https://doi.org/10.20965/jrm.2021.p1385

Paper:

Visual SLAM Framework Based on Segmentation with the Improvement of Loop Closure Detection in Dynamic Environments

Leyuan Sun[*,**,***], **Rohan P. Singh**[*,**,***], **and Fumio Kanehiro**[*,**,***]

[*]Department of Intelligent and Mechanical Interaction Systems, Graduate School of Science and Technology, University of Tsukuba
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577 Japan
[**]CNRS-AIST JRL (Joint Robotics Laboratory), International Research Laboratory (IRL)
1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan
[***]National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan
E-mail: {son.leyuansun, rohan-singh, f-kanehiro}@aist.go.jp
[Received March 12, 2021; accepted June 18, 2021]

Most simultaneous localization and mapping (SLAM) systems assume that SLAM is conducted in a static environment. When SLAM is used in dynamic environments, the accuracy of each part of the SLAM system is adversely affected. We term this problem as dynamic SLAM. In this study, we propose solutions for three main problems in dynamic SLAM: camera tracking, three-dimensional map reconstruction, and loop closure detection. We propose to employ geometry-based method, deep learning-based method, and the combination of them for object segmentation. Using the information from segmentation to generate the mask, we filter the keypoints that lead to errors in visual odometry and features extracted by the CNN from dynamic areas to improve the performance of loop closure detection. Then, we validate our proposed loop closure detection method using the precision-recall curve and also confirm the framework's performance using multiple datasets. The absolute trajectory error and relative pose error are used as metrics to evaluate the accuracy of the proposed SLAM framework in comparison with state-of-the-art methods. The findings of this study can potentially improve the robustness of SLAM technology in situations where mobile robots work together with humans, while the object-based point cloud byproduct has potential for other robotics tasks.

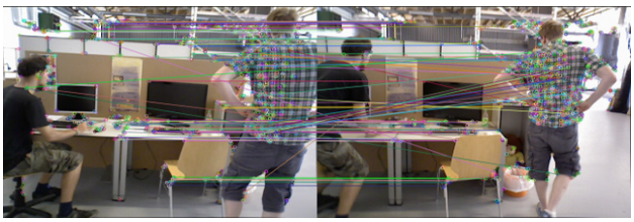
Keywords: visual SLAM, dynamic environment, loop closure detection

1. Introduction

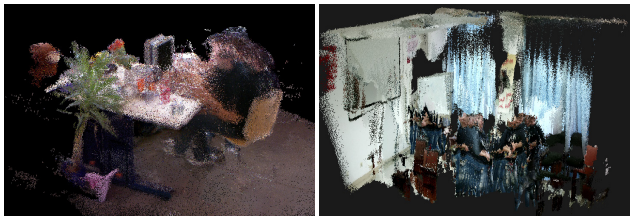
Nowadays, mobile robots can rely on simultaneous localization and mapping (SLAM) technology for autonomous travel under favorable conditions [1, 2], however, several issues must be addressed when SLAM technology is used in robotics applications in a dynamic environment. Most visual odometry calculations in a visual SLAM framework are based on keypoint tracking, which is based on the strict assumption that the positions of the extracted keypoints are constant in a global sense. In other words, all objects in the frame must be static and rigid. When keypoints are extracted from a moving human, as shown in **Fig. 1(a)**, these keypoints participate in camera pose estimation, which introduces outliers into the system. In severe cases, the system tends to even lose track of the camera. Another issue that must be considered is that the point cloud corresponding to dynamic objects exists in the final dense point cloud, as shown in **Fig. 1(b)**, even if they are projected into correct global positions. This is not useful for subsequent tasks such as navigation.

Loop closure detection (LCD) is an important component of SLAM. It affects the accuracy of an estimated trajectory and global map consistency over a long period of time. Because LCD provides the correlation between the current data and all historical data, it also helps in the re-localization of a SLAM system when the tracking is lost. Therefore, it improves the accuracy and robustness of the entire SLAM system. However, when dynamic objects occur, LCD becomes more difficult because these objects add new features and occlude some original features, as shown in the left-hand illustration of **Fig. 1(c)**. Although many dynamic SLAM frameworks based on ORB-SLAM2 [3] have been proposed, most of them do not focus on the problem of LCD in dynamic environments.

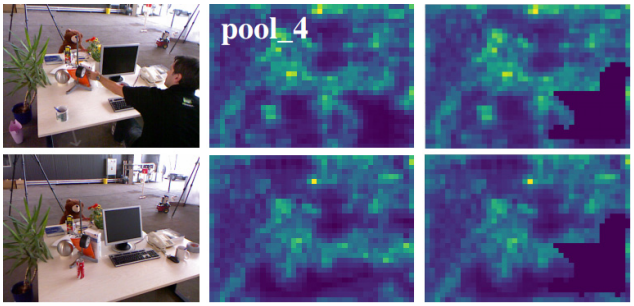
This paper presents different segmentation methods to meet different requirements, which are the main contributions of our previous work [4]. Moreover, we use the VGG-16 [5], which is a simple architecture with high accuracy in the object recognition task [6], and it is also used for the convolution processing of the deep-learning based segmentation FCN-VGG16 [7] in our proposed framework. By comparing each layer's output, we found that the pooling 4 layer of VGG-16 provides maximum ac-



(a) Example of an incorrect keypoint matching result in a dynamic environment.



(b) Point clouds corresponding to dynamic objects remaining in the map.



(c) Dynamic objects make loop closure detection difficult, as shown in the first column. Heatmaps generated using a CNN without and with union masks, as shown in the second and third columns, respectively.

Fig. 1. Problems observed when SLAM is used in a dynamic environment.

curacy for feature extraction in our task. In addition, we manually generated dynamic loop closure datasets for testing and used the precision-recall curve as a quantitative evaluation metric.

Another contribution of our previous work [4] is dense point cloud generation with static objects. Because we have used semantic segmentation in this study, we also added an object-oriented point cloud mapping function. In addition, we developed a graphical user interface (GUI) that helps the user to extract the point cloud information of each object to facilitate subsequent tasks such as pose estimation and object manipulation.

The remainder of this paper is structured as follows. Section 2 discusses the work related to the dynamic SLAM and LCD. Section 3 describes the proposed visual SLAM framework and details of the proposed convolutional neural network (CNN)-based feature extraction with a union mask for the loop closure detection problem in a dynamic environment. Section 4 presents the experimental results, including the loop closure experiments and the accuracy of the entire SLAM system on different types of datasets. Finally, the conclusions are presented in Section 5.

The main contributions of this work are as follows:

- We propose and validate a CNN-based feature extraction with the union mask method for loop closure detection in a dynamic environment, not only using the precision-recall curve but also on the SLAM framework.
- We propose an integrated SLAM framework that can solve three main problems in dynamic SLAM: camera tracking, removal of dynamic objects from scene point clouds, and loop closure detection.

2. Related Work

2.1. Dynamic SLAM

Most dynamic SLAM systems solve the camera-tracking problem through dynamic object segmentation. Dynamic objects can be categorized into moving objects, such as humans and animals, and movable objects, such as a chair, which can be moved by a human. Objects in these two categories are usually segmented using deep learning-based methods and geometry-based methods [4].

- Xiao et al. [8] used a single-shot multi-box detector (SSD) [9] to detect a human for post-processing keypoint filtering. The shape of the mask is a bounding box that causes valid keypoints near the dynamic objects to be filtered. This process may cause a large amount of useful information to be lost, especially when moving objects occupy the bulk of the image.
- To determine which pixels belong to the dynamic objects, Wang et al. [10] and Cheng et al. [11] proposed methods to use the multi-view geometry relationship represented by the fundamental matrix and essential matrix, respectively. These matrices are calculated before dynamic object segmentation; hence, these methods assume that most pixels of the image correspond to a static scene. In addition, the back ends of these frameworks do not address the point cloud map reconstruction problem.
- DynaSLAM [12] combines deep learning-based segmentation with geometry-based segmentation and provides a dense point cloud map that excludes dynamic objects. However, the segmentation method Mask R-CNN [13] that was adopted is not a real-time method, which limits the scenarios in which DynaSLAM can be used.
- DS-SLAM [14] uses SegNet [15] as a semantic segmentation structure and a fundamental matrix-based moving object consistency check to select point clouds corresponding to moving objects after semantic segmentation. However, unlike our framework, the system does not generate a static point cloud.

