Pole-like Objects Mapping and Long-Term Robot Localization in Dynamic Urban Scenarios

Zhihao Wang, Silin Li, Ming Cao, Haoyao Chen* and Yunhui Liu

Abstract—Localization on 3D data is a challenging task for unmanned vehicles, especially in long-term dynamic urban scenarios. Due to the generality and long-term stability, the pole-like objects are very suitable as landmarks for unmanned vehicle localization in time-varying scenarios. In this paper, a long-term LiDAR-only localization algorithm based on semantic cluster map is proposed. At first, the Convolutional Neural Network (CNN) is used to infer the semantics of LiDAR point clouds. Combined with the point cloud segmentation, the long-term static objects pole/trunk in the scene are extracted and registered into a semantic cluster map. When the unmanned vehicle re-enters the environment again, the relocalization is completed by matching the clusters of the local map with the clusters of the global map. Furthermore, the continuous matching between the local and global maps stably outputs the global pose at 2Hz to correct the drift of the 3D LiDAR odometry. The proposed approach realizes localization in the long-term scenarios without maintaining the high-precision point cloud map. The experimental results on our campus dataset demonstrate that the proposed approach performs better in localization accuracy compared with the current state-of-the-art methods. The source of this paper is available at: http://www.github.com/HITSZ-NRSL/long-term-localization.

I. INTRODUCTION

Simultaneous Localization and Mapping (SLAM) can provide location and map information for unmanned vehicles in unknown environments. Many classical SLAM methods [1]–[3] have been proposed and applied in static environments. Because traditional point feature descriptors, such as SIFT, SURF and BRIEF, are robust to uniform changes in light intensity and slight changes in viewpoint. However, the real urban environments are time-varying, such as, lighting changing, weathers and seasons varying, and the movement of objects (e.g., cars parked on both sides of the road may be driven away after a while), etc. And most descriptors do not take the large environmental changes into account. Thus, the dynamic environments introduce difficulties for a unmanned vehicle to localize itself or place recognition when it re-enters a known and mapped environment. Some literatures [4] rely on regularly updating high-definition point cloud maps to deal with the problem of dynamic environments, which is time and computation consuming.

To accomplish the long-term place recognition in dynamic environments, many approaches [5] tried to take multi-experiences into the localization framework. These approaches revealed inherent drawbacks because they need to capture various conditions for the same place as a prior to increase the size of the database of experiences. Contrast to visual appearance, the physical structure of a place rarely changes over time. Hence, utilizing structural information is beneficial for long-term localization [6]. In that sense, a single observation using LiDAR could represent a canonical characteristic of a place, eliminating the need of multiple experiences for robust localization. In this line of research, handcrafted descriptor-based [7] and learning-based [8] methods for place recognition over point clouds have been widely proposed. However, these studies hardly captured the long-term localization requirements, including a slow but massive structural variance (e.g., construction and demolition) and unexpected viewpoint from the road topology change.

Inspired by the work of [9], pole-like objects, such as street lamps, poles of building and tree trunks, etc., are ubiquitous in urban areas; they are long-term stable and invariant under seasonal and weather changes, and their geometric shapes are well-defined. These advantages make pole-like objects suitable as landmarks for accurate and reliable relocalization. Thus, an approach based on semantic cluster is presented for long-term relocalization in urban dynamic environments, relying on pole/trunk landmarks extracted from mobile LiDAR data. We are not the first one to propose this kind of technique in long-term problem. However, to the best of our knowledge, this work is the first one that only uses 3D LiDAR to extract pole-like objects for place recognition and localization in long-term scenarios. To summarize, the main contributions of this paper are three-fold.

- A method to extract semantic cluster of pole-like objects from raw 3D LiDAR points and create a robust semantic cluster map is proposed to resolve the long-term challenge.
- A semantic cluster association algorithm based on geometric consistency is proposed to relocalize unmanned vehicles in long-term scenarios.
- A long-term and real-time localization system is developed based on the robust semantic cluster relocalization module.

II. RELATED WORKS

Thanks to the rapid development of deep learning, some methods [10] based on image semantic segmentation have
been proposed to overcome environmental changes. Compared with the low-level point features, it is a more robust way to describe the environmental changes by using the high-level semantic features. Since geometric properties of the environments are typically more stable than its photometric characteristics, a geometry matching method [11] is proposed to achieve long-term camera localization by extracting object-level features in the environment. Compared with the change of visual characteristics, the structure of the environment is stable, so many solutions based on laser sensors have been researched. Scan Context Image(SCI) [12] is proposed to encode the top view of LiDAR point clouds into a three-channel color image, convert the place recognition problem to a classification problem.

