IJRCS

International Journal of Robotics and Control Systems
Vol. 4, No. 1, 2024, pp. 88-104
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

ASCEE

Research on Indoor 3D Reconstruction Technology Based on Semantic Visual Simultaneous Localization and Mapping

Yu Liang [a, 1], Cao Lijia [a,b,c,2,*], Fu Changyou [a, 3]

[a] School of Computer Science & Engineering, Sichuan University of Science & Engineering, Yibin 644000, China
[b] Artificial Intelligence Key Laboratory of Sichuan Province, Yibin 644000, China
[c] Key Laboratory of Enterprise Informatization and IoT Measurement and Control Technology of Sichuan Province University, Zigong 643000, China
[1] 861221301@qq.com; [2] caolj@suse.edu.cn; [3] fcybill@163.com
* Corresponding Author

ARTICLE INFO

Article history
Received December 26, 2023
Revised January 30, 2024
Accepted February 12, 2024

Keywords
Visual SLAM;
Semantic SLAM;
3D Reconstruction;
Object Detection;
Dynamic Features

ABSTRACT

In response to the challenge that traditional visual simultaneous localization and mapping (SLAM) systems, based on the assumption of a static environment, struggle to achieve real-time indoor 3D reconstruction in complex dynamic scenes, this paper proposes a real-time indoor 3D reconstruction algorithm based on semantic visual SLAM. By leveraging object detection to obtain 2D semantic information and providing prior information for geometric methods, the fusion of the two effectively suppresses dynamic features, reduces reliance on deep learning methods, and ensures the algorithm's real-time performance. Experimental results on dynamic scenes in the TUM RGB-D dataset show that our algorithm maintains nearly unchanged real-time performance while achieving an average performance improvement of approximately 97.56% and 97.31% on the TUM dataset and Bonn dataset, respectively, compared to the ORB-SLAM2 system. Moreover, our algorithm can reconstruct more intuitive indoor global Octo-map and semantic metric maps compared to sparse point cloud maps, effectively enhancing the scene perception capability of mobile robots and laying the foundation for performing advanced tasks. Furthermore, our algorithm demonstrates a 3.5-10.5 times improvement in real-time performance compared to other mainstream semantic SLAM systems. Experimental results on the NVIDIA Jetson AGX Xavier confirm that our algorithm can run in real time on low-power platforms such as mobile robots or drones. However, the drawbacks of our algorithm include lower reconstruction accuracy in low-texture and large-scale scenes and ineffective suppression of dynamic features in low-dynamic scenes. Future work will consider replacing and improving deep learning methods and integrating IMU and other sensors to enhance system usability.

1. Introduction

In the modern field of computer vision, indoor 3D reconstruction technology has always been a research direction of great concern [1], [2]. With the rapid development of applications such as mobile robots, autonomous driving [3], VR/AR [4]-[6], the demand for accurate and efficient indoor 3D reconstruction is also growing day by day [7].

https://doi.org/10.31763/ijrcs.v4i1.1266

In indoor real-time 3D reconstruction, visual SLAM techniques are commonly used [8]. A complete visual SLAM system typically consists of four modules: visual odometry [9], backend optimization, loop closing [10], and mapping [11]. As a technology that integrates structured environmental information and camera localization [12], visual SLAM has attracted widespread attention in recent years. Traditional visual SLAM methods, represented by the ORB-SLAM series [13], [14], primarily focus on the geometric structure of the scene and camera pose estimation. These techniques have matured and perform well in certain scenarios. However, with the advancement of SLAM research, SLAM has entered the era of robust perception [15], posing higher requirements for the robustness of the system and the high-level nature of map reconstruction to address more complex application scenarios [16]. Traditional visual SLAM systems are mostly based on the assumption of a static scene [17], and they often fail to handle dynamic objects or handle them crudely using geometric methods, leading to erroneous data associations [18]. In real-world scenarios, which often include dynamic objects such as vehicles and pedestrians, the accuracy and robustness of these systems are generally poor [19]. Additionally, these systems typically only construct globally consistent metric maps of the scene, lacking semantic information about objects. Relying solely on geometric information often fails to provide sufficient semantic understanding [20]. Ideally, we want mobile robots to understand scene environments like humans do, enabling them to perform high-level tasks. However, the scene geometric metric maps reconstructed by traditional visual SLAM systems typically contain only simple geometric information such as points, lines, and planes, lacking semantic information about the scene [21]. With the development of deep learning methods, researchers have begun integrating semantic segmentation or object detection methods into visual SLAM systems [22]. This has led to the emergence of semantic visual SLAM systems, which aim to integrate semantic information to achieve more accurate and semantically meaningful reconstruction results.

