IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 10, 2020, accepted February 26, 2020, date of publication March 2, 2020, date of current version March 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2977684

Dynamic Scene Semantics SLAM Based on Semantic Segmentation

SHUANGQUAN HAN AND ZHIHONG XI
School of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China
Corresponding author: Zhihong Xi (xizhihong@hrbeu.edu.cn)

ABSTRACT Simultaneous Localization and Mapping (SLAM) have become a new research hotspot in the field of artificial intelligence applications such as unmanned driving and mobile robots. Most of the current SLAM research is based on the assumption of static scenes, and dynamic objects in the indoor environment are inevitable. The assumption based on static scenes greatly limits the development of SLAM and the application of SLAM system in real life. At the same time, the semantic segmentation is added to the SLAM system to generate a semantic map with semantic information, which can enrich the understanding of the mobile carrier to the environment and obtain high-level perception. In this paper, we combine the visual SLAM system ORB-SLAM2 and PSPNet semantic segmentation network, and propose a PSPNet-SLAM system, which uses optical flow and semantic segmentation to detect and eliminate dynamic points to achieve dynamic scenes semantic SLAM. We performed experiments on the TUM RGB-D dataset. The results show that compared with other SLAM systems, PSPNet-SLAM can reduce the camera pose estimation error in indoor dynamic scenes to different degrees and improve the camera position estimation accurately.

INDEX TERMS SLAM, semantic SLAM, indoor dynamic scene, semantic segmentation.

I. INTRODUCTION

The Simultaneous Localization and Mapping (SLAM) problem can be described as a robot moving from an unknown location in an environment without a priori knowledge. In the process of moving, it can locate itself according to the position estimation and map. At the same time, it can build an incremental map based on its own location to realize the autonomous positioning and navigation. The SLAM technology based on visual sensor is called Visual Simultaneous Localization and Mapping (VSLAM) technology. After the acquisition of RGB-D cameras with fast acquisition speed, rich acquisition information, high measurement accuracy, and relatively low price, VSLAM has been widely applied to many fields.

Over the past 30 years, many scholars have carried out research on SLAM and achieved outstanding results, making the SLAM system more mature and able to make a good performance, such as ORB-SLAM2 [1], RGBD-SLAM-V2 [2]. However, the traditional SLAM research is mostly based on the assumption of static scenes, while the existence of dynamic objects in real-life scenes is inevitable. The assumption based on static scenes greatly limits the SLAM development research and the application of the SLAM system in real life. When there are moving objects in the scene, the feature points of the dynamic objects are unstable. In the SLAM system based on feature points, when the unstable feature points are tracked, the pose estimation will be seriously affected, resulting in large trajectory errors and even system collapse. In addition, a typical SLAM system usually builds a map based on geometric information. This method only provides the structural information of the environment and its location information. It lacks an abstract understanding of the map information and cannot provide the semantic information of the surrounding environment for the perception and navigation of the mobile carrier, which limits the perception and navigation effects.

In this paper, we take advantage of the semantic segmentation network in scene understanding and propose a PSPNet-SLAM system by combining the visual SLAM system ORB-SLAM2 and PSPNet [3] semantic segmentation network. The combination of the two parts can perceive scenes from both geometric and semantic levels [4], enrich the abstract understanding of the environment, alleviate the dependence on environmental characteristics, and obtain high-level perception. At the same time, dynamic points are

The associate editor coordinating the review of this manuscript and approving it for publication was Leo Chen.

detected and eliminated using optical flow and semantic segmentation to implement a semantic SLAM system in dynamic scenes. Not only can it greatly reduces the interference of dynamic objects on pose estimation and improve the accuracy of pose estimation, but it can also generate semantic maps with semantic information, which can enrich mobile carriers' understanding of the environment and obtain high-level perception. The rest of the structure of this paper is as follows: the second part reviews the related work, the third part introduces the PSPNet-SLAM system in detail, the fourth part details the experimental results, and the fifth part introduces the conclusions and future work.

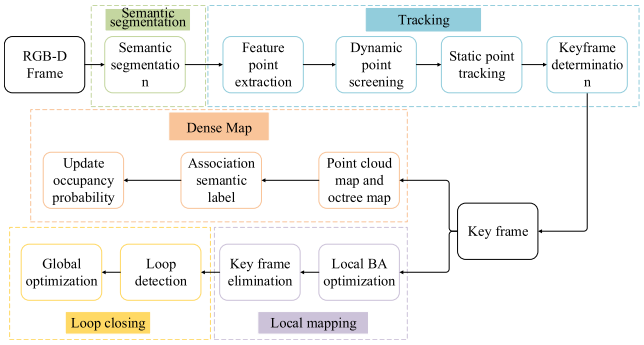


FIGURE 1. System's overall framework.

II. RELATED WORK

Semantic SLAM means that the SLAM system can obtain the geometric information of the surrounding environment and identify the independent objects in the environment during the construction process, which can get the semantic information of the position, posture, category and texture of the mobile carrier [5]. In this case, semantic SLAM can provide perception and understanding for applications in the field of artificial intelligence such as mobile robots and driverless.