Although many methods have been proposed for localization, few empirical studies showed the effectiveness of LiDAR descriptors on long-term localization capability in urban areas. As the high-level semantic information is robust to describe the environmental changes, a novel semantic graph [13] is proposed for point clouds by reserving the semantic and topological information of the raw point clouds. Thus, place recognition is modeled as graph matching problem. However, this work has not been verified on the long-term place recognition algorithm. A global localization algorithm [14] is proposed by using only a single 3D LiDAR scan at a time, which relies on learning-based descriptors of point cloud segments. But its performance on long-term dataset is underperformance and not stable.

To address above problems, a long-term localization algorithm is proposed on the semantic cluster map of pole/trunks objects. The semantic cluster map is built by extracting the pole-like objects from raw 3D LiDAR points by CNN. Then, the semantic cluster association algorithm based on geometric consistency verification is proposed to obtain the pose transformation between the local semantic cluster map and the global semantic cluster map. Finally, the long-term localization system utilized the pose transformation to correct the drift of LiDAR odometry in real time for high accuracy localization.

III. SYSTEM OVERVIEW

The experimental unmanned vehicle system is shown in Fig.1(a). The RS-Ruby-Lite 3D LiDAR with 80 laser-beam has a vertical field of view (FOV) of 40° and a horizontal FOV of 360°. And the inertial sensor MTi-100 is also equipped with the unmanned vehicle MR1000. Global Navigation Satellite System (GNSS)/Inertial Navigation System(INS) are used as the ground truth for comparison. The proposed system is validated on the laptop with Intel Core i7-7820HK CPU@2.90GHz on Robot Operating System (ROS).

The framework of the long-term localization system is shown as Fig.1(b). The semantic clusters of pole-like objects are extracted by the semantic lables of the point clouds, which are obtained directly through CNN. Then, the semantic clusters of pole-like objects are registered in the global semantic cluster map using the six-degree-of-freedom(6-DOF) pose obtained by SLAM module. When the unmanned vehicle re-enters the environment where the global semantic cluster map has been established, the local semantic cluster map is built by using the pose from SLAM module and the semantic clusters of the current frame. Then, a semantic cluster association algorithm based on the semantic labels and the geometric structure of semantic clusters is proposed to calculate the pose transformation between the local and the global semantic cluster map. The relocalization module includes the semantic clusters registration and association. And the relocalization module continuously runs and stably outputs the global pose at 2Hz to correct the drift of the LiDAR odometry. The fine 6-DOF pose is published in real time after a drift correction. The high accuracy localization of the unmanned vehicle is realized in the long-term scenarios without maintaining a high precision point cloud map.

IV. METHOD

A. Semantic Cluster Extraction

The convolutional neural network RangNet++ [15] is used for semantic segmentation of point clouds. The semantic labels of LiDAR point clouds are annotated on self-made campus dataset and the network of RangNet++ is retrained for inferring the semantic labels of point clouds. The range-image based method [16] is used to segment the point clouds into clusters based on the semantic labels. For each cluster, the label of cluster is voted by the statistical number of the point labels in the cluster, by considering the imperfection of point cloud semantic segmentation. In the end, two kinds of semantic clusters, i.e., trunk and pole, are obtained.

Since the semantic cluster map and each semantic cluster need to be visited frequently, designing a semantic cluster manager is necessary to efficiently manage various attributes and operations of clusters. The manager includes 3D centroid points, 2D centroid points and several attributes, such as label, ID, point clouds and memory address. The memory structure is shown in the Fig.2.

B. Semantic Cluster Registration

The registration of semantic clusters will build a map, and the map includes two types: global semantic cluster map
clusters of each frame from current environment. As the global map need register the ago. The local map represents the geometric information of environments, such as the environment of several months before. The geometric information of pole-like objects in history is represented as shown in Fig.3(b). The clusters mainly include poles and trunks, and they are perpendicular to the horizontal plane in the environment. Therefore, the 2D centroid points are able to represent the geometric information of the clusters, and are registered into the cluster map. The registered 2D semantic cluster centroids are the XY-plane projection of 3D centroids, as shown in Fig.3(b).

C. Semantic Cluster Association

The geometric distribution of neighbor clusters in a certain neighborhood around each cluster is utilized for clusters matching. The geometric information of the cluster to be matched is described as the relative positions and angles between the cluster and its neighbors. And all the geometric positions are simplified on the XY plane for reducing computation cost.