Semantic visual SLAM integrates sensor data and visual information to model indoor scenes and understand environments. It identifies different categories of objects, scene semantics, and semantic relationships, and matches and integrates them with the map's topological structure [23]. This semantic understanding capability enhances indoor 3D reconstruction, making it more intelligent and practical. It has wide-ranging applications in various fields such as indoor navigation, smart homes, and VR/AR experiences [24].

However, semantic visual SLAM technology still faces many challenges in indoor 3D reconstruction [25]. One is the problem of semantic map reconstruction. How to fuse semantic information into 3D reconstruction and improve scene understanding and application effects is an important issue [26]. Second, the processing of dynamic objects. The existence of dynamic objects will lead to instability and uncertainty of the reconstruction results [27], [28]. Therefore, how to effectively deal with dynamic objects is an urgent problem to be solved, and there are some works on the solution of this problem [29], [30]. Finally, for real-time application scenarios, semantic visual SLAM systems are generally required to run in real-time in the scene [31], which puts high demands on their real-time performance [32]. Some current semantic visual SLAM systems themselves have higher accuracy and robustness, for example, systems using semantic segmentation methods like Mask R-CNN [33] or SegNet [34] face challenges in running in real-time on embedded platforms carried by mobile robots or drones [35]. For instance, Dyna-SLAM runs at a frame rate of only 2 frames per second on the NVIDIA Jetson TX2 platform. Assuming an image size of $M \times N$, where M is the image height and N is the width, and considering N_d dynamic objects to track, building a semantic map of size K results in an overall computational complexity of $O(M \times N \times K) + O(N_d + L) + O(K)$, where L is the complexity of the dynamic object tracking algorithm. Thus, the algorithm would reach an astonishingly high level of complexity, rendering it almost non-real-time. Therefore, how to optimize the system algorithm and reduce the computational complexity to meet the real-time requirements is also an important research direction.

This paper aims to conduct in-depth research and analysis on indoor 3D reconstruction technology based on semantic visual SLAM, and in view of the existing problems in current research, this paper adds two parallel threads, target detection and semantic map reconstruction. The target detection thread utilizes an SSD [36] detection head, and by replacing the backbone network with MobileNetV3 [37], its detection speed is improved. The purpose of integrating the target detection thread is to obtain 2D semantic information through neural networks to provide prior information for the subsequent dynamic feature suppression method. Meanwhile, relying primarily on geometric information helps reduce the dependency on deep learning methods, thereby decreasing computational load and improving real-time performance. The semantic map reconstruction thread establishes intuitive global Octo-maps [38] and semantic metric maps with category and coordinate information, enabling mobile robots or drones to understand high-level environmental information for intelligent task execution. The main research contributions of this paper are as follows:

(1) Two brand new parallel threads, target detection and semantic map reconstruction, are added to ORB-SLAM2 [9] to build an RGB-D semantic visual SLAM system for indoor real-time 3D reconstruction in dynamic scenes that is more accurate and robust than the ORB-SLAM2 system;

(2) The method of suppressing dynamic features is improved. By combining semantic information and epipolar geometry constraints, geometric information obtained from epipolar geometry constraints is mainly relied on to suppress dynamic features under the premise of ensuring accuracy, rather than overly relying on deep learning methods, which better ensures the real-time performance of the system.

The experimental results demonstrate that our algorithm, running on the NVIDIA Jetson AGX Xavier, achieves an average performance improvement of approximately 97.56% in dynamic scenes compared to ORB-SLAM2, with an increase in processing time of only 6 seconds per frame. Moreover, compared to mainstream semantic visual SLAM algorithms, our algorithm exhibits a 3.5-10.5 times improvement in real-time performance. However, our algorithm still shows limitations in performance in scenarios with weak textures and large-scale scenes. The suppression of dynamic features is not sufficiently effective in low dynamic scenes. In future work, we plan to explore opportunities for improving deep learning methods or integrating other sensors such as IMU to address these limitations.

2. Algorithm Framework

The algorithm framework of this paper is based on ORB-SLAM2, using a depth camera as the visual sensor to achieve localization and mapping tasks. ORB-SLAM2 primarily consists of three threads: tracking, loop detection, and map construction. It has been widely validated using various datasets and is considered one of the most advanced and widely used visual SLAM systems to date. By utilizing ORB-SLAM2 as the framework for our system, we can effectively perform global localization and map construction tasks. Using an RGB-D camera allows for direct acquisition of depth information, aiding the algorithm in distinguishing between static and dynamic environmental elements. Additionally, it provides more accurate geometric structure and spatial distribution information of objects within the environment, thereby enhancing the reliability of localization and navigation [39].