For the research of semantic SLAM, the recent typical research mainly includes the following works. Li *et al.* combined SLAM with Convolutional Neural Network (CNN), selected key frames for semantic segmentation, and used 2D semantic information and adjacent keys for three-dimensional mapping. The correspondence between the frames is three-dimensionally constructed [6]. McCormac *et al.* combined ElasticFusion and CNN to calculate the pose and build a dense map using the dense SLAM system ElasticFusion. The convolutional neural network predicts the object class of each pixel, and the Bayesian update is used to generate the result of the recognition and SLAM. The associated information is integrated into a dense semantic map [7]. CubeSLAM is a cube-based 3D object detection and SLAM system that implements object-level mapping, positioning and dynamic object tracking [8]. Compared with feature-based SLAM, Yang and Scherer combine cubeSLAM and Pop-up SLAM to make the map denser, more compact and semantically more meaningful [9].

In dynamic scenes, Kim *et al.* propose obtaining static objects in the scene by calculating the difference in projections of successive depth images on the same plane [10]. Sun *et al.* distinguish dynamic static objects by calculating the intensity difference of successive RGB images, and dividing the quantized depth image to complete pixel classification [11]. Badrinarayanan *et al.* proposed a DS-SLAM scheme. DS-SLAM combines SLAM with SegNet [12] network to filter the dynamic parts using semantic information and motion feature points in dynamic scenes, thereby improving the accuracy of pose estimation [13]. Li *et al.* proposed a static weighting method for key frame edge points to indicate the probability that a point is a part of a static environment, and reduce the influence of dynamic objects on pose estimation [14]. Bescos *et al.* combine multi-view geometry and deep learning to detect and segment objects without priori dynamic mark and generate a more complete scene map by repairing the background frame occluded by the dynamic object [15]. Alcantarilla *et al.* detect moving objects through the scene stream representation of the stereo camera [16].

This paper focuses on the SLAM pose estimation and semantic map construction in indoor dynamic scenes. By combining ORB-SLAM2 with semantic segmentation network PSPNet, a semantic SLAM system in indoor dynamic scenes is proposed. Firstly, the optical flow is used to judge and cull the dynamic point, and then it is judged whether the remaining feature points fall within the priori segmented dynamic object. And the second screening is performed, which is the feature points falling within the prior dynamic object are taken as dynamic points and eliminated. Camera poses estimation is performed only by static feature points, which reduce the influence of dynamic objects on pose estimation, and generates point cloud maps and semantic octree maps with semantic information.

III. SYSTEM INTRODUCTION

A. PSPNet-SLAM SYSTEM

In this section, we will detail the PSPNet-SLAM system. Figure 1 shows the overall framework of the system. The PSPNet-SLAM system is based on the ORB-SLAM2 system. As one of the mature SLAM schemes, ORB-SLAM2 system scheme is inspired by the parallel design of tracking process and mapping process proposed by PTAM [17], which innovatively proposes three thread modes: real-time tracking feature point thread, local map optimization thread, loop detection and optimization thread. The three thread result of ORB-SLAM2 achieves a very good tracking and mapping effect, and can ensure global consistency of the trajectories and maps.

This paper adds semantic segmentation thread and dense map construction thread based on ORB-SLAM2 system. The whole system are divided into five threads: semantic segmentation thread, tracking thread, dense map construction thread, local map optimization thread and loop detection, and Optimize threads. First of all, each frame captured by

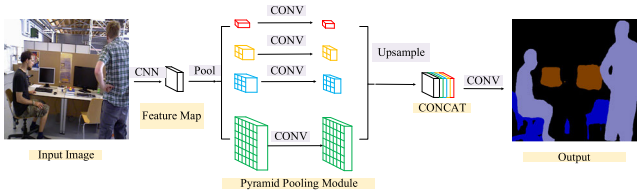


FIGURE 2. PSPNet network structure.

the camera is segmented by a semantic segmentation thread, and the categories on the image are divided pixel by pixel. Then the tracking thread extracts the ORB feature points, and determines whether the ORB feature points are dynamic points, eliminate s the selected dynamic points, and uses static points to track. Then use keyframes to build local maps and update global maps, and loop detection is carried out.

B. SEMANTIC SEGMENTATION NETWORK

Compared with DS-SLAM using SegNet semantic segmentation network, the semantic segmentation network in this SLAM system adopts PSPNet based on caffe [18].

The SegNet network is based on the Fully Convolutional Network (FCN) [19], which is modified by the VGG-16 network, but the FCN has several problems: first of all, the FNC lacks the ability to infer from context; secondly, it cannot make up the association between labels through the relationship between categories; thirdly: the model may ignore small objects, while large objects may exceed the FCN acceptance range, resulting in discontinuous predictions. In summary, FCN does not handle the relationships and global information between scenes very well. The PSPNet [8] network proposes a pyramid scene analysis network, which can embed the difficult to analyze scene information features into the FCN prediction framework, integrate the local information with the global features, and proposed an optimization strategy for moderately monitoring losses, which can obtain global scene information and effectively handle relationships between scenes.