Let \( M_l \) denote the local semantic cluster map of current frame with \( P \) clusters. The map registered by IV.B is loaded as the global semantic cluster map \( M_g \), which contains \( Q \) clusters. And the deinitions are shown as follows:

\[
\begin{align*}
M_l & = \{C_{l1}, C_{l2}, ..., C_{lP}\} \\
M_g & = \{C_{g1}, C_{g2}, ..., C_{gQ}\}
\end{align*}
\]

In the local map \( M_l \), the neighbor clusters of \( C_{li} \) within the search radius \( SR \) are denoted as \( N_{C_{li}} \). Similarly, \( N_{C_{gj}} \) denotes the neighbor clusters of \( C_{gj} \) within the search radius \( SR \) in \( M_g \). And the number of neighbor clusters is \( k_i \) and \( k_g \), respectively. \( N_{C_{li}} \) and \( N_{C_{gj}} \) are denoted as follows:

\[
\begin{align*}
N_{C_{li}} &= \text{RadiusSearch}(M_l, C_{li}, SR) \\
N_{C_{gj}} &= \text{RadiusSearch}(M_g, C_{gj}, SR)
\end{align*}
\]

The edges are the lines between \( C_{li} \) and its neighbor clusters in \( N_{C_{li}} \), and these edges belong to the local cluster edge set \( E_{C_{li}} \). Similarly, \( E_{C_{gj}} \) denotes the cluster edge set between \( C_{gj} \) and its neighbor clusters in \( N_{C_{gj}} \). The expressions are shown as follows:

\[
\begin{align*}
E_{C_{li}} & = \{e_{l1}, e_{l2}, ..., e_{lP}\} \\
E_{C_{gj}} & = \{e_{g1}, e_{g2}, ..., e_{gQ}\}
\end{align*}
\]

The principles for matching two clusters are the same labels and enough matched edges. Then, the matching problem of semantic clusters is transformed into the matching of...
edges in the cluster edge sets. \( C_i \) and \( C_{gj} \) are successfully matched if more than \( N_c \) pairs of edges can be matched in \( E_{C_i} \) and \( E_{C_{gj}} \), \( N_c \) denotes the minimum number of matched edges. Otherwise, they are not matched.

1) Edge Association: For the edge matching process of \( e_i^1 \) in \( E_{C_i} \), the length of each edge \( e_i^j \) in \( E_{C_{gj}} \) are calculated, and the top \( n \) edges whose length are nearest to the length of \( e_i^1 \) are selected as candidate edges of \( e_i^1 \), denoted as \( \{i e_i^1, i e_i^2, ..., i e_i^n\} \), and \( n \) is a constant(default: 5). \((e_i^1, i e_i^j)\) denotes the candidate edge pair, and the distance of the candidate edge pair is \( D_{li}^{ij} \), which will be defined later. Thus, the problem of edge matching is formulated as follows:

\[
\text{Min}(D_{li}^{i1}, D_{li}^{i2}, ..., D_{li}^{ij}, ..., D_{li}^{in}) < \delta_c
\]  

(5)

where \( \delta_c \) is the maximum tolerance distance for successful matching of edge pairs.

For matching \( e_i^1 \) not only the length and label of edge \( e_i^1 \) should be the same, but also the geometric relationships between the edge \( e_i^1 \) and the neighbors of \( e_i^1 \) should also be consistant. Therefore, sub-edge is introduced for verifying the neighbor geometric relationships. The sub-edges of \( e_i^1 \) is the edges in edge set \( E_{C_i} \) except \( e_i^1 \), and composed the sub-edge set \( SE_{E_i} \). Similarly, the edges in \( E_{C_{gj}} \) except \( i e_i^j \) are called sub-edge set \( SE_{gj} \) of \( i e_i^j \):

\[
SE_{E_i} = \{e_i^1, ..., e_i^p, ..., e_i^k\}, p \neq i
\]

\[
SE_{gj} = \{e_i^j, ..., e_i^q, ..., e_i^k\}, q \neq j
\]

(6)

The distance \( D_{li}^{ij} \) of the candidate edge pair \((e_i^1, i e_i^j)\) is defined as the average distances of successfully matched sub-edge pairs between the sub-edge sets \( SE_{E_i} \) and \( SE_{gj} \). And the matching of sub-edge pairs is in the following subsection.

2) Sub-Edge Association: The sub-edge \( e_i^p \) of the matching edge \( e_i^1 \) in \( E_{C_i} \) includes elements \( [d_{li}^p, \theta_{li}^p] \), where \( d_{li}^p \) is the length of \( e_i^p \), and \( \theta_{li}^p \) is the clockwise angle between \( e_i^p \) and \( e_i^1 \). As the same, \( [d_{lj}^q, \theta_{lj}^q] \) denotes the elements of sub-edge \( e_i^q \) in \( E_{C_{gj}} \), where \( d_{lj}^q \) is the length of \( e_i^q \), and \( \theta_{lj}^q \) is the clockwise angle between \( e_i^q \) and \( e_i^1 \). A cluster association example is shown in Fig.4.

