \\ \mathbf{M}_t &= \{ i \mid i < t \wedge \delta(\Delta T_i) > \delta_{\min} \} \end{aligned} \tag{4}$$

where ΔT_i represents how much deviation there is between odometry and motion prediction, $\delta(\Delta T_i)$ is maximum point displacement, see [13] for detail.

Finally, we perform a robust optimization minimizing residuals for all elements in $\mathscr{C}(\tau_t)$ and repeat an iterative process similar to other ICP methods.

D. Adaptive Voxel Map

The widely used approaches to store local maps are voxel grids [2], surfels [8] or *ikd-Tree* [22]. Hash table is used to store voxels in recent works [13], [23], allowing memory-efficient map storage and faster nearest neighbor search. Our adaptive voxel map is based on [13], with higher storage density for some critical semantic points and the ability to update point cloud semantic labels (see Algorithm 1). When the number of points in the voxels is in the range of $[N_1, N_2]$, the ordinary points $p_i^* \in \mathbf{C_{ordinary}}$ are used to replace the unlabeled points that have been stored, while the critical points $p_i^* \in \mathbf{C_{critical}}$ are stored in additional storage space. In this way, we can guarantee that our adaptive voxel map contains richer semantic information than other approaches. We also remove voxels that are too far away from the current pose p_t to save memory usage.

Algorithm 1: Adaptive Voxel Map

Input : current point cloud in map coordinate $\mathbf{P^*_{pre,t}}$, current pose p_t, local map M_{t-1}
Output: updated local map M_t

foreach *semantic point* $p_i^* \in \mathbf{P^*_{pre,t}}$ **do**
  $v_i^* \in \mathbb{Z}^3 \leftarrow$ Get voxel grid index from p_i^*;
  $n_i \leftarrow$ The number of points already in $\mathbf{P}(\mathbf{v_i^*})$;
  if $n < N_1$ **then** Add p_i^* to $\mathbf{P}(\mathbf{v_i^*})$;
  else if $n \geq N_1 \& n \leq N_2$ **then**
    if $l(p_i^*) \in \mathbf{C_{ordinary}}$ **then**
      foreach $p_{i,j} \in \mathbf{P}(\mathbf{v_i^*})$ **do**
        if $l(p_{i,j}) = 0$ **then**
          Replace $p_{i,j}$ with p_i^*;
        end
      end
    else if $l(p_i^*) \in \mathbf{C_{critical}}$ **then** Add p_i^* to $\mathbf{P}(\mathbf{v_i^*})$;
  end
end
foreach *non-empty* $\mathbf{P}(\mathbf{v_k}) \in M_{t-1}$ **do**
  if $||\mathbf{P}(\mathbf{v_k})_{\mathbf{center}} - p_t||_2 \geq r_{max}$ **then**
    Remove $\mathbf{P}(\mathbf{v_k})$
  end
end

IV. Experiments

We design our experiments to demonstrate that our system (i) outperforms state-of-the-art odometry algorithms in most cases, and (ii) is robust to erroneous semantic segmentation results. All parameters are shown in Table I. For all sequences from KITTI and KITTI-360, we correct LiDAR scans for an angle of 0.205° to be set up as [19].

We evaluated our approach on a laptop with a 4.70 GHz 14-core Intel i7-12700H CPU with 16GB of RAM and an Nvidia GeForce RTX 3070 Ti with 8 GB RAM. The Cylinder3D [5] model for point cloud semantic segmentation is trained using all sequences from [15] and semantic labels are provided for KITTI Odometry Benchmark [17]. It is noteworthy that we use the same model when testing other datasets, which means that our model inevitably produces greater errors when segmenting these data than KITTI Odometry Benchmark. Moreover, We also evaluated Rangenet++ [4] to prove that our approach is also suitable for other semantic segmentation models. Both semantic segmentation and odometry part are implemented on ROS2 Humble [30].