The overall system flowchart after improvement is shown in Fig. 1. The algorithm adds target detection and improves the map reconstruction thread based on the idea of multi-thread operation, which can effectively improve the operation efficiency of the system. In the target detection thread, we use deep learning methods to obtain 2D semantic information of targets to provide prior information of dynamic objects for subsequent dynamic feature suppression methods, while avoiding excessive dependence of the system on deep learning methods, resulting in increased computation and decreased real-time performance. The semantic map reconstruction thread associates the 2D semantic information of key frames with the 3D point cloud of the map by establishing an intuitive semantic metric map to better help mobile robots or drones understand high-level environmental information to perform intelligent tasks [40]. Through the semantic map reconstruction thread, more intuitive semantic metric maps and global Octo-map can be obtained compared to the sparse point cloud maps generated by ORB-SLAM.

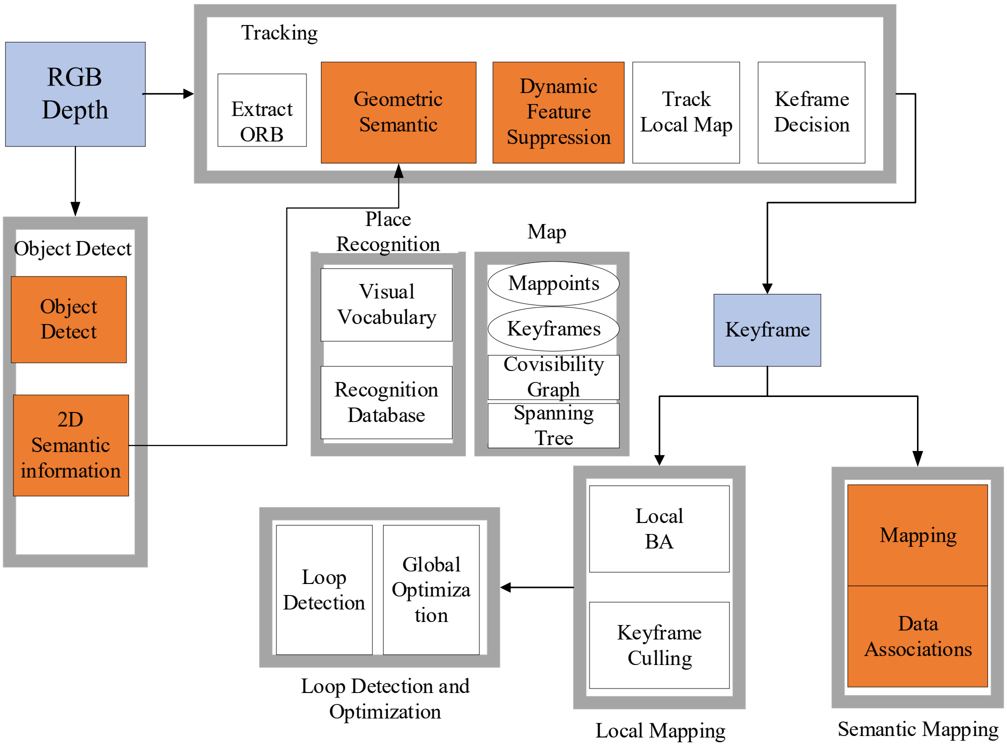


Fig. 1. Improved system framework

3. Algorithm Process

3.1. Target Detection Thread

Previous semantic visual SLAM algorithms mostly used semantic segmentation methods, which generally lack good real-time performance. Therefore, this paper's algorithm replaces semantic segmentation with object detection. To maximize the speed of object detection, the paper chooses the single-stage object detector SSD. The core idea of SSD is to transform the object detection task into a regression problem, predicting both the object categories and positions simultaneously by applying a convolutional neural network at multiple locations and scales in the image. To improve its detection performance, the paper will use a feature pyramid network to fuse feature representations from different levels. The main idea is to integrate high-level features with low-level features in the backbone network, complementing spatial information lacking in high-level features with low-level features to enhance the accuracy of the model.

At the same time, in order to provide more efficient and accurate target detection capabilities on embedded devices, this paper uses the lighter MobileNetV3 [11] feature extraction network instead of the traditional heavier VGG16 network to build the MobileNet-SSDLite network, as shown in Fig. 3. It replaces all the standard convolutions in the SSD prediction layer with depth wise separable convolutions. Unlike traditional convolutions, it divides the calculation of "convolution + channel adjustment" into two steps, reducing the amount of computation to achieve lightweight. As shown in Fig. 2, convolution is first performed with convolution kernels of the same number of channels as the input image, the size of the convolution kernel is K×K, and then N groups of 1×1 convolutions with M channels are used to adjust the number of channels of the convolution feature map obtained earlier to make the number of output feature map channels N. The ratio of the amount of computation between depth wise separable convolution and traditional convolution is:

$$\frac{HWK^2M + HWMN}{HWK^2MN} = \frac{1}{N} + \frac{1}{K^2} \tag{1}$$

where HWK^2M is the amount of computation for convolution with convolution kernels with the same number of channels as the feature map, $HWMN$ is the amount of computation for channel adjustment, and HWK^2MN is the amount of computation for traditional convolution.