Fig. 4. Cluster association diagram. (a) shows the neighbor semantic clusters of \( C_i \) in local semantic cluster map, and (b) shows the neighbors of \( C_{gj} \) in the global cluster map. Brown dots and green dots represent trunks and poles, respectively. Thick solid line with arrows in (a) represents the edge to be matched, and thin solid lines with arrows represent the sub-edges. \( \delta, \theta \) represent the element of sub-edge. The thick dotted lines in (b) indicate the candidate edges of the edge to be matched in (a).

The matching success of two sub-edges \( e_i^a \) and \( e_i^b \) should satisfy four conditions: (a) The semantic cluster labels of two sub-edges are the same; (b) The length error between two sub-edges is under a certain threshold; (c) The angle error between two sub-edges is under a certain threshold; (d) The distance between two sub-edges, denoted as \( d_{li}^{pq} \), is also under a certain threshold. And the distance \( 19 \) is defined as follows:

\[
d_{li}^{pq} = \sqrt{d_{li}^p \cdot d_{li}^q + d_{li}^q \cdot d_{li}^j - 2 \cdot d_{li}^p \cdot d_{li}^j \cdot \cos(\theta_{li}^p - \theta_{li}^q)}
\]  

(7)

The four conditions of sub-edges matching are mathematically formulated as follows:

\[
\begin{align*}
\text{Label}(e_i^a) & = \text{Label}(e_i^b) \\
|d_{li}^a - d_{lj}^b| & < \delta_d \\
|\theta_{li}^a - \theta_{lj}^b| & < \delta_g \\
d_{li}^{pq} & < \delta_se
\end{align*}
\]

(8)

where \( \text{Label()} \) denotes the neighbor cluster label; \( \delta_d \) indicates the threshold of distance error between sub-edges; \( \delta_g \) indicates the threshold of the angle error of sub-edges; \( \delta_se \) indicates the threshold of the distance of sub-edges.

3) Semantic Cluster Association: If more than \( N_{sc} \) sub-edges of the candidate edge pair \((e_i^1, i e_i^j)\) are satisfied with (8), the candidate edge pair is matched successfully. Then, the distance \( D_{li}^{ij} \) of candidate edge pair can be computed as the first line of (9). If the candidate edge pair is not matched, the distance \( D_{li}^{ij} \) is given infinite \( DOUBLE\_MAX \):

\[
D_{li}^{ij} = \begin{cases} 
\text{log}_{kli}^{-1}, & k_{li}^{-1} \cdot \sum_{pq}^{k_{li}^{-1}} k_{ij}^{pq} \cdot d_{li}^{pq} = N_{sc} \\
DOUBLE\_MAX, & \text{otherwise} 
\end{cases}
\]

(9)

where \( k_{lj}^{-1} \) indicates the number of sub-edge pairs successfully matched, \( k_{li}^{-1} \) indicates the number of all sub-edges in \( SE_{E_i} \), and \( N_{sc} \) is a threshold. As shown in (9), the distance \( D_{li}^{ij} \) considered the distances of all the matched sub-edges and the proportion of the matched sub-edges \( k_{ij}^{pq} \) to all sub-edges in \( SE_{E_i} \). Considering these two factors is to avoid the false matching caused by small average distance of sub-edges and the low proportion of successful matched sub-edges pairs.

From (5) and (9), the best candidate edge pair is formulated as follows:

\[
\hat{D_{li}} = \text{Min}(D_{li}^{i1}, D_{li}^{i2}, ..., D_{li}^{ij}, ..., D_{li}^{in}) < \delta_c
\]

(10)

where \( \hat{D_{li}} \) is the distance of the best candidate edge pair, \( \delta_c \) denotes the threshold of the distance.

After matched the candidate edge pairs, the labels and the matched edge pairs numbers of the two semantic clusters to be matched should also be considered. Thus, the matching
of clusters are denoted as follows:

\[
\text{Label}(C_{li}) = \text{Label}(C_{gj}) \quad k_e \geq N_e
\]  

(11)

where \( k_e \) is the matched edges pairs numbers, and \( N_e \) represents the minimum number of edge pairs successfully matched. Algorithm 1 shows the algorithm of semantic cluster association. An matching result example of semantic cluster association is shown in Fig.3(c) and (d).