Moreover, none of the aforementioned frameworks consider the problems associated with LCD in dynamic environments.

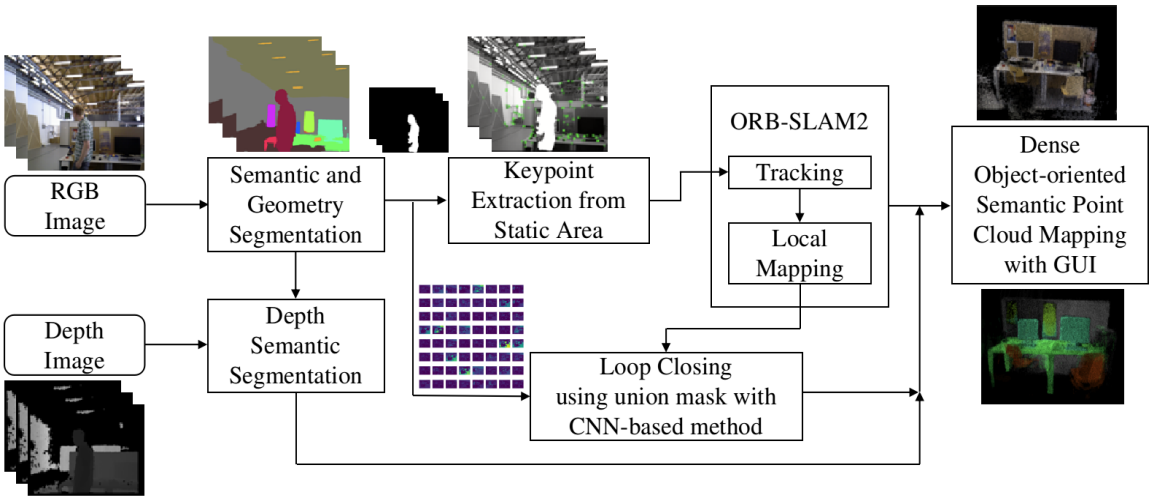


Fig. 2. Overview of the framework. Rectangles: processing; rounded rectangles: data.

2.2. LCD

In general, there are two ways to extract features for LCD. One method uses manually designed features such as ORB, SIFT, and SURF, while the other uses CNNs to extract features.

- FAB-MAP [16] is similar to Bag of Words (BoW) [17] with the manually designed feature SURF used to perform place recognition. The experimental results showed that FAB-MAP is sufficiently fast to perform online LCD in mobile robotics. However, its limitations are similar to those of the BoW model, which means that it is not robust to the environments where moving objects exist. ORB-SHOT SLAM [18] improves the traditional BoW performance using 3D SHOT descriptors to describe the ORB corners and train a 3D vocabulary based on Bag-of-Visual-Words (BoVW). Although better trajectory estimation results were achieved compared with ORB-SLAM2, the dynamic environment situation was not considered in the experiments.
- Ref. [19] uses pre-trained AlexNet to extract features and compare the performance of each CNN layer. Sünderhau et al. found that the mid-layer of the network was the most robust for the place recognition task. In addition, they showed that the performance of the traditional BoW was worse than that of the CNN-based feature extraction method. They also found that the mid-layer was the most robust layer for the place recognition task. However, [19] and other similar related works [20–24] did not focus on the dynamic environment, which is the main difference between from our research.

Most importantly, none of these studies evaluated the performance of their methods with a SLAM system in dynamic environments.

3. SLAM Framework

3.1. Framework Overview

An overview of the proposed SLAM framework is shown in **Fig 2**. The differences with the original ORB-SLAM2, which is the basic framework of this study, are as follows:

1. Uniformly distributed keypoints are extracted only from static areas.
2. A dense RGB static object point cloud map or the object-oriented semantic map can be generated using the GUI.
3. A CNN-based method for features extraction with a union mask, especially for the LCD thread in a dynamic environment, is implemented.

3.2. Segmentation

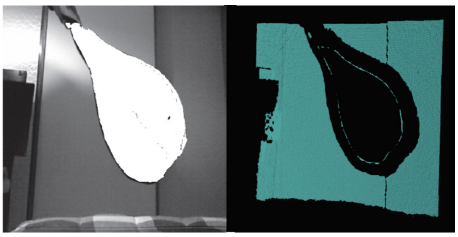
3.2.1. Semantic Segmentation

We used a semantic segmentation framework FCN-VGG16 [7] that was trained on the dataset presented in [25]. This segmentation model can identify up to 150 classes of objects in daily life, and the classes for dynamic objects must be specified in advance. In this study, we consider that a human is the only dynamic object in most environmental scenes. We mask the regions segmented out by the trained network in the "Human" class by marking the corresponding pixels in white.

3.2.2. Geometry-Based Segmentation

We also employed geometry-based segmentation (GS), which was proposed in our previous work [4]. Compared with deep learning-based methods, the advantage of GS is that it can identify moving objects, as shown in **Fig. 3(b)**. A limitation of this method is that it is a homography-based method, and the accuracy of the homography matrix depends on whether the points used for its computation lie on a planar surface, as shown in **Fig 3(a)**. In

(a) Continuous walking human segmentation, the background is a planar surface.



(b) Point cloud of waving tennis racket segmentation, with the segmented image and its point cloud.

Fig. 3. Example of geometry-based segmentation.

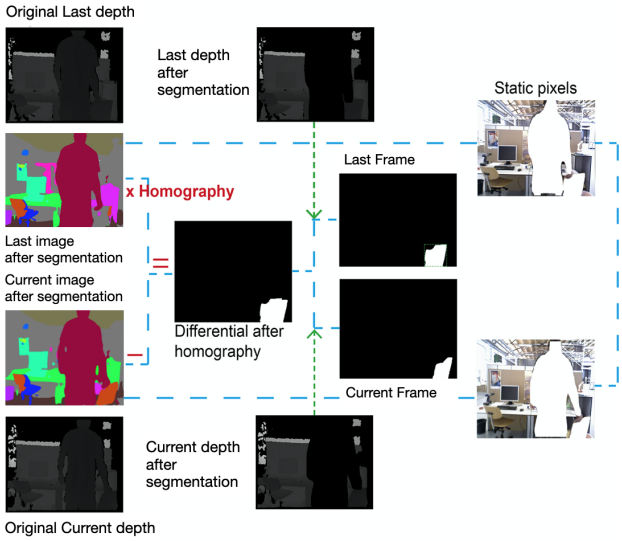


Fig. 4. Semantic segmentation + GS method.

addition, the points used to compute the homography matrix should belong to static objects, which means that the dynamic objects should not occupy a large part of the image.

3.2.3. Semantic Segmentation Fused with GS

The combination of a semantic segmentation method and the GS method is suitable for many situations, for example, a situation in which a human pulls a chair from a desk. As shown in **Fig. 4**, the human is first segmented by the semantic segmentation network, and then the GS detects the chair when it is moving. Because we already have a mask image that filters out the pixels belonging to the human, we can use the pixels in the static area to calculate the homography. This improves the accuracy of the homography matrix, especially when moving humans occupy most of the view in adjacent frames. We do not pre-define the chair as a movable object that can be segmented through semantic segmentation because the chair is only moving for a short duration. Most of the time, the chair is static in this dataset. It is difficult to pre-define all objects that have the potential to move, therefore, this combined segmentation is suitable for obtaining a relatively complete static dense point cloud map in a dynamic environment.

3.3. Uniformly Distributed Keypoint Extraction and Matching with Dilated Mask

As mentioned in [4], uniformly distributed keypoints (**Figs. 5(b)–(d)**) can improve the accuracy of SLAM, especially when bad illumination and blurred image situations occur, as shown in **Fig. 5(a)**. After using the dilated mask, we can solve the first problem of dynamic SLAM, that is, difficult camera tracking due to keypoints extracted from dynamic objects, as shown in **Fig. 6**.