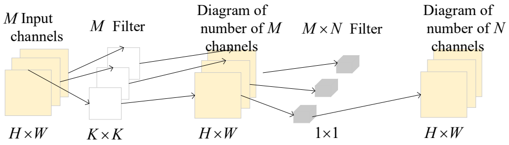

M Input channels
M Filter
Diagram of number of M channels
M × N Filter
Diagram of number of N channels
H × W
K × K
H × W
1×1
H × W

Fig. 2. Depth wise separable convolution

In the MobileNet-SSDLite network, the convolution layers 11, 13, 14_2, 15_2, 16_2, and 17_2 are responsible for the target detection work. These convolution layers have convolution kernels of different sizes and depths, which are used to slide over the feature map to generate candidate boxes and perform category classification and position regression on the candidate boxes.

Tests were conducted on the VOC2007 test set [41], and the results are shown in Table 1. It can be seen that both the real-time performance and accuracy are excellent, making it suitable for use as the object detection algorithm in this paper. (Test setup: Intel i7-11700 + NVIDIA RTX3060ti).

Table 1. Comparison of common object detection algorithms

Modules	mAP/%	FPS	MB
SSD	73.16	17.4	107.0
Tiny-YOLOV3 [42]	62.76	22.1	37.8
Tinier-YOLO [43]	67.91	26.1	10.1
MobileNetV2-SSD [44]	72.03	28.6	37.0
MobileNetV3-SSDLite	75.65	28.3	35.5

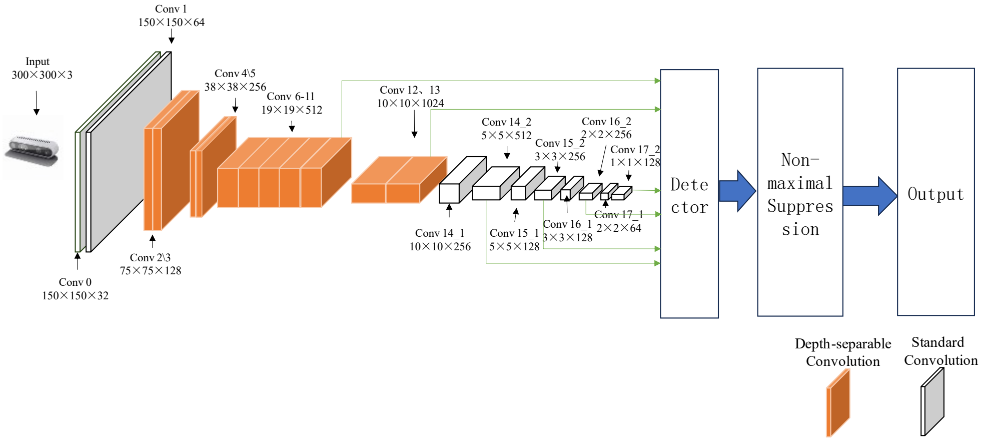


Fig. 3. MobileNet-SSDLite network architecture

3.2. Epipolar Geometry Constraints and Dynamic Feature Suppression Methods

The epipolar geometry constraint is one of the important principles of stereo image processing in the field of computer vision. As shown in Fig. 4, I_1 and I_2 represent the motion we need to find between the two cameras, with optical centers O_1 and O_2respectively. In there is a feature point P_1 in I_1, which corresponds to a feature point P_2 in I_2.The two are matched ORB features from consecutive frames. If matched correctly, the two points are projections of the same point in different planes in 3D space. In space, $\underset{O_1p_1}{\rightarrow}$ and $\underset{O_2p_2}{\rightarrow}$ intersect at point P, O_1, O_2, and point P determine a plane called the epipolar plane. The intersection e_1,e_2 of O_1,$O_2$and I_1,I_2 is called the epipole, O_1 O_2 is the baseline, and l_1, $l_2$on the intersection of the epipolar plane and I_1 I_2 is called the epipolar line [12]. The epipolar geometry constraint obtains geometric information to determine whether the feature point is dynamic mainly by calculating the fundamental matrix F or essential matrix E from the pixel positions of the matching points and then calculating the distance between the current frame and its corresponding epipolar line. The larger the value, the more likely it is to be a dynamic feature point.