Algorithm 1: Semantic Cluster Association

**Input:** Two clusters that need to be matched \( C_{li}, C_{gj} \) and their respective cluster map \( M_l, M_g \)

**Output:** Result of the two clusters associated True/False

1. if \( \text{Label}(C_{li}) \neq \text{Label}(C_{gj}) \) then
2. \quad return False;
3. end

4. Search nearest neighbor clusters and get neighbor edges \( E_l, E_g \) of \( C_{li}, C_{gj} \) by (3) and (4).

5. \( k_e \leftarrow 0 \)

6. for \( e^l_i \in E_l \) do
7. \quad candidates = FindNEdgePairCandidates\( (E_g, e^l_i) \)
8. \quad for \((e^g_j, e^g_k) \in \text{candidates} \) do
9. \quad \quad \( k_{se} \leftarrow 0 \)
10. \quad \quad for \( e^g_p \in SE_{e^g_j} \) do
11. \quad \quad \quad \text{Compute sub-edge matching result } \text{IsSubEdgeMatch} \text{ by (8) and (7)}
12. \quad \quad \quad if \text{IsSubEdgeMatch is True then}
13. \quad \quad \quad \quad \quad \quad k_{lj} \leftarrow k_{se} + 1
14. \quad \quad \quad end
15. \quad \quad end
16. \quad end
17. Compute edge pair distance \( D^{gj}_{li} \) by (9)
18. end

19. if \( \text{IsEdgeMatch} \) is True then
20. \quad \( k_e \leftarrow k_e + 1 \)
21. end
22. end
23. return \( k_e \geq N_e \) (((11)))

D. Long-term Relocalization

Algorithm 1 is utilized to match the semantic clusters in the local map with the global map, and provides \( n \) pairs of semantic cluster matched pairs. Let \( c_i = (C_{li}, C_{gj}) \) denote a matched pair of semantic clusters, where \( C_{li} \in M_l \) and \( C_{gj} \in M_g \). Semantic cluster matching pairs obtained from semantic cluster association algorithm are coarse correspondences. Thus, a geometric consistency method [20] will be used to eliminate the false positive matched pairs, and finally fine correspondences are remained.

For two matched pairs \( (c_i, c_j) \), if their euclidean distance error in the local and global map is under a certain threshold, \( (c_i, c_j) \) is geometric consistent, formulated as follows:

\[
|d_l(c_i, c_j) - d_g(c_i, c_j)| \leq \epsilon \quad (12)
\]

where \( \epsilon \) indicates the maximum tolerance euclidean distance error; \( d_l \) represents the euclidean distances of \( C_{li} \) and \( C_{gj} \) in \( M_l \), and \( d_g \) represents the euclidean distances of \( C_{gj} \) in \( M_g \).

Then, the 6-DOF pose transformation between the local map \( M_l \) and the global map \( M_g \) is computed by matching the corresponding semantic cluster centroid points and semantic cluster point clouds. Firstly, the 3D centroid points pairs are extracted from semantic cluster matched pairs, and the RANSAC [21] algorithm is used to filter out wrong matched pairs. Secondly, the coarse coordinate transformation matrix \( T_{gl} \) from the local map to the global map is calculated by using iterative closest point (ICP) algorithm with (13).

\[
T_{gl} = \arg \min_{T_{gl}} \frac{1}{2} \sum_{k=1}^{k} \|O_{li} - T_{gl}O_{gj}\|^2
\]  

(13)

where \( k \) is the number of the filtered semantic cluster matched pairs. \( O_{li} \) and \( O_{gj} \) are the 3D centroid points of the clusters \( C_{li} \) and \( C_{gj} \).

Thirdly, the local and global point clouds of the matched cluster pairs are respectively extracted from cluster manager. Using the coarse pose transformation \( T_{gl} \) as the initial value, the fine transformation matrix \( \hat{T}_{gl} \) is obtained by the ICP algorithm again with the extracted point clouds instead of the 3D centroid points, shown as follows:

\[
\hat{T}_{gl} = \arg \min_{T_{gl}} \frac{1}{2} \sum_{k=1}^{k} \|P_{li} - \hat{T}_{gl}P_{gj}\|^2
\]  

(14)

where \( P_{li}, P_{gj} \) are the extracted point clouds of the clusters \( C_{li} \) and \( C_{gj} \); \( \hat{T}_{gl} \) represents the global pose of the local map in the global map.

E. Long-term Localization

Fig. 5. long-term localization process. \( \{W\} \) denotes the world frame, \( \{L\} \) denotes the LiDAR frame.