3.4. LCD

3.4.1. LCD in Dynamic Environments

In ORB-SLAM2, the BoW model is used to calculate the similarity between two frames. The purpose of BoW is to describe a frame according to "what type of features are within it." These features are also called "words," and many words together form a dictionary. A frame can be converted into a vector of words using a dictionary. The similarity between two frames is then obtained by calculating the distance between the two vectors.

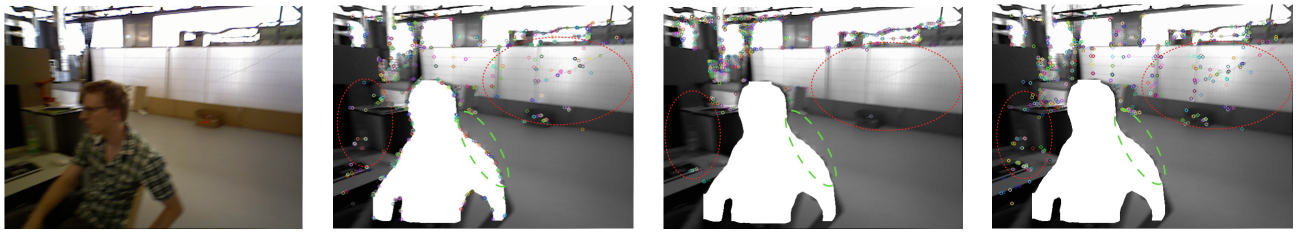
In a static environment, the BoW is a robust and effective method [17]. However, when moving objects exist in the environment, the new words extracted from dynamic objects and the words hidden by the dynamic objects tend to have a negative effect on the calculation of the similarity score in the LCD.

Moreover, many studies [19–24] have confirmed that using a CNN to extract features is more robust and accurate than manually handcrafted features. Considering the above, we propose a CNN-based feature extraction method with a union mask for LCD in a dynamic environment.

3.4.2. Proposed Method for LCD in a Dynamic Environment

The main idea is similar to the solution for the first problem, that is, using only the features from the static area. Because we use the FCN-VGG16 semantic segmentation model, to fully exploit this architecture, we compare the output of each layer in its convolutional part, VGG-16. The dimensionality of each layer used is shown in **Table 1**.

Figure 7 shows the process of the proposed method used to calculate the similarity score between the two images. We use the first layer (pooling 1) as an example to explain the details as follows.



(a) Original image with bad illumination and blur before segmentation.

(b) Keypoints extracted using the ORB-SLAM2 extractor after segmentation.

(c) Keypoints extracted using the OpenCV extractor after segmentation.

(d) Keypoints extracted using the uniformly distributed extractor with a dilated mask.

Fig. 5. Comparison of the keypoint extractors in the proposed framework and ORB-SLAM2. The green ellipses indicate contours with no keypoints, while the red ellipses represent regions with uniformly distributed keypoints.

Fig. 6. Keypoints matching a with dilated mask.

Table 1. Dimensionality of each layer in VGG-16.

Layer	Dimensions	Layer	Dimensions
Pooling layer 1	$240 \times 320 \times 64$	Pooling layer 5	$15 \times 20 \times 512$
Pooling layer 2	$120 \times 160 \times 128$	Fully connected layer 6	4096×1
Pooling layer 3	$60 \times 80 \times 256$	Fully connected layer 7	4096×1
Pooling layer 4	$30 \times 40 \times 512$	Softmax layer 8	1000×1

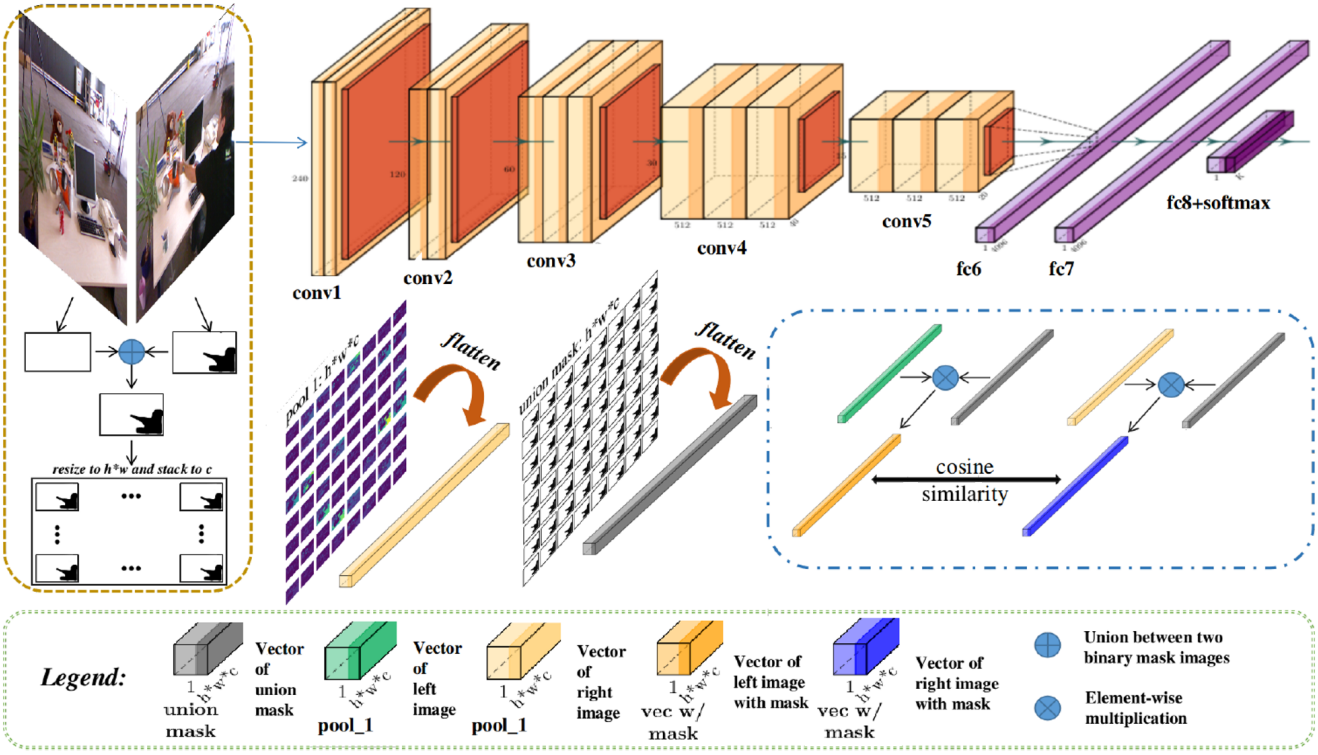


Fig. 7. Proposed CNN-based method for LCD in dynamic environment.

Fig. 8. Union mask generation.

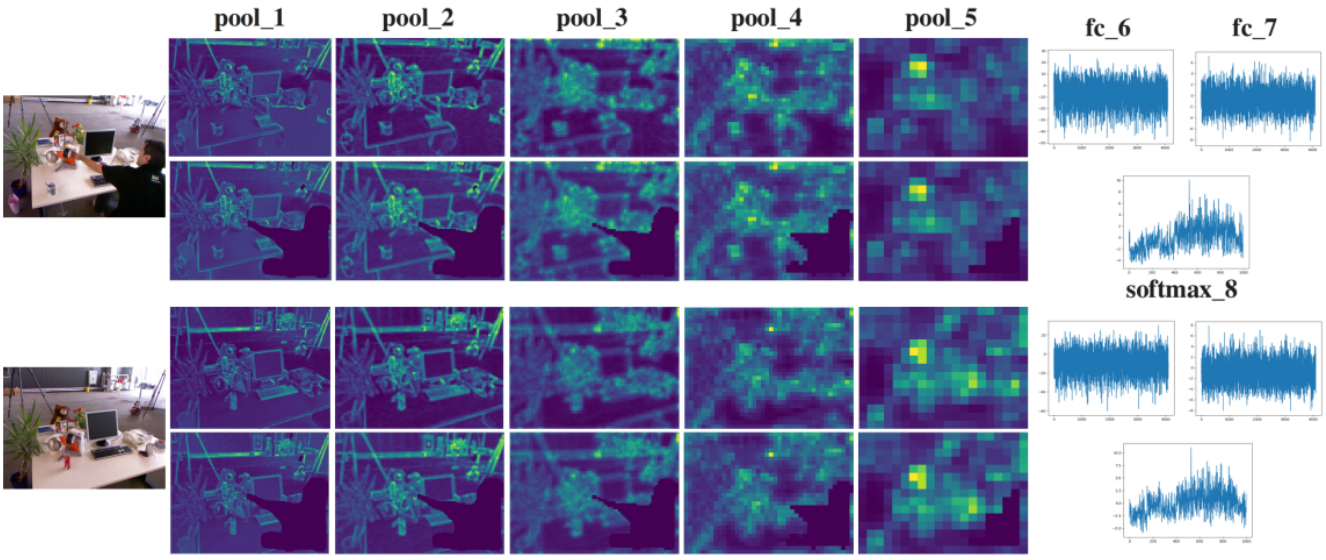


Fig. 9. Heatmaps of different layers used in VGG-16. Here, the first row is the original heatmaps, while the second row was generated using our proposed method.