A. KITTI Odometry Benchmark

The KITTI Odometry Benchmark [17] consists of 11 training sequences with LiDAR scans. We test the Relative

TABLE I
PARAMETERS OF SAGE-ICP

Datasets	S_{road}	S_{plant}	S_{object}	$S_{vehicle}$	$S_{building}$	$S_{unlabel}$	γ_0	θ_0	N_1	N_2	r_{max}	τ_0	$\delta_{\min}$
KITTI Odometry	0.6m	0.9m	0.8m	0.6m	1.0m	1.0m	0.4	0.5	20	40	100m	2m	0.1m
KITTI Raw/KITTI-360	1.0m	1.0m	0.5m	0.5m	0.5m		0.8						

S_{road} is downsampling grid size of road, parking, sidewalk and other ground; S_{plant} stands for vegetation and terrain; S_{object} stands for trunks, lane-markings, poles, traffic signs and other objects, small but critical points mentioned in Section III-B; $S_{building}$ stands for buildings and fences.
Due to the different semantic segmentation accuracy of different datasets, we select appropriate parameters for three datasets separately.

TABLE II
RELATIVE POSE ERROR ON KITTI AND KITTI-360

KITTI Odometry	00	01	02	03	04	05	06	07	08	09	10	Avg.
LOAM [2]	0.78/-	1.43/-	0.92/-	0.86/-	0.71/-	0.57/-	0.65/-	0.63/-	1.12/-	0.77/-	0.79/-	0.84/-
KISS-ICP [13]	0.51/0.19	0.72/**0.11**	0.52/0.15	0.66/**0.16**	0.35/0.14	0.30/0.14	0.26/**0.08**	**0.32**/0.17	0.82/**0.18**	0.49/**0.13**	0.57/0.19	0.50/0.15
SuMa++* [8]	0.65/0.22	1.63/0.47	3.54/**0.14**	0.67/0.47	0.34/0.27	0.40/0.19	0.47/0.27	0.39/0.28	1.01/0.34	0.58/0.20	0.67/0.30	0.94/0.28
SA-LOAM$^{\circ\star}$ [11]	0.59/**0.18**	1.89/0.48	0.79/0.27	0.87/0.46	0.59/0.35	0.37/0.16	0.52/0.24	0.41/0.18	0.84/0.25	0.77/0.25	0.78/0.35	0.76/0.28
Ours-RangeNet	0.51/0.19	0.65/**0.11**	0.50/0.15	0.66/**0.16**	0.33/**0.12**	0.28/**0.13**	**0.25**/0.09	0.33/**0.15**	**0.80/0.18**	**0.47**/0.14	0.55/0.17	**0.48**/0.15
Ours-Cylinder3D	0.51/0.19	0.67/**0.11**	0.51/0.16	**0.64**/0.17	0.33/0.13	0.29/0.14	0.26/0.09	**0.32/0.15**	0.81/**0.18**	**0.47**/0.14	0.55/0.17	0.49/0.15
Ours-ST	**0.50**/0.19	**0.64/0.11**	**0.49**/0.15	0.67/**0.16**	**0.32**/0.14	**0.27/0.13**	**0.25/0.08**	**0.32/0.15**	**0.80/0.18**	0.48/0.14	**0.51/0.16**	**0.48/0.14**

KITTI-360	00	-	02	03	04	05	06	07	-	09	10	Avg.
KISS-ICP [13]	0.73/0.33		**0.61/0.30**	**0.58/0.20**	**0.75/0.35**	0.71/0.43	0.89/0.38	0.51/**0.15**		0.78/**0.31**	**0.70/0.25**	**0.69**/0.30
Ours-RangeNet	0.72/0.33		**0.61**/0.31	0.65/0.22	0.79/0.36	**0.68**/0.42	**0.67/0.31**	0.55/0.17		0.78/**0.31**	0.79/0.27	**0.69**/0.30
Ours-Cylinder3D	**0.67/0.30**		0.62/0.31	0.62/0.21	0.77/**0.35**	0.69/**0.40**	0.74/0.34	**0.50/0.15**		**0.77**/0.32	0.77/0.27	**0.69/0.29**