The PSPNet network structure is shown in Figure 2, which training 20 categories on the PASCAL VOC dataset [20]. The input image is extracted by the convolutional neural network, and the extracted feature map is passed through the pyramid pooling module to obtain features with overall information at different scales. After upsampling, the feature maps of different levels generated by the pyramid are connected. Finally, the classification of each pixel is obtained through the convolutional layer.

C. DYNAMIC POINT SCREENING

The entire process of ORB-SLAM2 is based on feature points. The feature point method extracts feature points from each frame of the picture, and performs matching of adjacent frames through the invariant descriptors of the feature points. Then, the camera poses and map point are more robustly recovered through the polar geometry, and finally the camera pose and map structure optimized by

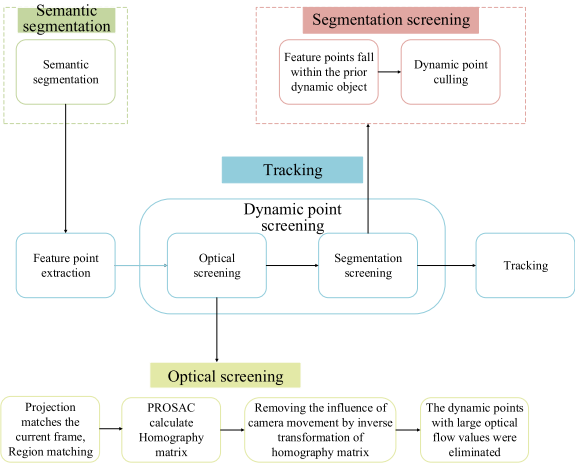


FIGURE 3. Dynamic point screening process.

minimizing projection error. The feature points extracted from each frame are detected by clustering and other operations to loop detect or relocate. Feature points run through the entire process of SLAM and are the cornerstone of the SLAM system based on the feature point method. The selection of feature points determines the quality of the later process of construction and optimization.

The screening process of dynamic points is shown in Figure 3.

Compared with ORB-SLAM2, this system has a more strict feature point selection strategy to ensure more correct matching points to meet the requirements of reducing the trajectory error and achieve stable operation of the system under dynamic scenes. We combine two methods to achieve dynamic point screening. One method is to use optical flow to judge dynamic points for screening, and the other is to use feature points falling within the priori segmented dynamic objects as dynamic points. Our dynamic point screening strategy is to first use the optical flow to judge the dynamic points for filtering and culling. The feature points with large optical flow values are removed as dynamic point, and the feature points with small optical flow values are used as static background object feature points. Then, it is judged whether the remaining feature points fall within the segmented a priori dynamic object. For the feature points falling within the prior dynamic object, they are taken as dynamic points and eliminated as a secondary screening to ensure more correct matches. The transformation matrix T is calculated by matching the remaining correct stable feature points. On the one hand, the method of secondary screening can ensure more correct static matching feature points, and the second is to avoid the occurrence of false positives caused by the priori dynamic objects remaining stationary. All of these improvements make the system more robust and work well even in poor scenes.

D. PROSAC SOLVES THE HOMOGRAPHY MATRIX

The solution of the homography matrix H is solved by the Progressive Sample Consensus (PROSAC) [21], which

eliminates the problem of unstable iterations. The process of the PROSAC algorithm is to pre-order the sample points, select the matching sample points, and then estimate the matching model. It uses the result of the initial set matching of the points as the basis for sorting, so that the sampling results are sorted according to the high-to-low score of the matching result. In this way, the sampling that is most likely to get the best parameters will appear earlier, which reduces the number of iterations of the model, improves the speed, and the model correct rate is high.

The PROSAC algorithm can effectively eliminate the mismatched points in the image matching. In the image matching process, the ratio of the Euclidean distance β is established for each pair of feature points. The calculation formula is as follows:

$$\beta = \frac{d_{\min}}{d_{\min 2}} \tag{1}$$

where d_{min} is the minimum Euclidean distance and d_{min2} is the second Euclidean distance. The smaller the ratio, the smaller the distance and the better the quality of the feature point matching. A quality factor can be introduced to measure the quality of the match, ie:

$$\gamma = \frac{1}{\beta d_{\min}} \tag{2}$$

The quality factor is introduced into the feature point grading, so that the feature point matching quality is improved, the number of iterations is reduced, and the time complexity of the algorithm is reduced. The algorithm steps are as follows.