1. The input image size is 640×480, and the binary mask image is obtained from semantic segmentation (0 is the human area, 1 is the static area).

2. Union processing is performed between the two binary mask images, as shown in **Fig. 8**.

3. The union mask image is resized to 240×320 and stacked into 64 channels. This is then flattened as the union mask vector (the 1st vector in the legend of **Fig. 7**).

4. The feature maps are extracted from the pooling 1 layer with a size of $240 \times 320 \times 64$, and then flattened as a vector (the 2nd and 3rd vectors in the legend of **Fig. 7**).

5. Element-wise multiplication is performed between the union mask vector and the two image vectors from the previous step. Then, we obtain two image vectors, as shown by the 4th and 5th vectors in the legend of **Fig. 7**.

6. The cosine similarity between two vectors is calculated using Eq. (1), which is a well-known distance function commonly used in CNN-based place recognition tasks [19–24] between two vectors.

To visualize the features extracted by each layer, we summed all the feature maps in that layer to produce one heatmap. From the initial layers shown in **Fig. 9**, we can see that the CNN focuses on the boundary information of the objects, and the deeper the CNN layers, the more abstract features it learns.

A limitation of this method is that in the case where the dynamic objects occupy a large fraction of the image, significant information will be lost. Even in the case of humans, using very little information to calculate the similarity is a very challenging task. In addition, element-wise multiplication with a binary mask vector will insert 0 elements in the final two vectors at the same position, increasing the similarities to some extent. However, the impact of this limitation depends on how different the unmasked areas of the images are, and the extent to which the area is occupied by the dynamic objects.

$$cosine(\mathbf{vec}_1, \mathbf{vec}_2) = \frac{\mathbf{vec}_1 \cdot \mathbf{vec}_2}{\|\mathbf{vec}_1\| \cdot \|\mathbf{vec}_2\|} \quad \ldots \quad (1)$$

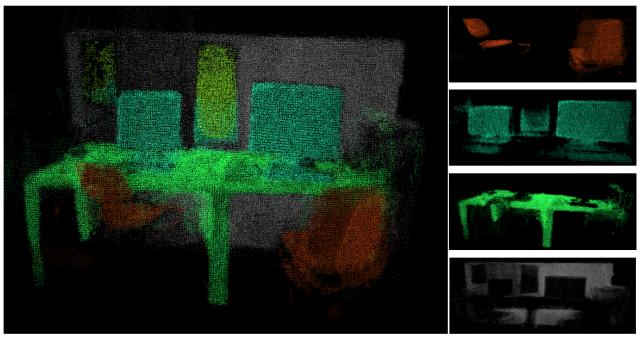
3.5. Dense Point Object-Oriented Point Cloud Generation with a GUI

The formula for calculating a 3D point cloud from a 2D color image and depth image is as follows:

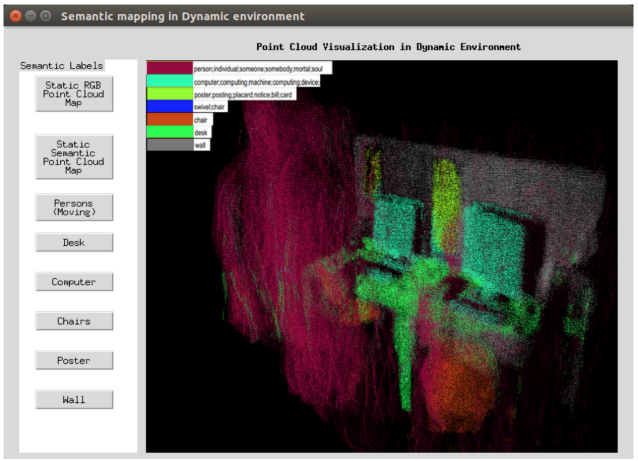
$$\begin{cases} z = \dfrac{d}{s} \\ x = (u - c_x) \cdot \dfrac{z}{f_x} \\ y = (v - c_y) \cdot \dfrac{z}{f_y} \end{cases} \quad \ldots\ldots\ldots\ldots \quad (2)$$

Here, u and v represent the coordinates in the frame coordinate system, whereas x, y, and z represent the coordinates in global coordinate system. Further, f_x, f_y, c_x, c_y, and s are the camera parameters. The construction of the point cloud map requires the projection of multiple frames from different perspectives in the environment. To avoid redundancy in the 3D point cloud and unnecessary calculations, we only project the extracted keyframes, which are selected in the same way as in ORB-SLAM2.

As the label of each pixel from the semantic segmentation is known, we can also perform depth-based semantic segmentation for object-oriented point cloud generation. To guarantee the accuracy of the transformation matrix $T \in \mathbb{R}^{4\times4}$, all static pixels are used for its estimation. The GUI provides common object labels in an indoor environment for user selection. Users can not only see whether all the objects have been projected accurately (including moving humans), as shown in **Fig. 10(b)**, but also select the label of an object of interest using the left buttons so only that object is shown, as shown in the right-hand illustration of **Fig. 10(a)**.



(a) Object-oriented point cloud map.



(b) GUI.

Fig. 10. Dense point cloud map generation with the proposed GUI.

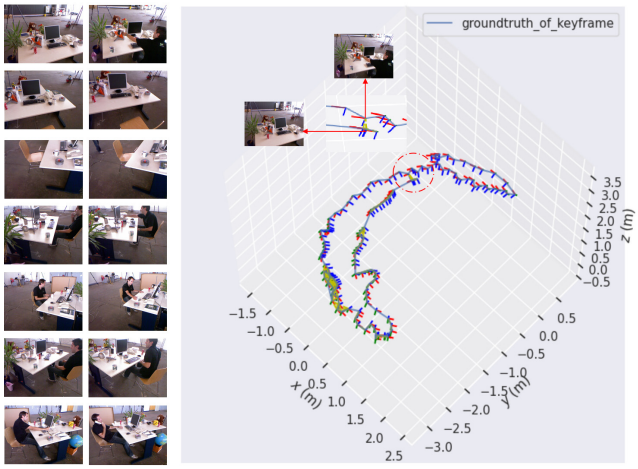


Fig. 11. Generated groundtruth loop pairs.

4. Experimental Results and Evaluation

4.1. LCD and Evaluation

We used the well-known TUM RGB-D dataset and benchmark [26] to perform the SLAM system experiments. However, this dataset does not provide the ground truth of the loop closure pairs. We followed the same way as [27] to generate them using the steps detailed below.

1. Select the keyframes computed by ORB-SLAM2 from this sequence, as the similarity score is only computed among the keyframes.

2. Calculate the relative transform matrices between every two selected keyframes. For example, $T_i \in SE(3)$ and $T_j \in SE(3)$ represent the poses of two selected keyframes, and $T_i^{-1}T_j$ is the relative transform matrix between them.

3. Compute the distance and rotation angle from the above matrix as shown in Eq. (3). We use Eq. (4) to calculate the angle, and R is the rotation matrix part in $T_i^{-1}T_j$. If the result is smaller than a threshold, these pairs are defined as the ground-truth loop pairs.

$$D_{i,j} = dis\left(T_i^{-1}T_j\right) + angle\left(T_i^{-1}T_j\right) \quad \ldots\ldots \quad (3)$$

$$\theta = \arccos\left(\frac{tr(R)-1}{2}\right), \quad R \in SO(3) \quad \ldots\ldots \quad (4)$$

The generated ground-truth loop pairs are shown in **Fig. 11**, and are denoted by the yellow dashed lines. Each row on the left is a loop pair, from which it can be seen

Fig. 12. Human existence pairs in the Garden Points Walk dataset.

that there are several types of loop pairs in this dataset:

- A human exists in one of the frames.
- A human exists in both frames, but the posture is different.
- Static environment.