KITTI-PART	30	31	32	33	34	Avg1		t0	t1	t2	t3	Avg2
KISS-ICP [13]	0.56/**0.13**	1.75/0.59	1.56/0.43	1.41/**1.30**	**0.93**/0.16	1.24/0.52		0.33/**0.22**	**0.33/0.18**	0.94/0.44	0.74/0.38	0.59/0.31
Ours-RangeNet	0.55/0.16	2.18/1.04	2.28/0.72	**1.39/1.30**	0.95/**0.15**	1.47/0.67		**0.32**/0.24	0.38/0.25	0.88/0.41	**0.71**/0.40	0.57/0.32
Ours-Cylinder3D	**0.51/0.13**	**1.67/0.50**	**1.39/0.35**	**1.39/1.30**	0.95/**0.15**	**1.18/0.49**		0.33/0.25	**0.33**/0.20	**0.85/0.39**	0.72/**0.37**	**0.56/0.30**

RTE in % / RRE in degrees per 100m. The best performance for each sequence is marked in bold numbers. Approaches marked with * indicate semantic assisted odometry and $^{\circ}$ contain loop closures. The result of LOAM is from the origin paper [2] while both SuMa++ and SA-LOAM are from [11].
We renamed KITTI road sequences "2011_09_26_drive_0015 - 2011_09_26_drive_0032" into KITTI-PART sequences 30-34, and KITTI-360 test SLAM "00-03" into "t0-t3".

Translation Error (RTE) in percentage and the Relative Rotational Error (RRE) in degrees per 100 m defined by KITTI, which averages the trajectory errors of length ranging from 100 to 800 m. We present the results of our approach with semantic segmentation values by Cylinder3D (Ours-Cylinder3D) and Rangenet++ (Ours-Rangenet), and also report the results with ground-truth segmentation values (Ours-ST) respectively. The state-of-the-art approach KISS-ICP [13] is chosen as our baseline. Besides, we compare our results with LOAM [2], SuMa++ [8] and SA-LOAM [11] in Table II. LOAM is the state-of-the-art pure LiDAR-based method on the KITTI benchmark, while SuMa++ and SA-LOAM are semantic-aided LiDAR SLAM systems.

Both KISS-ICP and our approach outperform other methods on most sequences. Compared with KISS-ICP, our method has better performance on each sequence with different margins, especially 01 and 10. These sequences are collected from highways or the countryside, which means that there are more dynamic vehicles and fewer geometric feature points in the raw point clouds. We also find that in other sequences collected from urban, most feature points can be obtained directly by classical methods, *i.e.*, building corner points or planes. Our method is slightly better than the baselines though the superiority of our semantic method is not as obvious as that in the dynamic or featureless environments. Although semantic labels generated by Cylinder3D or Rangenet++ are not completely reliable, the results of Ours-Cylinder3D, Ours-Rangenet++, and Ours-ST show that semantic information with partial errors can also benefit localization. Besides, similar to [8], we find that removing all points belonging to vehicles leads to worse performance as parked cars are also a reliable source of feature points in some environments.

B. Generalization Validation

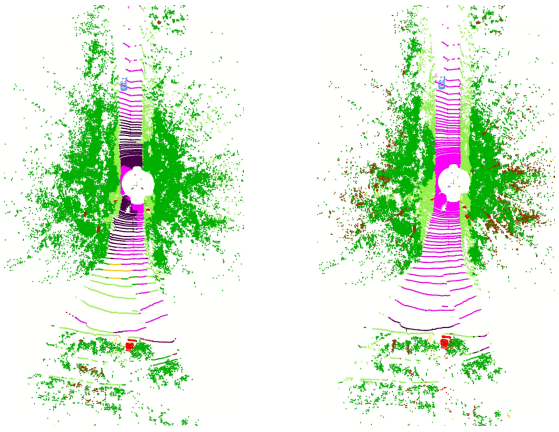
To further evaluate the generalization ability of our system and the advantages of our approach in dynamic scenes, we compare the same metrics with baseline on KITTI raw data and KITTI-360. The scans in KITTI raw data are not motion-corrected, but the data frame numbers correspond across all sensor streams, and GPS/OXTS data is reliable due to less interference. We selected the first five sequences from the road category collected in the countryside or highway. There are few distinct features in these scenes and flooded with high-speed moving vehicles, which are challenging for SLAM. Results of KITTI-PART in Table II show that our semantic information-assisted method with the Cylinder3D model achieves more robust and accurate localization performance than KISS-ICP, which suffers from false correspondences on moving objects. Meanwhile, lower rotation error means that our method can capture small but critical feature points from degraded environments such as road signs on the highway. Rangenet++ has poor segmentation accuracy for far and small objects (see Fig. 3), resulting in large errors.