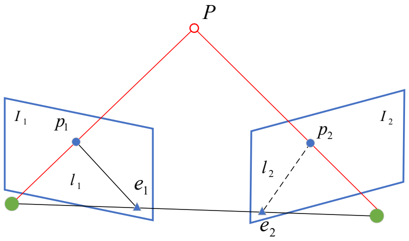


Fig. 4. Epipolar geometry constraints

Taking the fundamental matrix Fas an example, it can be expressed as:

$$P_1 = [x_1, y_1, 1], P_2 = [x_2, y_2, 1]$$

Where x and y are the coordinates of feature points in the pixel coordinate system, P_1 $P_2$are matching feature points of the same spatial point P in the previous and current frames. According to the fundamental matrix F, the epipolar line l_2 in the current frame can be calculated as follows:

$$l_2 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = FP_1 = F \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \tag{2}$$

X, Y, Z are line vectors. The epipolar geometry constraint can be expressed as:

$$P_2^T F P_1 = P_2^T l_2 = 0 \tag{3}$$

The distance P_i of the feature point d_i to the epipolar line can then be expressed as:

$$d_i = \frac{|P_i^T F P_1|}{\sqrt{X^2 + Y^2}} \tag{4}$$

In general, due to the influence of various types of noise, the feature point P_2 in the current frame cannot exactly fall on the epipolar line l_2. Assuming that when the camera moves, the spatial point P also moves accordingly to P', which matches P_2' in the current frame. If the distance d_2' to the epipolar line is less than a threshold σ, then we can usually consider point P to be a static spatial point, otherwise it is dynamic and can be removed accordingly [45]. The method based on epipolar geometry constraints is relatively simple and stable. It can achieve point-to-line mapping, reducing the number of points to be matched, thereby improving efficiency and robustness. A comparison with methods based on motion detection [46], segmentation [47], [48], and optimization [49] is shown in Table 2.

Table 2. Comparison of typical dynamic feature suppression methods

Methods	Database	Average error	Average runtime
Motion detection	KITTI	0.32	0.25s
Segmentation	KITTI	0.28	0.35s
Optimization	KITTI	0.26	0.45s
Epipolar geometry	KITTI	0.24	0.15s

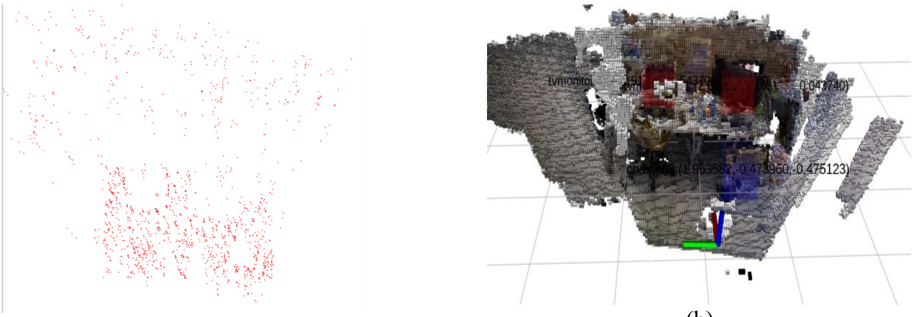
Of course, as an empirical value, the threshold σ often has problems that are difficult to define in applications. If the value is set too large, dynamic points may be ignored, and if set too small, static points may be incorrectly identified as dynamic points. The pure epipolar geometry method cannot understand the semantic information of the scene and can only process feature points according to the set σ. This obviously cannot complete the task very well. It is a good solution to first distinguish between dynamic and static targets based on the prior 2D information obtained from the target detection thread and then combine semantic information with epipolar geometry constraints to suppress dynamic features, which is also the method adopted in this paper. The idea of selecting σ in this paper is very simple. The value is directly chosen to be a threshold that can significantly suppress dynamic feature points in practical engineering. In static scenes, $\sigma = 0$, and in dynamic environments, it is set to 50. Additionally, dynamic weight values ϖ are set, with higher probabilities of movement assigned higher values (e.g., 5 for humans) and objects less likely to move assigned lower values (e.g., 2 for chairs).

Firstly, the target detection thread observes whether dynamic objects are detected in the current frame. If they do not exist or if the current feature point is not within the bounding box of a detected dynamic object, the offset distance of the current feature point is calculated and compared directly with the standard empirical threshold σ to determine whether to remove it. If dynamic objects are detected and the current feature point is within their bounding box, the offset distance of the current feature point is calculated and compared with the product of the dynamic weight value ϖ and the standard empirical threshold σ. Based on the result, suppression is decided.