In addition, we manually selected the human existence pairs from a popular public dataset in the place recognition research field called the Gardens Points Walk dataset [a]. In this dataset, there are some pedestrians walking in a garden, as shown in **Fig. 12**.

LCD is a type of binary classification problem, and the commonly used evaluation metric is the Precision-Recall (P-R) curve. For comparison using the P-R curve and Area Under Curve (AUC) in **Figs. 13** and **14**, we plotted the original result of each layer to determine the most accurate layer for LCD. From the results, it can be seen that the mid-layer pooling 4 is more accurate than the other layers in VGG-16 and the BoW used in ORB-SLAM2. This is because the initial layers only focus on the edge information, which is similar to manually hand-crafted features, meanwhile, the deeper layers retain the semantic information but lose the spatial information. Because the VGG-16 network architecture is used for object classification, it only focuses on the types of objects in the image. A similar conclusion was reached in [19] with regard to the AlexNet architecture.

In addition, because we employ the union binary mask, the keypoints extracted from humans can be filtered in the traditional BoW method. It can be observed that the performance of not only the traditional BoW method with a union mask but also that of the most accurate layer pooling 4 is improved using the proposed method. We also tested the performance of each layer used in VGG-16 with the union mask, as shown in **Table 2**. Our proposed method improved the result of each layer and the most accurate result is obtained from the mid-layer pooling 4 in VGG-16 architecture.

We also used a dataset published in [28], in which a walking human appears in the loop closure position, as shown in **Fig. 15**. In this dataset, two humans walk along a circular trajectory around a meeting table in an office. When ORB-SLAM2 is applied to this dataset, the estimated trajectory not only drifts, as shown in **Fig. 16(a)**,

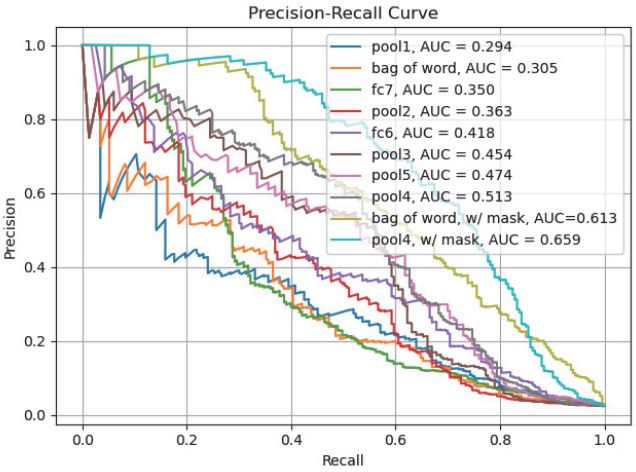


Fig. 13. Precision-recall curve for the dynamic dataset *fr2_desk_with_person*.

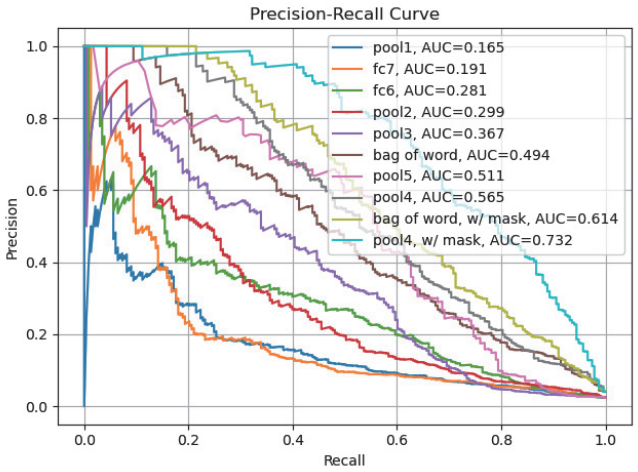


Fig. 14. Precision-recall curve for the Gardens points walk dataset.

but the loop closure in **Fig. 16(b)** is also not detected. However, using the similarity score calculated by the proposed method, our SLAM framework can detect and correct the loop, as shown in **Fig. 17**. The difference between **Figs. 16(a)** and **17(a)** lies in the improvement in the visual odometry obtained with the uniformly filtered keypoint extraction in the matching process introduced in our previous work [4].

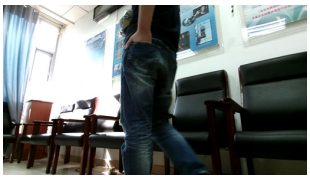
A prior study [28] reported the detection and correction of the loop in this dataset. However, their system required not only the information from the RGB-D camera but also the state estimation from the wheel encoder. In contrast, our framework requires only one camera as hardware.

4.2. Comparison of Visual Odometry and Map Reconstruction

In this subsection, we evaluate the visual odometry and map reconstruction of our proposed framework on the dynamic sequence of the TUM RGB-D dataset. The scenes in this dynamic dataset are divided into low and high dy-

Table 2. AUC comparisons between each layer in VGG-16 and BoW with and without the union mask.

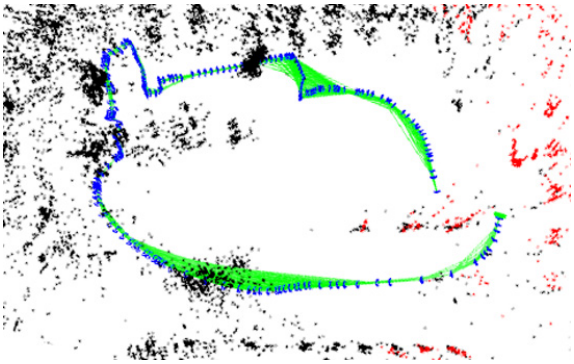
Datasets \AUC \Layer		pool 1	pool 2	pool 3	pool 4	pool 5	fc 6	fc 7	BoW
TUM	without mask	0.294	0.363	0.454	**0.513**	0.474	0.418	0.350	0.305
dataset	with mask	0.356	0.431	0.559	**0.659**	0.512	0.498	0.412	0.387
Garden points	without mask	0.165	0.299	0.367	**0.565**	0.511	0.281	0.191	0.494
walk dataset	with mask	0.211	0.373	0.434	**0.732**	0.662	0.358	0.249	0.614



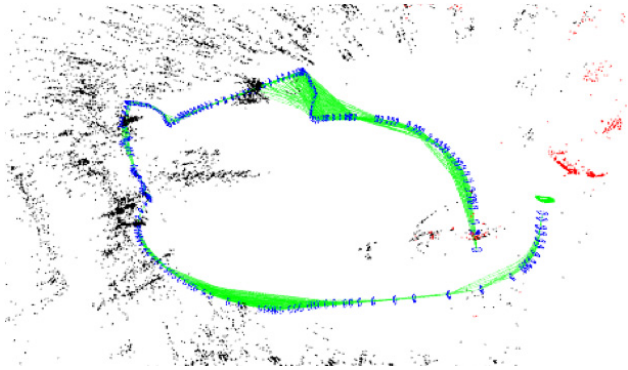
(a) The first time a human occurs at the loop closure position.

(b) The second time a human occurs at the loop closure position.

Fig. 15. Human occurrences at the loop closure position.



(a) Trajectory before the loop closure event.



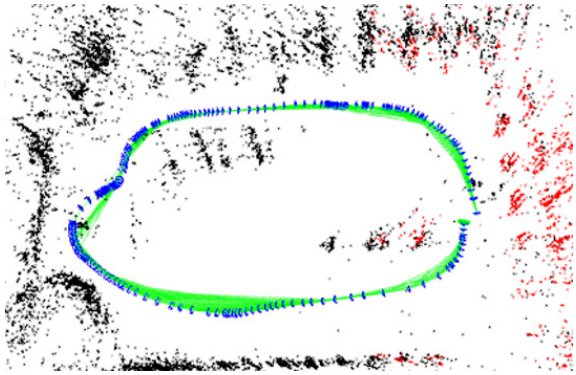
(b) Trajectory after the loop closure event.