(1) Calculate the minimum Euclidean distance d_{min} of the feature point, and solve the Euclidean distance ratio β;

(2) Calculate the quality factor and measure the quality of the matching points;

(3) Select the hypothesis generation set and semi-random sampling: according to the descending order of the matching quality, select m points, and calculate the quality sum by each of the four groups and sort them;

(4) Computational model: select the four highest-quality matching points in the sorted matching pairs, and calculate the homography matrix H;

(5) Model verification

1) Calculate the set of support points: After removing the above four sets of points, calculate the corresponding projection points based on the homography matrix H, and calculate the error e between the remaining points and the projection points, and compare with the error threshold δ, $e < \delta$ is an inner point, and vice versa is an outer point;

2) Update the model and iteration parameters: if the number of inner points $t > T$ which is seted the threshold number of inner points, the number of inner points is updated to t, calculate the homography matrix H again, and calculate the new inner point; otherwise, the number of iterations is increased by 1, repeating the above steps ;

3) If the number of iterations < maximum iteration number Im, return the homography matrix H, otherwise return no result.

E. SEMANTIC MAP

The system simultaneously establishes a semantic point cloud map and a semantic octree map, which can better present the indoor scene. The objects identified by the semantic segmentation are labeled with different color information, and the dynamic objects (in the data set is the person) in the scene got a good rejection. In an octree, a node stores information about whether it is occupied. When all child nodes are occupied or not occupied, it is not necessary to expand this node. When adding information to a map, since the actual objects are often connected together, the blanks are often connected together, so most octree nodes do not need to be expanded to the leaf level. Therefore, compared with the semantic point cloud map, the semantic octree map takes up about one percent of the semantic point cloud map, saves a lot of hard disk space, provides navigation maps for robots, and provides high levels of perceived information.

In an octree, it is described by a probability log (Log-odds). Let $y \in R$ be the probability logarithm and x be the probability between 0 and 1, then the transformation between them is described by the logarithmic transformation:

$$y = \log it(x) = \log(\frac{x}{1-x}) \tag{3}$$

The inverse transformation is:

$$x = \log it^{-1}(y) = \log(\frac{\exp(y)}{\exp(y)+1}) \tag{4}$$

In mathematical terms, let a node be n and the observed data be z. Then the probability logarithm of a node from the beginning to the time t is L(n|z1:t), then the time t + 1 is:

$$L(n|z_{1:t+1}) = L(n|z_{1:t-1}) + L(n|z_t) \tag{5}$$

Use y to express whether the node is occupied. When it is continuously observed that the node is ''occupied'', y increases, otherwise y decreases. When querying the probability of a node, the logit is inversed and y is transformed into probability.

IV. EXPERIMENT AND ANALYSIS

A. DATA SET

The TUM RGB-D data set required for the experiment is a large open-source dataset from TUM (Technische Universität München) containing RGB-D data and ground truth data to establish a new benchmark for visual ranging and visual SLAM system evaluation.

The test data set of this paper mainly uses five sequences in the data set, namely freiburg3_sitting_static (f_s_static), freiburg3_walking_static(f_w_static), freiburg3_walking_xyz(f_w_xyz), reiburg3_walking_halfsphere(f_w_halfhere) and freiburg3_walking_rpy(f_w_rpy). In the freiburg3_sitting_static sequence, two people sit at the desk, the camera is kept in place, which is regarded as a low dynamic

TABLE 1. Absolute trajectory error results.

Sequence		f_w_xyz	f_w_static	f_w_rpy	f_w_halfhere	f_s_static
ORB-SLAM2 meter(m)	RMSE	0.801255	0.341436	0.830807	0.4512938	0.008661
	Mean	0.670056	0.313265	0.741004	0.3756166	0.007618
	Median	0.569857	0.268446	0.711878	0.2997844	0.006817
	S.D.	0.437586	0.1354	0.356189	0.2496192	0.004117
DS-SLAM meter(m)	RMSE	0.026494	0.008074	0.597722	0.0313056	0.00681
	Mean	0.021179	0.007201	0.453992	0.0226248	0.005888
	Median	0.017452	0.006629	0.3324	0.0226248	0.005132
	S.D.	0.015863	0.003649	0.288128	0.0163846	0.003422
DynaSLAM meter(m)	RMSE	0.015672	**0.006899**	0.041785	0.0301564	0.006397
	Mean	**0.013486**	**0.005959**	0.031252	0.0258218	0.00555
	Median	0.011842	**0.005264**	0.024051	0.02185842	0.004999
	S.D.	0.007982	0.003474	0.027564	0.01557	0.003179
PSPNet-SLAM meter(m)	RMSE	**0.015622**	0.007283	**0.033358**	**0.0255974**	**0.005839**
	Mean	0.013531	0.006456	**0.026175**	**0.0222732**	**0.005025**
	Median	**0.011749**	0.005874	**0.020435**	**0.0196024**	**0.004408**
	S.D.	**0.007805**	**0.003366**	**0.020613**	**0.0126034**	**0.002973**
Improvement (Compare To ORB-SLAM2) percentage(%)	RMSE	98.05%	97.87%	95.98%	94.33%	32.58%
	Mean	97.98%	97.94%	96.47%	94.07%	34.03%
	Median	97.94%	97.81%	97.13%	93.46%	35.34%
	S.D.	98.22%	97.51%	94.21%	94.95%	27.80%
Improvement (Compare To DS-SLAM) percentage(%)	RMSE	41.03%	9.79%	94.42%	18.23%	14.26%
	Mean	36.11%	10.35%	94.23%	1.55%	14.64%
	Median	32.67%	11.39%	93.85%	13.36%	14.10%
	S.D.	50.80%	7.76%	92.85%	23.08%	13.14%
Improvement (Compare To DynaSLAM) percentage(%)	RMSE	0.32%	-5.57%	20.17%	15.12%	8.73%
	Mean	-0.34%	-8.33%	16.24%	13.74%	9.45%
	Median	0.78%	-11.58%	15.03%	10.32%	11.83%
	S.D.	2.22%	3.11%	25.22%	19.05%	6.49%