The prior semantic information obtained by the target detection thread endows the method based on geometric information with the ability to understand the environment at a higher level. Adopting different dynamic weight values for regions with different probabilities of movement overcomes the difficulty of selecting the empirical threshold. Since it does not overly rely on semantic information, this suppression algorithm addresses the weakness of poor real-time performance caused by relying solely on deep learning methods.

3.3. Semantic Map Reconstruction Thread

In general, three-dimensional point cloud maps constructed from point cloud data contain a large amount of unnecessary information in the environment, occupying significant memory space, with poor readability, and sparse point cloud maps cannot be directly used for robot motion planning and other tasks. The approach in this paper is that immediately after a new keyframe arrives, the Mapping thread utilizes its depth image and pose to generate a three-dimensional ordered point cloud and publishes it to ROS [50], constructing an Octo-map. The Octo-map, superior in spatial representation, memory efficiency, and query efficiency compared to point clouds, can provide obstacle and surface information directly applicable to navigation. As shown in Fig. 5, the comparison between the Octo-map and sparse point cloud clearly indicates that the Octo-map provides more information. Addressing the lack of semantic information in the Octo-map, this paper utilizes the 2D bounding boxes obtained during the suppression of dynamic features to acquire 3D point cloud information within the bounding box regions, aiming to obtain 3D semantic object information. By calculating the average depth of the point cloud within the bounding box and comparing it with the depth of the point cloud, if the difference is small, it is considered as a cluster of point clouds belonging to the target object, which is then retained. The size and spatial coordinates are calculated to obtain 3D object semantic information, continuously updating the 3D semantic object information database, and using it to establish a semantic metric map. The semantic metric map contains semantic information such as object categories and coordinates, which can assist mobile robots in performing intelligent navigation, target grasping, and other semantic operations. The process of obtaining 3D semantic object information is shown in Fig. 6.



(a) (b)

Fig. 5. Comparison between Coefficient (a). Sparse Point Cloud Map and (b). Octo-map

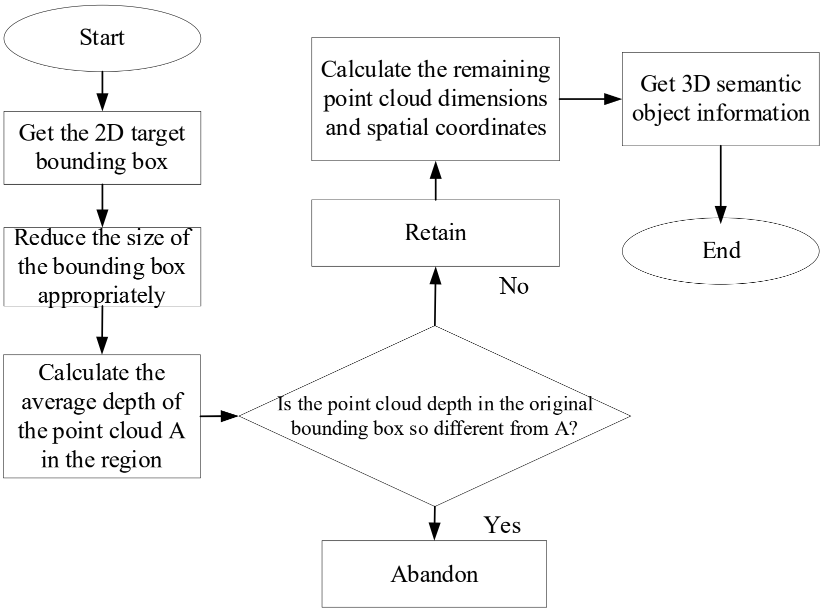

Start
Get the 2D target bounding box
Reduce the size of the bounding box appropriately
Calculate the average depth of the point cloud A in the region
Is the point cloud depth in the original bounding box so different from A?
No
Retain
Calculate the remaining point cloud dimensions and spatial coordinates
Get 3D semantic object information
End
Yes
Abandon

Fig. 6. Process of obtaining 3D semantic object information

4. Experimental Results Analysis

In this section, we will conduct experiments to validate the improvements described earlier and verify the effectiveness of the enhanced algorithm. The overall experiment is primarily conducted on the NVIDIA Jetson AGX Xavier edge computing device, which boasts powerful computing capabilities and low power consumption characteristics. With 32 TOPs of computing power and power consumption ranging from 10 to 30W, it is more suitable as a computing platform for mobile robots or drones compared to other workstation platforms. Using an RGB-D camera as its sensor, it can directly measure depth information, eliminating scale drift issues. The algorithm proposed in this paper is designed for indoor environments, so there is little need to consider drawbacks such as depth cameras being susceptible to light interference.

The steps of the experiment are: first, the performance of the improved SLAM system is evaluated on public datasets. Then the effectiveness of the dynamic feature suppression method is verified. Finally, the real-time performance of the improved system is evaluated, and real-time 3D reconstruction of indoor scenes on the TUM dataset is performed.