Fig. 16. ORB-SLAM2 fails to detect and correct the loop in a dynamic environment.

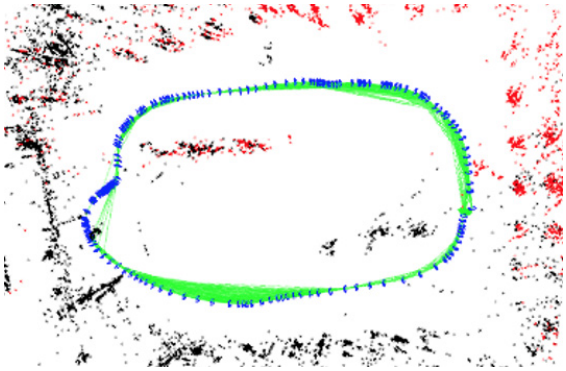
namic scenes. When humans sit at a desk, talk with each other, and gesticulate, this is defined as a low dynamic scene. When walking in an office scene, it is defined as a high dynamic scene.

Table 3 compares the results of Dynamic-SLAM [8] and ORB-SLAM2 with semantic segmentation using the ATE [29] and a common metric for the evaluation of visual odometry called the relative pose error (RPE) [30]. We compared these two methods because both use real-time and deep learning-based segmentation methods. According to **Table 3**, the accuracy of ORB-SLAM2 with semantic segmentation was higher than that of Dynamic-SLAM in most sequences. Although Dynamic-SLAM is a monocular SLAM and cannot provide a dense point cloud map at the back end of the SLAM system, the proposed framework can provide a dense object-oriented point cloud map, as shown in **Fig. 10(a)**, and a dense RGB point cloud map as shown in **Fig. 18**.

GS and the related method [10] belong to the same category, that is, they are suitable for moving object segmentation. GS cannot segment multiple dynamic objects, and in some high dynamic environments with multiple dynamic objects, [10] performs better than GS. In contrast, in some sequences, there is only one moving human



(a) Trajectory before the loop closure event.



(b) Trajectory after the loop closure event.

Fig. 17. Detection and correction of the loop using the proposed method in a dynamic environment.

Table 3. Accuracy comparison of Dynamic-SLAM [8] and the proposed framework with semantic segmentation (some data from a previous study [8] are shown).

TUM RGB-D datasets Sequence: dynamic objects		ATE [cm] RMSE			Translation RMSE [cm/frame] RMSE			Rotation RMSE [deg/frame] RMSE		
		Dynamic-SLAM	Semantic segmentation	Improvement	Dynamic-SLAM	Semantic segmentation	Improvement	Dynamic-SLAM	Semantic segmentation	Improvement
Low dynamic environment	*fr2/desk_with_person*	1.873	0.286	**84.73%**	1.958	0.502	**74.36%**	0.833	0.212	**74.55%**
	fr3/sitting_xyz	0.601	0.237	**60.57%**	0.998	0.404	**59.52%**	0.613	0.276	**54.98%**
	fr3/sitting_halfsphere	1.461	0.967	**33.81%**	1.451	0.723	**50.17%**	0.551	0.376	**31.76%**
	fr3/sitting_rpy	3.448	3.702	−7.37%	4.303	3.876	**9.92%**	0.991	1.031	−4.04%
High dynamic environment	*fr3/walking_xyz*	1.324	1.306	**1.36%**	1.796	0.703	**60.86%**	0.598	0.372	**37.79%**
	fr3/walking_halfsphere	2.139	0.435	**79.66%**	2.192	0.723	**67.02%**	0.666	0.353	**46.99%**
	fr3/walking_rpy	6.025	0.623	**89.66%**	5.605	0.423	**92.45%**	1.149	0.506	**55.96%**

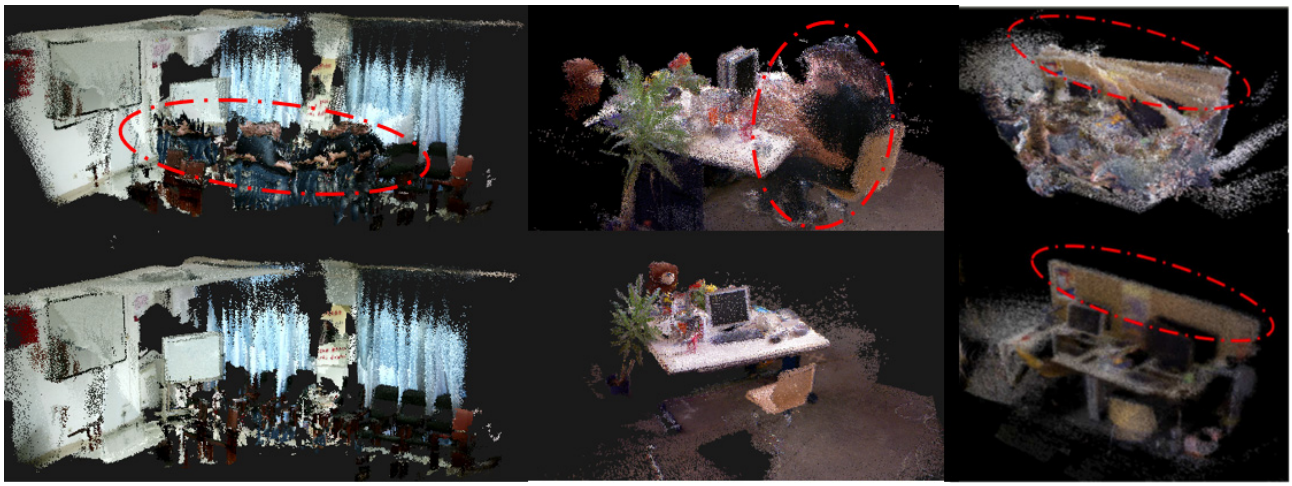


Fig. 18. Comparisons between the dense RGB point cloud maps. The top row shows the RGB-D images directly projected by the original ORB-SLAM2 estimated pose, while the bottom row shows the result of our proposed framework.

Table 4. Accuracy comparison of [10] and the proposed framework with GS.

Sequence	ATE [m] RMSE			RPE translation [m/s] / rotation [deg/s] RMSE		
	Reference [4]	GS	Improvement	Reference [4]	GS	Improvement
fr3_sitting_static	0.0066	0.0039	**40.91%**	0.0077/0.2595	0.0045/0.2238	**41.56%/13.76%**
fr3_sitting_halfsphere	0.0196	0.0165	**15.82%**	0.0245/0.5643	0.0212/0.5803	**13.47%**/−2.84%
fr3_walking_static	0.3080	0.4156	−34.94%	0.1881/3.2101	0.2423/3.8765	−28.81%/−20.76%

in most frames, such as *fr3_sitting_halfsphere*. Moreover, in the low dynamic sequences such as *fr3_sitting_static*, its absolute trajectory error (ATE) and relative pose error (RPE) are lower than those of [10], as shown in **Table 4**.

The other difference between these methods is that the back end of [10] does not contain global optimization and LCD. In contrast, our proposed framework has these features, which improve the accuracy of the entire SLAM system. Although [10] is an RGB-D SLAM method, it does not provide a dense point cloud map. In contrast, the dense point cloud map obtained with GS is shown in our previous work [4].

In **Table 5**, we compare the semantic segmentation of GS and DynaSLAM [12], as both are deep learning-based methods fused with geometry-based methods. It can be observed that the accuracies of the two methods are similar. This is because although the accuracy of semantic segmentation [7] (mIoU is approximately 56 from [b]) is lower than that of the Mask R-CNN [13] (mIoU is approximately 74.4 from [c]) used in DynaSLAM to segment the human, our proposed framework has an improved CNN-based LCD method at the back end of the system, which improves the accuracy of the entire SLAM system.