KITTI-360 has nine sequences that are much longer than KITTI, with very minimal overlap with KITTI in terms of trajectories. The testing SLAM sequences are fragments of

TABLE III
TIME CONSUMPTION ANALYSIS (MS)

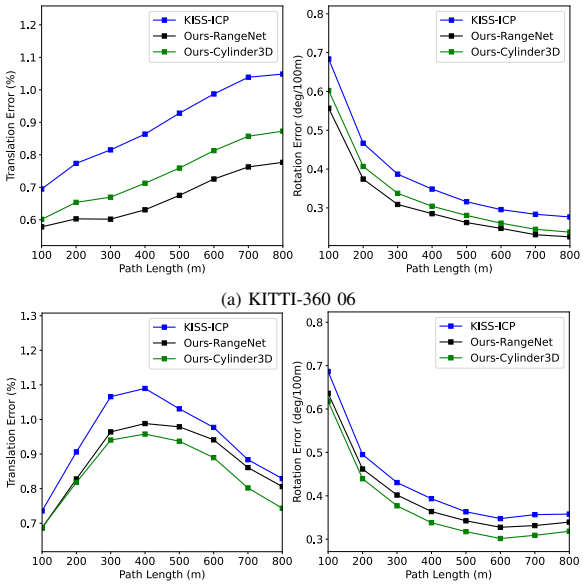
Semantic Segmentation				Registration				
MLP-Frame1	MLP-Frame2	conv	publish	RDI*	Downsampling	Optimization	Update	Total
66.94	94.67	168.42	20.82	30.09	40.13	15.13	0.82	86.17

* Removing Dynamic Instance



(a) Cylinder3D (b) Rangenet++

Fig. 3. Semantic segmentation results. (a) Cylinder3D is more accurate for the segmentation of small targets, *e.g.*, trunks (marked with brown), and road signs (marked with red). (b) Rangenet++ is more accurate for semantic segmentation over large areas nearby, *e.g.*, roads (marked with pink).



(a) KITTI-360 06

(b) KITTI-360 t2

Fig. 4. Relative pose errors versus length ranging from 100 to 800m. (a) The field of view is filled with a large number of buildings and intersecting roads. (b) Sequence t2 is similar to KITTI raw data, in that only a small amount of feature information can be extracted in the environment.

origin sequence 08, and we renamed them into sequences t0-t3, listed in KITTI-PART. Our method performs better or equally compared to the baseline methods in most sequences (see Table II and Fig. 4) but fails in 03 and 10 due to many vans and trucks being incorrectly inferred into buildings. This error can be mitigated by improving the semantic segmentation model in future work.

C. Real-time Performance and Ablation Studies

Table III presents the processing time of each part of our system. The semantic segmentation part can achieve real-time performance thanks to three independent processes (see Section III-A). The frame rate of the registration part is about 11.60 Hz, which has surpassed the frame rate of LiDAR. These results show that our system is ready for real-world applications.

To verify the positive effect of each part of our system on improving the accuracy of pose estimation, we removed parts removing dynamic instance (RDI, Section III-B), semantic subsampling (SS, Section III-B), semantic assisted association (SAA, Section III-C) and adaptive voxel map (AVM, Section III-D) respectively. We present the testing results of sequence t2 in Table IV to verify the effectiveness of each module. Note that removing the semantic subsampling part can even lead to increased error, because many tiny semantic key features are filtered out by the traditional downsampling method, leaving many isolated noise-like points challenging for semantic-based point-pair association.

TABLE IV
ABLATION STUDIES

	RTE (%)	RRE (deg)
Base	0.9392	0.44
Ours-wo-RDI	0.8818	0.41
Ours-wo-SS	0.9627	0.45
Ours-wo-SAA	0.8537	0.40
Ours-wo-AVM	0.8680	0.41
Ours	**0.8467**	**0.39**

V. CONCLUSIONS

This paper presents a semantic information-assisted ICP. We construct an online pose estimation system integrating pure LiDAR semantic information. Evaluation of the proposed method on KITTI Odometry Benchmark, KITTI-360, and specifically KITTI road sequences demonstrates that our approach can improve the localization accuracy in dynamic scenes with fewer effective geometric features. Since the current semantic segmentation networks have poor generalization ability for different types of LiDAR, we can only use the same model of LiDAR (e.g., 64-channel mechanical LiDAR) for training and testing. In future work, we plan to explore the fusion of semantic information in loop closure detection and incorporate other sensors to further improve semantic segmentation and pose estimation.