TABLE 2. Translation drift results.

Sequence		f_w_xyz	f_w_static	f_w_rpy	f_w_halfhere	f_s_static
ORB-SLAM2 meter(m)	RMSE	0.411713	0.186739	0.424113	0.3234162	0.009353
	Mean	0.307457	0.079623	0.28631	0.9217376	0.006214
	Median	0.212753	0.014046	0.143121	0.3532302	0.007413
	S.D.	0.273214	0.168892	0.312542	1.2100034	0.004469
DS-SLAM meter(m)	RMSE	0.03392	0.01057	0.162316	0.0302166	0.007532
	Mean	0.025801	0.009223	0.101548	0.0231482	0.006507
	Median	0.019973	0.008139	0.046238	0.0231482	0.005626
	S.D.	0.021981	0.005146	0.126579	0.0152312	0.003792
DynaSLAM meter(m)	RMSE	0.020635	**0.00932**	0.059172	0.0279774	0.006198
	Mean	0.017051	**0.007972**	0.042291	0.0240924	0.007052
	Median	**0.015454**	**0.006952**	0.030183	0.0225154	0.00614
	S.D.	0.01069	0.004829	0.042288	0.0131718	0.004135
PSPNet-SLAM meter(m)	RMSE	**0.019565**	0.009529	**0.043692**	**0.0262494**	**0.007025**
	Mean	**0.017203**	0.008371	**0.035528**	**0.023058**	**0.006094**
	Median	0.015635	0.007512	**0.030127**	**0.0210834**	**0.005332**
	S.D.	**0.009524**	**0.004548**	**0.025379**	**0.0125358**	**0.003493**
Improvement (Compare To ORB-SLAM2) percentage(%)	RMSE	95.22%	94.90%	89.70%	91.88%	24.89%
	Mean	94.40%	89.49%	87.59%	97.50%	25.61%
	Median	92.63%	46.52%	78.95%	94.03%	28.07%
	S.D.	96.51%	97.31%	91.88%	98.96%	21.63%
Improvement (Compare To DS-SLAM) percentage(%)	RMSE	42.02%	9.85%	73.08%	13.13%	6.73%
	Mean	33.33%	9.24%	65.01%	0.39%	6.35%
	Median	21.72%	8.05%	34.84%	8.92%	5.23%
	S.D.	56.67%	12.63%	79.95%	17.70%	7.87%
Improvement (Compare To DynaSLAM) percentage(%)	RMSE	4.70%	-2.24%	25.16%	6.13%	14.31%
	Mean	2.54%	-5.01%	15.99%	6.62%	13.59%
	Median	-1.17%	-8.06%	0.19%	6.35%	13.16%
	S.D.	10.90%	5.82%	39.98%	4.83%	15.51%

sequence, and the other four sequences are two people walking through the office, which is regarded as a high dynamic sequence. In freiburg3_walking_static sequence, camera is held in place. And in the freiburg3_walking_xyz sequence, camera moves in three directions (x, y, z). While in the freiburg3_walking_halfspher sequence, camera moves on a small hemisphere of approximately one meter diameter, and in the freiburg3_walking_rpy sequence in the camera along the main axis (roll-pitch-yaw) rotate at the same position.

In addition, the data set provides methods for system evaluation - Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) [22]. ATE represents the global consistency of the trajectory, while RPE measures the translation and rotational drift.

B. EXPERIMENTAL RESULTS

In this section, the experimental results of the SLAM system in this paper will be presented to illustrate the performance of the system in the TUM RGB-D dynamic scene dataset. All experiments were performed on a computer equipped with an Intel i7 CPU, GTX1070 GPU and 16 GB of memory.