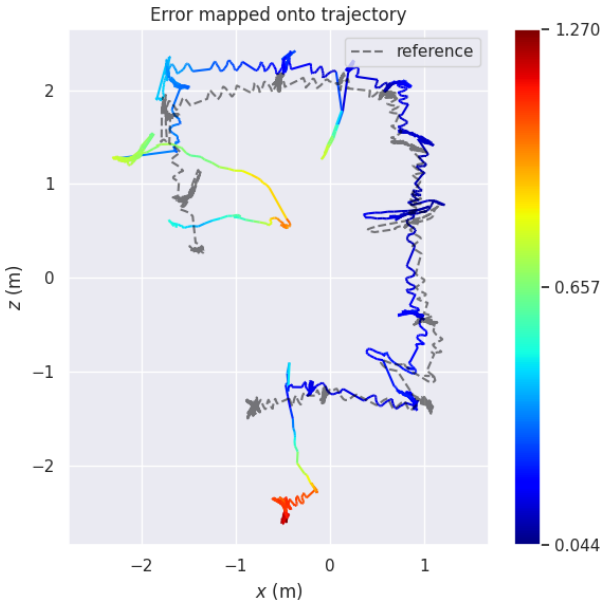
We also tested our proposed framework using a high dynamic dataset called HRPSlam [31], which was captured by a walking humanoid robot. We compared our method with PFH [32], which is structured based on ElasticFusion [33] and the original ORB-SLAM2, using the ATE and RPE metrics (data from [32]). The corresponding results are shown in **Table 6**. Because PFH is not

Table 5. Comparison between the accuracies of DynaSLAM [12] and the proposed framework with semantic segmentation and GS.

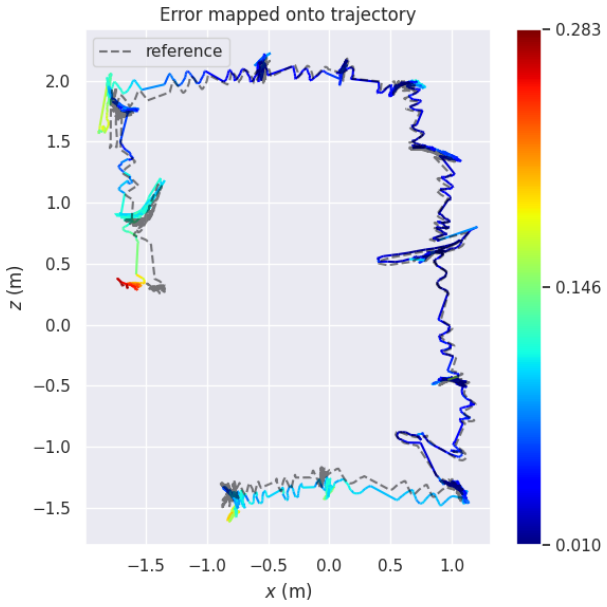
Sequence (under RGB-D)	DynaSLAM [12]	Proposed framework with semantic segmentation and GS	Improvement
	RMSE [m]	RMSE [m]	[%]
w_halfsphere	0.025	0.024	**4.00**
w_xyz	0.015	0.017	−13.33
w_rpy	0.035	0.032	**8.57**
w_static	0.006	0.006	0
s_halfsphere	0.017	0.011	**35.29**
s_xyz	0.015	0.012	**20.00**

Table 6. Comparison between the accuracies of ORB-SLAM2, PFH, and the proposed framework with semantic segmentation on the HRPSlam2 dynamic dataset.

HRPSlam2	Translation ATE RMSE [m]	Translation RPE RMSE [m]
ORB-SLAM2	0.50	0.231
PFH	0.09	0.070
Proposed framework with semantic segmentation	**0.07**	**0.015**



(a) Reference and ORB-SLAM2 results.



(b) Reference and results of the proposed framework with semantic segmentation.

Fig. 19. Trajectory comparison with the reference (ground truth).

an open-source software, we only visualized the trajectories of our framework and ORB-SLAM2 shown in **Fig. 19** using evo [d]. We found that our proposed framework achieves the highest accuracy for this dataset.

5. Conclusions and Discussion

In this study, we addressed three main problems that arise when conventional SLAM is used in a dynamic environment. The proposed framework extracts the keypoints from the static area in the image and solves the camera tracking problem. With the accurate pose and semantic label of each pixel, we can generate an object-oriented point cloud, which solves the map reconstruction problem. The LCD was improved using the proposed CNN-based feature extraction with the union mask method.

Comparisons with related methods were not only conducted using the well-known TUM RGB-D dataset and benchmark, but also on a high dynamic dataset captured by a real walking humanoid robot. In addition, we generated a human LCD dataset for comparing the results of each layer used in VGG-16. We found that the mid-layer pooling 4 was the most accurate layer, which was subsequently used to perform the element-wise multiplication with a binary union mask vector, resulting in further improvement in the performance. Finally, we used a dataset that includes loop closure in a human dynamic environment to compare the estimated trajectory of the proposed SLAM framework using the proposed CNN-based feature extraction with the union mask method and the trajectory of ORB-SLAM2 with the original BoW method. Finally, the results of the visual odometry experiment demonstrated that similar or better results could be obtained compared with the related methods owing to the uniformly distributed keypoint extractions and dilated

masks.

Recently, there have been some exciting studies on end-to-end self-supervised deep learning-based methods for camera ego-motion estimation, such as Monodepth2 [34]. However, the accumulated error in long-term ego-motion without global bundle adjustment optimization and loop closure detection makes it difficult to obtain results comparable to those of geometry-based SLAM technology such as ORB-SLAM2. CNN-SLAM [35] overcomes the limitations of traditional monocular SLAM, which involves the depth map reconstruction using CNN-predicted depth fused with ORB-SLAM. However, it does not consider the dynamic environmental conditions of the entire system and experiments.

Considering multi-view geometry, deep learning (DL), and their combination, when good quality and sufficient features are extracted, pure geometry-based SLAM is more reliable than pure DL, owing to global bundle adjustment and LCD at the back end. Because deep learning can learn features from a large training dataset, it has better robustness in feature extraction, even in some challenging environments. To leverage these factors, we proposed a framework that uses a CNN at the front end, whereas the back end is a traditional multi-view geometry-based ORB-SLAM2. The main points of our work focus on dynamic environments, including camera tracking, map reconstruction, and loop closure detection.

In addition, it might be interesting to investigate ways to deeply and tightly couple the semantic information with SLAM. It is now clear that several parts can use the semantic information provided by segmentation to solve the problems occurring in dynamic environments. Future research can also focus on methods to further fuse SLAM with semantic information. In general, there are still many directions worth exploring for the application of deep learning to SLAM.

References:

[1] H. Date and T. Takubo, "Special Issue on Real World Robot Challenge in Tsukuba and Osaka," J. Robot. Mechatron., Vol.32, No.6, p. 1103, 2020.

[2] A. Handa, A. Suzuki, H. Date et al., "Navigation Based on Metric Route Information in Places Where the Mobile Robot Visits for the First Time," J. Robot. Mechatron., Vol.31, No.2, pp. 180-193, 2019.

[3] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," IEEE Trans. Robot., Vol.33, No.5, pp. 1255-1262, 2017.

[4] L. Sun, F. Kanehiro, I. Kumagai et al., "Multi-purpose SLAM framework for dynamic environment," IEEE/SICE Int. Symposium on System Integration (SII) pp. 519-524, 2020.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Proc. of Int. Conf. Learn. Representat., pp. 1-14, 2015.

[6] Y. Yoshimoto and H. Tamukoh, "FPGA Implementation of a Binarized Dual Stream Convolutional Neural Network for Service Robots," J. Robot. Mechatron., Vol.33, No.2, pp. 386-399, 2021.

[7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 3431-3440, 2015.

[8] L. Xiao, J. Wang, X. Qiu et al., "Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment," Robot. Auton. Syst., Vol.117, pp. 1-16, 2019.

[9] W. Liu, D. Anguelov, D. Erhan et al., "SSD: Single Shot MultiBox Detector," European Conf. on Computer Vision, pp. 21-37, 2016.

[10] R. Wang, W. Wan, Y. Wang et al., "A new RGB-D SLAM method with moving object detection for dynamic indoor scenes," Remote Sens., Vol.11, No.10, 1143, 2019.

[11] J. Cheng, Y. Sun, and M. Q. Meng, "Improving monocular visual SLAM in dynamic environments: an optical-flow-based approach," Adv. Robotics, Vol.32, No.12, pp. 576-589, 2019.