REFERENCES

[1] H. Lategahn and C. Stiller, "Vision-only localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 3, pp. 1246–1257, 2014.

[2] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time.," in *Robotics: Science and Systems*, vol. 2, pp. 1–9, Berkeley, CA, 2014.

[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] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4213–4220, IEEE, 2019.

[5] X. Zhu, H. Zhou, T. Wang, F. Hong, Y. Ma, W. Li, H. Li, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9939–9948, 2021.

[6] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11108–11117, 2020.

[7] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8479–8488, 2022.

[8] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4530–4537, IEEE, 2019.

[9] Y. He, L. Ma, Z. Jiang, Y. Tang, and G. Xing, "Vi-eye: Semantic-based 3d point cloud registration for infrastructure-assisted autonomous driving," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, pp. 573–586, 2021.

[10] S. A. Parkison, L. Gan, M. G. Jadidi, and R. M. Eustice, "Semantic iterative closest point through expectation-maximization.," in *BMVC*, p. 280, 2018.

[11] L. Li, X. Kong, X. Zhao, W. Li, F. Wen, H. Zhang, and Y. Liu, "Sa-loam: Semantic-aided lidar slam with loop closure," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7627–7634, IEEE, 2021.

[12] X. Kong, X. Yang, G. Zhai, X. Zhao, X. Zeng, M. Wang, Y. Liu, W. Li, and F. Wen, "Semantic graph based place recognition for 3d point clouds," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8216–8223, IEEE, 2020.

[13] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss, "Kiss-icp: In defense of point-to-point icp–simple, accurate, and robust registration if done the right way," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 1029–1036, 2023.

[14] P. J. Besl and N. D. McKay, "Method for registration of 3-d shapes," in *Sensor fusion IV: control paradigms and data structures*, vol. 1611, pp. 586–606, Spie, 1992.

[15] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9297–9307, 2019.

[16] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, "Towards 3d lidar-based semantic scene understanding of 3d point cloud sequences: The semantickitti dataset," *The International Journal of Robotics Research*, vol. 40, no. 8-9, pp. 959–967, 2021.

[17] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on Computer Vision and Pattern Recognition*, pp. 3354–3361, IEEE, 2012.

[18] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.

[19] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "Ct-icp: Real-time elastic lidar odometry with loop closure," in *2022 International Conference on Robotics and Automation (ICRA)*, pp. 5580–5586, IEEE, 2022.

[20] S. Bouaziz, A. Tagliasacchi, and M. Pauly, "Sparse iterative closest point," in *Computer Graphics Forum*, vol. 32, pp. 113–123, Wiley Online Library, 2013.

[21] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized gicp for fast and accurate 3d point cloud registration," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11054–11059, IEEE, 2021.

[22] W. Xu, Y. Cai, D. He, J. Lin, and F. Zhang, "Fast-lio2: Fast direct lidar-inertial odometry," *IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2053–2073, 2022.

[23] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

[24] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5135–5142, IEEE, 2020.

[25] H. Wang, C. Wang, C.-L. Chen, and L. Xie, "F-loam: Fast lidar odometry and mapping," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4390–4396, IEEE, 2021.

[26] T. Shan and B. 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)*, pp. 4758–4765, IEEE, 2018.

[27] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, "Sloam: Semantic lidar odometry and mapping for forest inventory," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 612–619, 2020.

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

[29] J. Arce, N. Vödisch, D. Cattaneo, W. Burgard, and A. Valada, "Padloc: Lidar-based deep loop closure detection and registration using panoptic attention," *IEEE Robotics and Automation Letters*, vol. 8, no. 3, pp. 1319–1326, 2023.

[30] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, "Robot operating system 2: Design, architecture, and uses in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.

[31] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1–4, IEEE, 2011.