The long-term localization algorithm is shown in the Fig.5, the global pose \( \hat{T}_{gl} \), stably output at 2Hz by the relocalization module, is used to correct the accumulated drift of the LiDAR odometry in real time. The global pose \( \hat{T}_{gl} \) at \( t_k \) is re-expressed as \( T^{WL}_{k} \). The LiDAR odometry outputs \( n \) poses continuously from \( t_k \) to \( t_{k+n} \) before the next global pose is updated, denoted as \( \{T^L_{k}, T^L_{k+1}, ..., T^L_{k+i}, ..., T^L_{k+n-1}\} \). During \( t_k \) to \( t_{k+n} \), the global pose estimation of the unmanned vehicle is calculated as follows:
\[ T_{k+i}^{WL} = T_{k}^{WL} \prod_{j=0}^{i} T_{k+j}^{L} \]  

where \( i \in [0, n - 1] \). So far, long-term localization has been finished by correcting the drift of LiDAR odometry with the real-time global pose of relocalization module.

V. EXPERIEMENTS

To evaluate the proposed relocalization and localization algorithms in long-term scenarios, several experiments were performed on the self-made campus dataset. It is worth noting that Carlevaris-Bianco [22] proposed NCLT dataset in long-term scene, but the Velodyne32 LiDAR in NCLT dataset is installed upside down, which makes the actual LiDAR point cloud sparsity equal to Velodyne16 LiDAR. While the proposed method is based on semantic segmentation of dense LiDAR point clouds, NCLT dataset cannot meet our requirements. Therefore, the proposed method is compared on the self-made dataset with the SCI algorithm [12], which is the state-of-the-art long-term LiDAR-based localization system.

A. Benchmark Dataset

![Fig. 6. Dataset Environment. The left image is several months earlier than the right image, the differences are the removed cars, the trees with different shapes, and the lighting conditions.](image)

The challenge of long-term localization is the environmental changes. So the self-made dataset recorded the data in the university town environment with high traffic volume and dense vegetation. The environment is shown in Fig.6.

![Fig. 7. Example of annotated semantic points.](image)

To use RangeNet++ to infer the semantic label, the self-made dataset is annotated with more than 1500 frames of 3D point clouds generated by LiDAR scanning, which covered a geographic area of 76,000 \( m^2 \) and included 120 million three-dimensional semantic points. And the marked labels with 10 categories included buildings, highways, sidewalks, bicycles, green plants, poles, tree trunks, vehicles, pedestrians, and other objects. An example of the annotated semantic point clouds by the labeling tool PointLabeler [23] is shown in Fig.7. The organization of the semantic point clouds is labeled and organized according to the SemanticKitti [23]. Four datasets were collected, and Table I showed the details of datasets. Although the datasets were collected within a month, some places of the environment have changed significantly due to the movement of cars and growing of trees; furthermore, different numbers of clusters in the global map were randomly removed to simulate the significant changes of environment. We will also continue collect more datasets and update the results in the future.

| Date       | Length(km) | Time      | Weather |
|------------|------------|-----------|---------|
| 2020-10-12 | 1.31       | Afternoon | Sunny   |
| 2020-10-18 | 1.28       | Morning   | Cloudy  |
| 2020-10-25 | 1.53       | Evening   | Cloudy  |
| 2020-11-05 | 1.4        | Afternoon | Sunny   |

B. Evaluation of Long-term Relocalization

To evaluate the success rate of relocalization in long-term scenarios, the earliest dataset 2020-10-12 was used to build the global semantic cluster map, and the other three datasets were used to create the local semantic cluster maps for relocalization with the global map. The definition of successful relocalization is expressed as follows:

\[ \| t_{\text{est}} - t_{\text{gt}} \| < \delta_{\text{relocalization}} \]  

where \( t_{\text{est}} \) and \( t_{\text{gt}} \) represent the estimated position of relocalization and the ground truth by GNSS; \( \delta_{\text{relocalization}} \) denotes the maximum tolerance distance between the estimated position and the ground truth when the relocalization is successful, and set to 10 m.

The success rate of relocalization of the SCI algorithm are divided into two categories, namely the success rate of the highest(top 1) scoring frame and the top 5 scoring frames. 120 positions were selected for the relocalization experiments on each dataset, and distributed as shown in Fig.8. And the parameters of clusters registration and association are shown in Table II. And the success rate of relocalization was shown in Table III. The left number of each cell represents the number of successful relocalizations in 120 experiments, and the right number of each cell represents the success rate. The results showed the success rate of the proposed relocalization algorithm higher than the SCI algorithm on the self-made dataset.