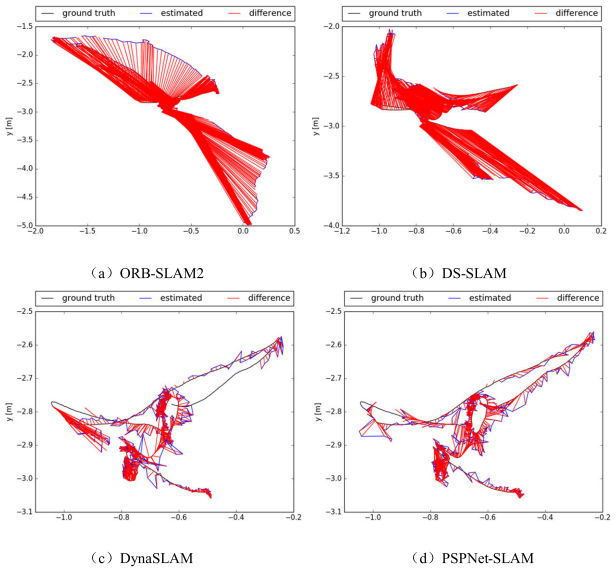
The quantitative comparison results are shown in Tables 1-3. The experimental results of the system and ORB-SLAM2, DS-SLAM and DynaSLAM in the five sequences of the data set were compared. The evaluation indexes are Root Mean Squared Error (RMSE), Mean Error (Mean), Median error and Standard Deviation error (SD). Root mean square error (RMSE) describes the deviation between the estimated value and the true value, so the smaller the value, the closer the representative estimated trajectory is to the true value; The average error reflects the average level of all estimated errors; the median error represents a medium level of all errors; Standard deviation (SD) reflects the degree of dispersion of system trajectory estimates. These kinds of objective evaluation algorithms show the difference between the estimated trajectory and the true value of the system, reflecting the stability and reliability of the system.

1) QUANTITATIVE RESULTS

The experimental results of the PSPNet-SLAM system and ORB-SLAM2, DS-SLAM and DynaSLAM in the five sequences of the data set are shown in Tables 1-3.

TABLE 3. Rotational drift results.

Sequence		f_w_xyz	f_w_static	f_w_rpy	f_w_halfhere	f_s_static
ORB-SLAM2 degree/100meters (°/100m)	RMSE	7.954316	3.259705	8.322576	6.483836	0.28514
	Mean	5.960232	1.430645	5.603499	3.831081	0.257029
	Median	4.139905	0.333768	2.755375	1.410445	0.244766
	S.D.	5.250082	2.873435	6.145101	5.229999	0.123434
DS-SLAM degree/100meters (°/100m)	RMSE	0.848542	0.275236	3.245959	**0.662985**	0.265777
	Mean	0.623587	0.246111	2.06628	**0.622621**	0.239274
	Median	0.460808	0.229568	0.997454	0.622621	0.229004
	S.D.	0.57523	0.123089	2.502902	0.408713	0.115683
DynaSLAM degree/100meters (°/100m)	RMSE	0.622853	**0.252227**	1.32067	0.7933	0.271277
	Mean	0.491677	0.224555	0.957276	0.692715	0.244294
	Median	0.401068	0.310065	0.684801	0.620858	0.232113
	S.D.	0.382359	**0.114855**	0.905488	0.386466	0.117932
PSPNet-SLAM degree/100meters (°/100m)	RMSE	**0.605015**	0.258299	**0.987776**	0.75518	**0.258071**
	Mean	**0.477198**	0.231103	**0.798435**	0.661354	**0.231536**
	Median	**0.396679**	**0.214762**	**0.672544**	**0.59199**	**0.220519**
	S.D.	**0.371917**	0.115336	**0.579908**	**0.364345**	**0.113966**
Improvement (Compare To ORB-SLAM2) percentage(%)	RMSE	92.39%	92.08%	88.13%	88.35%	9.49%
	Mean	91.99%	83.85%	85.75%	82.74%	9.92%
	Median	90.42%	35.66%	75.59%	58.03%	9.91%
	S.D.	92.92%	95.99%	90.56%	93.03%	7.67%
Improvement (Compare To DS-SLAM) percentage(%)	RMSE	28.70%	6.15%	69.57%	-13.91%	2.90%
	Mean	23.48%	6.10%	61.36%	-6.22%	3.23%
	Median	13.92%	6.45%	32.57%	4.92%	3.71%
	S.D.	35.34%	6.30%	76.83%	10.86%	1.18%
Improvement (Compare To DynaSLAM) percentage(%)	RMSE	2.86%	-2.41%	25.21%	4.81%	4.87%
	Mean	2.94%	-2.92%	16.59%	4.53%	5.22%
	Median	1.09%	30.74%	1.79%	4.65%	4.99%
	S.D.	2.73%	-0.42%	35.96%	5.72%	3.36%



(a) ORB-SLAM2 (b) DS-SLAM

(c) DynaSLAM (d) PSPNet-SLAM

FIGURE 4. Absolute trajectory error of ORB-SLAM2, DS-SLAM, DynaSLAM, and PSPNet-SLAM in fr3_w_rpy sequence.