4.1. Algorithm Performance Evaluation

This paper uses the publicly available TUM RGB-D dataset [51], which is a widely used indoor robot vision dataset provided by the Technical University of Munich, Germany and is widely used for evaluating the performance of visual SLAM systems. In the experiment, we selected three high-dynamic sequences fr3_walking_xyz, fr3_walking_halfsphere, fr3_walking_static (abbreviated as fwx, fwh, fws respectively) and the low-dynamic sequence fr3_ sitting_static (abbreviated as fss). There are two main indicators for evaluating errors in the SLAM field: absolute trajectory error (ATE) and relative pose error (RPE). This paper will compare them with ORB-SLAM2 as evaluation metrics, and the trajectory evaluation tool provided in TUM rgbd_bench_tools is used as the evaluation tool, with the camera sensor set to RGB-D camera, and the average of more than 30 experiments is taken. The comparison results of the experimental sequences are shown in Table 3, Table 4, and Table 5 (the results retain 4 decimal places and the improvement results retain 2 decimal places).

From the following chart results, it can be clearly observed that the running effect of the improved system in this paper in the high-dynamic fr3_walking sequences is much better than that of ORB-SLAM2, and in the low-dynamic fr3_sitting sequence, since the target dynamics in the scene of the dataset is low, and the poses of most objects are relatively fixed, the performance of ORB-SLAM2 on it is already quite good, and the improvement of the improved system in this paper is also limited, with an improvement of only 32.94%.

Table 3. Absolute trajectory error comparison

Sequences	ORB-SLAM2		Ours		Improvement (%)	
	RMSE	STD	RMSE	STD	RMSE	STD
fwx	0.7404	0.3759	0.0131	0.0063	98.23%	98.32%
fwh	0.8753	0.4077	0.0261	0.0162	97.02%	96.03%
fws	0.4080	0.1747	0.0071	0.0029	98.26%	98.34%
fss	0.0085	0.0041	0.0057	0.0028	32.94%	31.71%

Table 4. Relative pose error comparison (rotation)

Sequences	ORB-SLAM2		Ours		Improvement (%)	
	RMSE	STD	RMSE	STD	RMSE	STD
fwx	6.7516	4.1236	0.3268	0.2056	95.16%	95.06%
fwh	14.0656	8.3239	0.7867	0.3659	94.40%	95.60%
fws	5.4789	3.7841	0.2687	0.1065	95.10%	97.19%
fss	0.3807	0.1460	0.2576	0.1052	32.34%	27.95%

Table 5. Relative pose error comparison (translation)

Sequences	ORB-SLAM2		Ours		Improvement (%)	
	RMSE	STD	RMSE	STD	RMSE	STD
fwx	0.4519	0.2198	0.0184	0.0103	95.93%	95.31%
fwh	0.5867	0.3584	0.0286	0.0144	95.16%	95.98%
fws	0.3382	0.2107	0.0094	0.0047	97.28%	97.77%
fss	0.0104	0.0055	0.0067	0.0034	35.58%	38.18%

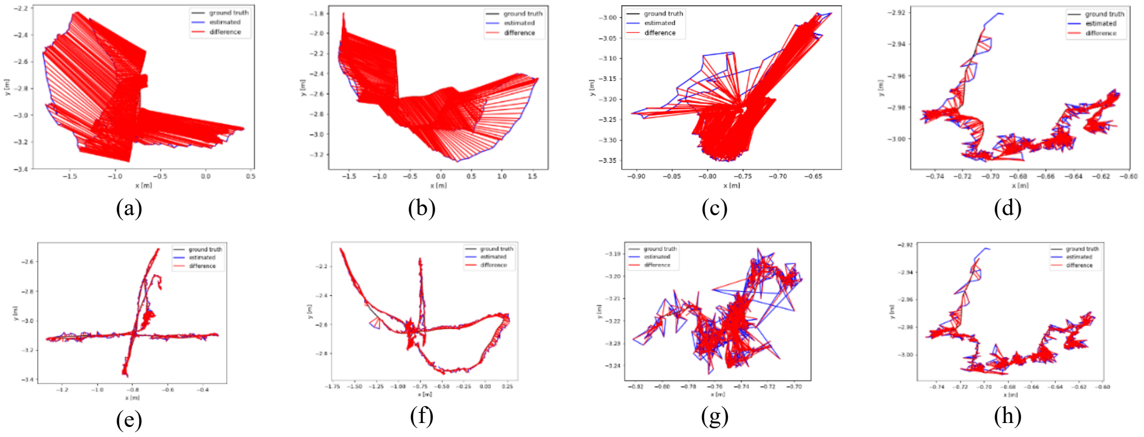
To describe the comparison more intuitively, the ATE and RPE comparison graphs are drawn using the trajectory evaluation tool of TUM, as shown in Fig. 7 and Fig. 8. Among them, in the ATE result graph, the black line is the true trajectory, the blue line is the estimated trajectory, and the red line is the error between the two. The shorter the red line, the smaller the error and the higher the accuracy of the system. RPE is used to calculate the difference in pose changes between the same two timestamps to estimate the system's drift.