Moreover, to verify how far unmanned vehicles have travelled when relocalization is successful, the travelled distance of successful relocalization were collected and sorted from 116 experiments, and the probability curve result is shown
in Fig. 9(a). When the unmanned vehicles travelled 24 m, the relocalization is done successfully with the probability of 90%; And with the success probability of 95% and 99% when the unmanned vehicles travelled 30 m and 43 m. The result illustrated that, by using the proposed relocalization algorithm in the long-term scenarios, the travelled distance of successful relocalization is relatively short compared with the scale of the whole map (almost 1.4 km).

To evaluate the performance of the proposed relocalization algorithm under large changes in the environment, different numbers clusters in the global semantic map were randomly selected to associate with local semantic cluster maps. Specifically, 827 semantic clusters were extracted from the built semantic map using the dataset 2020-10-12. When using the dataset 2020-11-05 for relocalization, a series of sub-global semantic cluster maps were randomly created by retaining 100%, 80% and 60% of the total clusters of the global map by dataset 2020-10-12, i.e., 827, 661 and 496 clusters retained, respectively. And the relocalization experiments were compared between above three sub-global maps with different cluster numbers by the dataset 2020-11-05. The number of semantic clusters of per meter along the unmanned vehicle trajectory is defined as the semantic clustering density $\rho$, formulated as follows:

$$\rho = \frac{N}{L}$$

(17)

where $N$ denotes the number of semantic clusters used in global map, and $L$ denotes the travelled distance.

For the three sub-global maps, the semantic cluster densities are calculated by (17) to be 0.59, 0.473 and 0.354 Clusters/m, respectively. 116, 68, and 62 times of the experiment were performed on the dataset 2020-11-05 to calculate the travelled distance when successfully relocated, the result was shown in Fig. 9(b). It can be seen that the larger density of semantic cluster map brings the shorter travelled distance. And when the semantic clustering density is not less than 0.473 Clusters/m, the unmanned vehicle only needs to travel no more than 50 m when the relocalization is successful with probability 90%.

C. Long-term Localization Analysis

The evaluation of long-term localization experiment is mainly to compare the position estimation accuracy of the LiDAR odometry. The dataset 2020-10-05 was used to build the global semantic cluster map, and the other three datasets were used to create local semantic cluster maps for long-term localization. The metrics of Root Mean Square Error (RMSE) is used for evaluation, and the results of the proposed long-term localization system and LOAM are compared with the ground truth. The results were shown in Table IV. The proposed approach does not need to utilize the IMU data for relocalization and localization, and therefore it was compared with the LOAM but not LIO-SAM.

![Graph](image)

Fig. 8. The positions distribution of the relocalization experiments. Red squares represent the selected positions.

Fig. 9. (a) The result of relocalization success probability vs the travelled distance. The cyan solid curve denotes the relocalization success probability, and the dot lines denote different probability metrics. (b) The result of relocalization success probability vs the travelled distance under different cluster densities. $C/m$ is the unit of density, denoting the number of semantic clusters of per meter.

### TABLE II

| Parameter | Description | Setting |
|-----------|-------------|---------|
| SR        | search radius | 50 m    |
| $\delta_d$ | threshold of subedge distance error | 0.3 m |
| $\delta_a$ | threshold of subedge angle error | 10$^\circ$ |
| $\delta_{se}$ | threshold of subedge distance | 0.2 m |
| $\delta_c$ | threshold of subedge distance | 0.25 m |
| $N_{se}$ | minimum matched pairs of subedges | 5 |
| $N_e$ | minimum matched pairs of edges | 5 |

### TABLE III

| Dataset | SCI(TOP1) | SCI(TOP5) | Proposed |
|---------|-----------|-----------|----------|
| 2020-10-18 | 99/82.5% | 114/95.0% | **120/100.0%** |
| 2020-10-25 | 97/80.8% | 108/90.0% | **116/96.6%** |
| 2020-11-05 | 103/85.8% | 112/93.3% | **118/98.3%** |

### TABLE IV

| Dataset | LOAM(m) | Proposed(m) |
|---------|---------|-------------|
| 2020-10-18 | 42.41 | **0.29** |
| 2020-10-25 | 44.32 | **0.34** |
| 2020-11-05 | 40.48 | **0.24** |
Since LOAM only relies on the LiDAR odometry, the RMSE was relatively large because of the accumulated error of the odometry. For the proposed method, the relocalization algorithm based on the semantic cluster map was used to correcting the accumulated error, and the position estimation between frames of the laser odometry is accurate, and therefore the RMSE of the proposed long-term localization system is smaller. The trajectories of ground truth, LOAM and the proposed method are shown in Fig.10(a) by using the dataset 2020-11-05 as an example, and the x,y,z position error are shown in Fig.10(b).