[12] B. Bescos, J. M. Fácil, J. Civera et al., "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," IEEE Robot. Autom. Lett., Vol.3, No.4, pp. 4076-4083, 2018.

[13] K. He, G. Gkioxari, P. Dollár et al., "Mask R-CNN," The IEEE Int. Conf. on Computer Vision (ICCV), pp. 2961-2969, 2017.

[14] C. Yu et al., "DS-SLAM: A semantic visual SLAM towards dynamic environments," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 1168-1174, 2018.

[15] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol.39, No.12, pp. 2481-2495, 2017.

[16] M. Cummins, "Newman pp. FAB-MAP: Probabilistic localization and mapping in the space of appearance," Int. J. Robot. Res., Vol.27, No.6, pp. 647-665, 2008.

[17] D. A. Filliat, "Visual bag of words method for interactive qualitative localization and mapping," Proc. 2007 IEEE Int. Conf. on Robotics and Automation, pp. 3921-3926, 2007.

[18] Z. Chai and T. Matsumaru, "ORB-SHOT SLAM: trajectory correction by 3D loop closing based on bag-of-visual-words (BoVW) model for RGB-D visual SLAM," J. Robot. Mechatron., Vol.29, No.2, pp. 365-380, 2017.

[19] N. Sünderhauf et al., "On the performance of convnet features for place recognition," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 4297-4304, 2015.

[20] F. Wang, R. Xiaogang, and H. Jing, "Visual Loop Closure Detection Based on Stacked Convolutional and Autoencoder Neural Networks," IOP Conf. Series: Materials Science and Engineering, Vol.563, No.5, 052082, 2019.

[21] A. R. Memon, H. Wang, and A. Hussain, "Loop closure detection using supervised and unsupervised deep neural networks for monocular SLAM systems," Robotics and Autonomous Systems, Vol.126, 103470, 2020.

[22] M. Lopez-Antequera et al., "Appearance-invariant place recognition by discriminatively training a convolutional neural network," Pattern Recognition Letters, Vol.92, pp. 89-95, 2017.

[23] H. Hu, Y. Zhang, Q. Duan et al., "Loop closure detection for visual slam based on deep learning," IEEE 7th Annual Int. Conf. on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), pp. 1214-1219, 2017.

[24] Y. Xia, J. Li, L. Qi et al., "Loop closure detection for visual SLAM using PCANet features," Int. Joint Conf. on Neural Networks (IJCNN), pp. 2274-2281, 2016.

[25] B. Zhou, H. Zhao, X. Puig et al., "Scene parsing through ADE20K dataset," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 633-641, 2017.

[26] J. Sturm, N. Engelhard, F. Endres et al., "A benchmark for the evaluation of RGB-D SLAM systems," IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 573-580, 2012.

[27] X. Gao and T. Zhang, "Unsupervised learning to detect loops using deep neural networks for visual SLAM system," Auton. Robots, Vol.41, No.1, pp. 1-18, 2017.

[28] D. Yang, S. Bi, W. Wang et al., "DRE-SLAM: Dynamic RGB-D encoder SLAM for a differential-drive robot," Remote Sens., Vol.11, No.4, 380, 2019.

[29] E. Olson and M. Kaess, "Evaluating the performance of map optimization algorithms," RSS Workshop on Good Experimental Methodology in Robotics, p. 15, 2009.

[30] K. Konolige, M. Agrawal, and J. Sola, "Large-scale visual odometry for rough terrain," Robotics Research, Berin: Springer, pp. 201-212, 2010.

[31] T. Zhang and Y. Nakamura, "HRPSlam: A benchmark for RGB-D dynamic SLAM and humanoid vision," Third IEEE Int. Conf. on Robotic Computing (IRC), pp. 110-116, 2019.

[32] T. Zhang, E. Uchiyama, and Y. Nakamura, "Dense RGB-D SLAM for humanoid robots in the dynamic humans environment," IEEE-RAS 18th Int. Conf. on Humanoid Robots (Humanoids), pp. 270-276, 2018.

[33] T. Whelan, S. Leutenegger, R. Salas-Moreno et al., "ElasticFusion: Dense SLAM without a pose graph," Robotics: Science and Systems, 2015.

[34] C. Godard, O. M. Aodha, M. Firman et al., "Digging into self-supervised monocular depth estimation," Proc. of the IEEE/CVF Int. Conf. on Computer Vision, pp. 3828-3838, 2019.

[35] K. Tateno, F. Tombari, I. Laina et al., "Cnn-slam: Real-time dense monocular slam with learned depth prediction," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 6243-6252, 2017.

Supporting Online Materials:

[a] Arren Glover. Day and Night with Lateral Pose Change Datasets, 2014. https://wiki.qut.edu.au/display/cyphy/Day+and+Night+with+Lateral+Pose+Change+Datasets [Accessed September 18, 2019]

[b] https://github.com/kjchalup/coco_segmentation [Accessed December 3, 2020]

[c] A. Ayanzadeh, "A Study Review: Semantic segmentation with Deep Neural Networks," March, 2019. https://www.researchgate.net/publication/331983081_A_Study_Review_Semantic_segmentation_with_Deep_Neural_Networks [Accessed December 6, 2020]

[d] Grupp M. evo, Python package for the evaluation of odometry and SLAM, 2017. https://github.com/MichaelGrupp/evo [Accessed August 22, 2018]

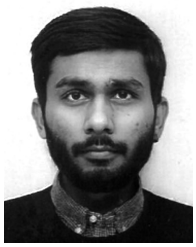
Name:
Leyuan Sun

Affiliation:
Department of Intelligent and Mechanical Interaction Systems, Graduate School of Science and Technology, University of Tsukuba
CNRS-AIST Joint Robotics Laboratory (JRL), International Research Laboratory (IRL)
National Institute of Advanced Industrial Science and Technology (AIST)

Address:
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan
AIST Tsukuba Headquarters and Information Technology Collaborative Research Center (Tsukuba Central 1), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan

Brief Biographical History:
2011-2015 Bachelor Course Student, Jiangsu University, Zhenjiang
2018-2020 Master Course Student, University of Tsukuba
2020- Ph.D. Student, University of Tsukuba

Main Works:
• "Multi-purpose SLAM framework for dynamic environment," IEEE/SICE Int. Symposium on System Integration (SII), pp. 519-524, 2020.



Name:
Rohan P. Singh

Affiliation:
Department of Intelligent and Mechanical Interaction Systems, Graduate School of Science and Technology, University of Tsukuba
CNRS-AIST Joint Robotics Laboratory (JRL), International Research Laboratory (IRL)
National Institute of Advanced Industrial Science and Technology (AIST)

Address:
1-1-1 Tennodai, Tsukuba, Ibaraki 305-8577, Japan
AIST Tsukuba Headquarters and Information Technology Collaborative Research Center (Tsukuba Central 1), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan

Brief Biographical History:
2013-2017 Bachelor Course Student, Delhi Technological University
2019-2021 Master Course Student, University of Tsukuba
2021- Ph.D. Student, University of Tsukuba

Main Works:
• "Rapid Pose Label Generation Through Sparse Representation of Unknown Objects," IEEE ICRA 2021, Xi'an, China, 2021.

Name:
Fumio Kanehiro

Affiliation:
Director, Cooperative Research Laboratory, CNRS-AIST Joint Robotics Laboratory (JRL), International Research Laboratory (IRL)
National Institute of Advanced Industrial Science and Technology (AIST)

Address:
AIST Tsukuba Headquarters and Information Technology Collaborative Research Center (Tsukuba Central 1), 1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan

Brief Biographical History:
2000- Electrotechnical Laboratory (ETL)
2001- National Institute of Advanced Industrial Science and Technology (AIST)
2007- Visiting Researcher, Laboratory for Analysis and Architecture of Systems, Centre National de la Recherche Scientifique (LAAS-CNRS)

Main Works:
• "Toward Industrialization of Humanoid Robots," IEEE Robotics & Automation Magazine, Vol.26, No.4, pp. 20-29, 2019.

Membership in Academic Societies:
• The Institute of Electrical and Electronics Engineers (IEEE)
• The Robotics Society of Japan (RSJ)