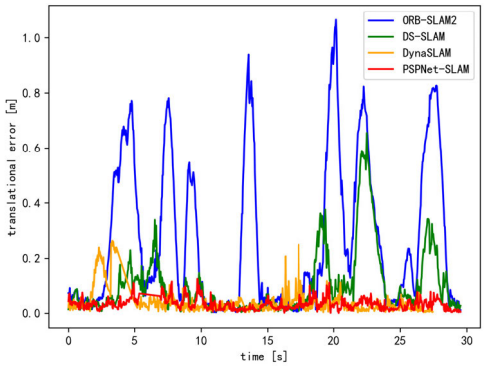
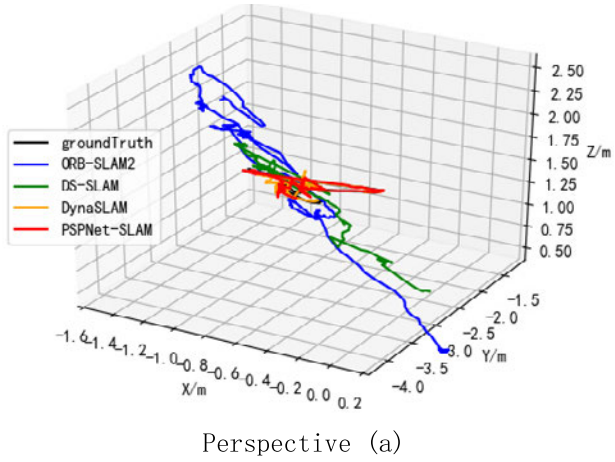


FIGURE 5. The relative pose error of ORB-SLAM2, DS-SLAM, DynaSLAM and PSPNet-SLAM in the fr3_w_rpy sequence.



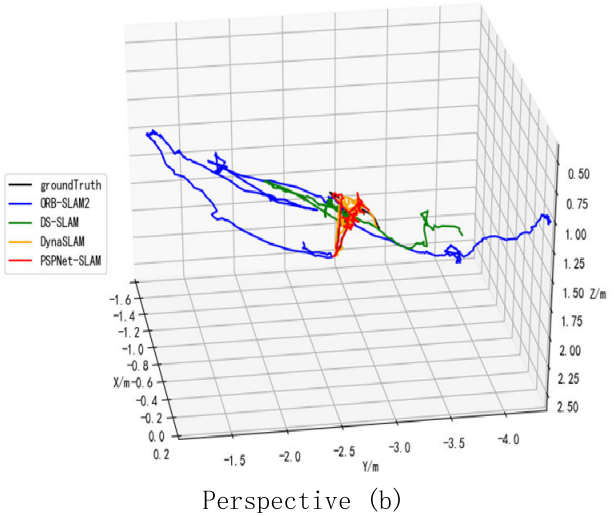
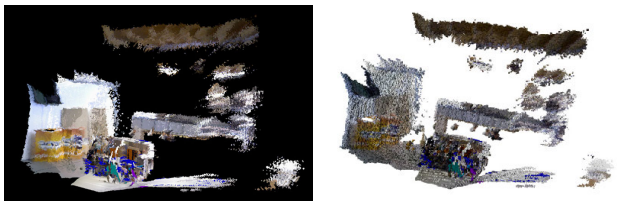


FIGURE 6. The camera trajectories of the ORB-SLAM2, DS-SLAM, DynaSLAM and PSPNet-SLAM in fr3_w_rpy.

It can be seen from Tables 1-3 that compared with the ORB-SLAM2 system, the SLAM system can greatly reduce the absolute trajectory error and the relative pose error in all sequences, improving performance and stability; Compared with DS-SLAM, the SLAM system in this paper has different degrees of decline in absolute trajectory error and translation drift error in five sequences. In the highly dynamic sequence freiburg3_walking_xyz(f_w_xyz) and freiburg3_walking_rpy(f_w_rpy), the improvement



(a)Point cloud map (b)Octree map

FIGURE 7. Semantic point cloud map and semantic octree map.

TABLE 4. Time evaluation.

Sequence	TIMES(m s)	f_w_xyz	f_w_static	f_w_rpy	f_w_halfhere	f_s_static
ORB-SLAM 2	mean tracking time	0.06798468	0.06163752	0.05969586	0.06625328	0.04936602
DS-SLAM	mean tracking time	0.121047	0.1092362	0.115098	0.119646	0.1027732
DynaSLAM	mean tracking time	0.4767328	0.5134688	0.03888276	0.477465	0.1662794
PSPNet-SLAM	mean tracking time	0.6584909	0.6125528	0.6375569	0.6599212	0.5943463

is obvious. Only in reiburg3_walking_halfsphere(f_w_halfhere), there is a small gap compared with the DS-SLAM results. Compared with the DynaSLAM system, the SLAM system can improve the performance and stability of most high dynamic sequences. The results show that the SLAM system can improve the robustness and stability of the SLAM system in highly dynamic scenes. However, in low dynamic sequences, such as the fr3_sitting_static sequence, the error reduction is small because ORB-SLAM2, DS-SLAM, and DynaSLAM can already handle low dynamic scenes well and achieve good performance, so the space that can be improved is limited.