![Image](image_url)

Fig. 10. (a) Trajectories of the ground truth, LOAM and the proposed long-term localization system. (b) Position errors along x-y-z axes within a time window.

VI. CONCLUSION

A novel relocalization method based on semantic cluster map is proposed to realize high accuracy relocalization and real-time localization in long-term environments. A robust semantic cluster map is built by extracting pole-like objects from raw 3D LiDAR points to resolve the long-term challenge. A semantic cluster association algorithm on the basis of geometric consistency verification is proposed to obtain the pose transformation between the local cluster map and the global cluster map. And a long-term and real-time localization system is developed based on the relocalization module and LiDAR odometry. Finally, several experiments are performed to illustrate the effectiveness of the proposed approach. In the future work, we will extend to utilize more objects in long-term environments and make it more general rather than relying heavily on pole-like objects.

REFERENCES

[1] Raúl 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.
[2] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In Robotics: Science and Systems, volume 2, 2014.
[3] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4758–4765. IEEE, 2018.
[4] François Pomerleau, Philipp Krüsi, Francis Colas, Paul Furgale, and Roland Siegwart. Long-term 3d map maintenance in dynamic environments. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 3712–3719. IEEE, 2014.
[5] W. Churchill and P. Newman. Experience-based navigation for long-term localisation. The International Journal of Robotics Research, 32:1645 – 1661, 2013.
[6] Y. Ye, Titus Cieslewski, Antonio Loquercio, and D. Scaramuzza. Place recognition in semi-dense maps: Geometric and learning-based approaches. In BMVC 2017, 2017.
[7] G. Kim and A. Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4802–4809, 2018.
[8] R. Dubé, D. Dugas, E. Stumm, J. Nieto, R. Siegwart, and C. Cadena. Segmatch: Segment based place recognition in 3d point clouds. In 2017 IEEE International Conference on Robotics and Automation (ICRA), pages 5266–5272, 2017.
[9] A. Schaefer, D. Büscher, J. Vertens, L. Luft, and W. Burgard. Long-term urban vehicle localization using pole landmarks extracted from 3-d lidar scans. In 2019 European Conference on Mobile Robots (ECMR), pages 1–7, 2019.
[10] Mans Larsson, Erik Stenborg, Lars Hammarstrand, Marc Pollefeys, Torsten Sattler, and Fredrik Kahl. A cross-season correspondence dataset for robust semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9532–9542, 2019.
[11] Tim Caselitz, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Matching geometry for long-term monocular camera localization. In ICRA Workshop: AI for long-term Autonomy, 2016.
[12] Giseop Kim, Byungjae Park, and Ayong Kim. 1-Day Learning, 1-Year Localization: Long-Term LiDAR Localization Using Scan Context Image. IEEE Robotics and Automation Letters, 4(2):1948–1955, 2019.
[13] Xin Kong, Xue meng Yang, Guangyao Zhai, Xiangrui Zhao, Xianfang Zeng, Mengmeng Wang, Yong Liu, Wanlong Li, and Fen Wen. Semantic graph based place recognition for point clouds. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 8216–8223. IEEE, 2020.
[14] S. Ratz, M. Dynczyk, R. Siegwart, and R. Dubé. Oneshot global localization: Instant-visual pose estimation. 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 5415–5421, 2020.
[15] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss. RangeNet++. Fast and Accurate LiDAR Semantic Segmentation. In IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS), 2019.
[16] I. Bogoslavskyi and C. Stachniss. Fast range image-based segmentation of sparse 3d laser scans for online operation. In Proc. of The International Conference on Intelligent Robots and Systems (IROS), 2016.
[17] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Rus Daniela. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5135–5142. IEEE, 2020.
[18] R.B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1–22, 2011.
[19] Zhichen Pan, Haoyao Chen, Silin Li, and Yunhui Liu. ClusterMap Building and Relocalization in Urban Environments for Unmanned Vehicles. pages 1–22, 2019.
[20] Mattia G Gollub, Renaud Dubé, Hannes Sommer, Igor Gilitschenski, and Roland Siegwart. A partitioned approach for efficient graph–based place recognition. In Proc. IEEE/RSJ Intl. Conf. Intell. Robots Syst./Workshop Planning, Perception Navigat. Intell. Veh., 2017.
[21] Konstantinos G Derpanis. Overview of the ransac algorithm. The International Journal of Robotics Research, 4(1):2–3, 2010.
[22] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. Building and Relocalization in Urban Environments for Unmanned Vehicles. pages 1–22, 2019.
[23] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyril Stachniss, and Jurgen Gall. Semantic kitti: A dataset for semantic scene understanding of lidar sequences. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 9297–9307, 2019.