2) QUALITATIVE RESULTS

Figure 4 shows the absolute trajectory error ATE of ORB-SLAM2, DS-SLAM, DynaSLAM and this SLAM system in the highly dynamic freiburg3_walking_rpy sequence. Figure 5 shows the RPE results of the ORB-SLAM2, DS-SLAM, DynaSLAM and this SLAM system in the highly dynamic freiburg3_walking_rpy sequence. It can be seen that the absolute trajectory error and relative pose error of the SLAM system in this paper have different degrees of reduction. Figure 6 shows the camera trajectories of the ORB-SLAM2, DS-SLAM, DynaSLAM and PSPNet-SLAM systems in the highly dynamic freiburg3_walking_rpy sequence. It can be seen that the camera trajectory of the system trajectory in this paper can well coincide with the ground truth trajectory.

3) SEMANTIC POINT CLOUD MAP AND SEMANTIC OCTREE MAP

The experimentally generated semantic point cloud map and semantic octree map are shown in Figure 7. The display and chair on the point cloud map and the octree map are colored.

4) TIME ANALYSIS

This part of the experiment is used to evaluate the tracking time of each system. The average tracking time is shown in Table 4. The PSPNet-SLAM proposed in this paper focuses on trajectory error and semantic map construction, so it does not optimize real-time performance of the PSPNet-SLAM. In PSPNet-SLAM, the process of solving homography matrix by PROSAC and removing camera motion by inverse transformation of homography matrix takes a part of time, so the real-time performance is not good enough.

V. CONCLUSION

The existence of dynamic objects has great influence on the estimation of trajectory and pose, and it is very practical to eliminate dynamic objects to reduce trajectory and pose error. At the same time, semantic segmentation network can detect a priori dynamic points. The ORB-SLAM2 system is combined with the PSPNet semantic segmentation network to filter the feature points by optical flow and semantic segmentation, detect and eliminate dynamic points, and use stable static feature points to perform motion estimation under dynamic scenes to complete the construction of semantic maps. The advantages of the system in reducing the trajectory and pose error are verified by experimental comparison. However, the effect of segmentation is still not ideal. On the point cloud map, the segmentation of the object point cloud is still not perfect. In the next work, the segmentation of the point cloud will be studied and the rendering of the map will be optimized.

REFERENCES

[1] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[2] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, Feb. 2014.

[3] H. Zhao, J. Shi, X. Qi, X. Wangh, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2881–2890.

[4] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[5] J. S. Yu, H. Wu, and G. H. Tian, "Semantic database design and semantic map construction of robots based on the cloud," *Robot*, vol. 38, no. 4, pp. 410–419, 2016.

[6] X. Li and R. Belaroussi, "Semi-dense 3D semantic mapping from monocular SLAM," 2016, *arXiv:1611.04144*. [Online]. Available: http://arxiv.org/abs/1611.04144

[7] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4628–4635.

[8] S. Yang and S. Scherer, "CubeSLAM: Monocular 3-D Object SLAM," *IEEE Trans. Robot.*, vol. 35, no. 4, pp. 925–938, Aug. 2019.

[9] S. Yang and S. Scherer, "Monocular object and plane SLAM in structured environments," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3145–3152, Oct. 2019.

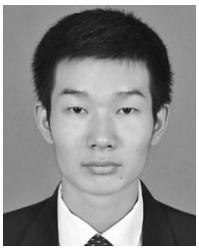
[10] D.-H. Kim and J.-H. Kim, "Effective background model-based RGB-D dense visual odometry in a dynamic environment," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1565–1573, Dec. 2016.

[11] Y. Sun, M. Liu, and M. Q.-H. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Robot. Auto. Syst.*, vol. 89, pp. 110–122, Mar. 2017.

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[13] C. Yu, Z. Liu, X.-J. Liu, F. Xie, Y. Yang, Q. Wei, and Q. Fei, "DS-SLAM: A semantic visual SLAM towards dynamic environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 1168–1174.

[14] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2263–2270, Oct. 2017.
[15] B. Bescos, J. M. Facil, J. Civera, and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 4076–4083, Oct. 2018.
[16] P. F. Alcantarilla, J. J. Yebes, J. Almazan, and L. M. Bergasa, "On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1290–1297.
[17] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.
[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2014, pp. 675–678.
[19] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
[20] M. Everingham and J. Winn, "The PASCAL visual object classes challenge 2007 (VOC2007) development kit," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2006.
[21] O. Chum and J. Matas, "Matching with PROSAC—Progressive sample consensus," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2005, pp. 220–226.
[22] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.



SHUANGQUAN HAN is currently pursuing the M.S. degree with the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China. His current research interests include visual SLAM, computer vision, image understanding, and deep learning.

ZHIHONG XI was born in 1965. She received the Ph.D. degree. She is a Professor. She is currently with the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China. Her research interests include image processing and indoor positioning.