The Bonn RGB-D Dynamic Dataset, provided by the University of Bonn, consists of 24 dynamic sequence datasets used for evaluating RGB-D SLAM [52]. To verify the algorithm's generalization performance, we conducted additional experiments on this dataset, selecting 5 representative sequences. The comparative performance with ORB-SLAM2 is shown in Table 6.

In order to further verify the effectiveness of the system, after comparison with ORB-SLAM2, this paper also compares with mainstream semantic visual SLAM algorithms such as those shown in Table 7. It can be seen from the results that although the accuracy of this paper's algorithm is slightly lower than that of DynaSLAM, the reason is that DynaSLAM uses MASK R-CNN for pixel-level semantic segmentation, which is not adopted in this paper. But this also makes the real-time performance of DynaSLAM extremely poor, almost impossible to apply to embedded platforms such as drones. The comparison with other algorithms can prove the advancement of the average accuracy of the improved algorithm in this paper.

Table 6. Absolute trajectory error comparison (Bonn datasets)

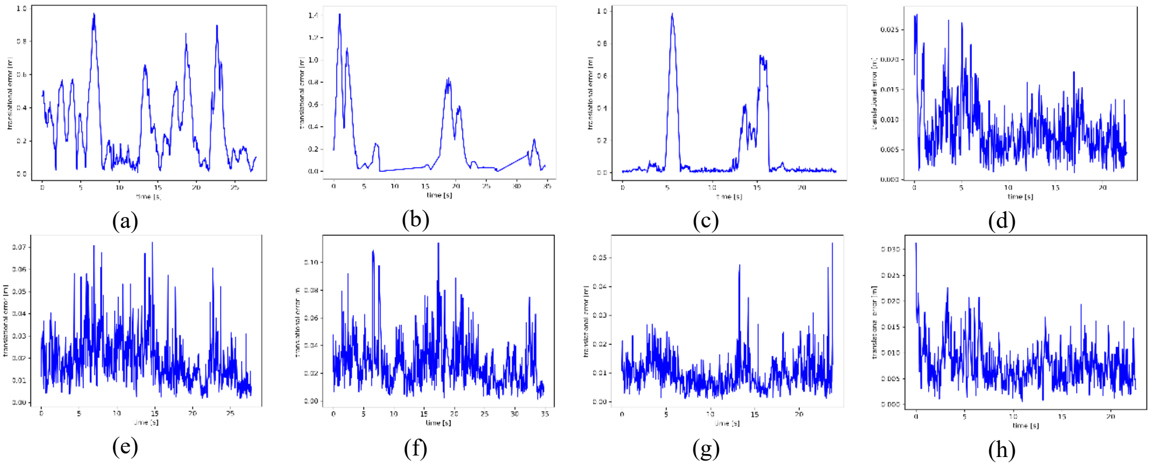
Sequences	ORB-SLAM2		Ours		Improvement (%)	
	RMSE	STD	RMSE	STD	RMSE	STD
Crowd1	0.9624	0.6571	0.0212	0.0137	97.80	97.91
Crowd2	1.4687	0.7576	0.0452	0.0347	96.92	95.42
Person_tracking1	0.8675	0.4687	0.0367	0.0125	95.76	97.32
Person_tracking2	1.0824	0.5878	0.0287	0.0143	97.35	97.56
Synchronous2	1.5789	0.5782	0.0146	0.0113	99.07	98.05



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 7. Absolute trajectory error comparison (a). fwx/ORB-SLAM2, (b). fwh/ORB-SLAM2, (c). fws/ORB-SLAM2, (d). fss/ORB-SLAM2, (e). fwx/Ours, (f). fwh/Ours, (g). fws/Ours, (h). fss/Ours



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 8. Relative pose error comparison (a). fwx/ORB-SLAM2, (b). fwh/ORB-SLAM2, (c). fws/ORB-SLAM2, (d). fss/ORB-SLAM2, (e). fwx/Ours, (f). fwh/Ours, (g). fws/Ours, (h). fss/Ours

Table 7. Comparison with mainstream semantic visual SLAM algorithms

Sequences	DynaSLAM	DetectSLAM [53]	DS-SLAM	System [54]	RDS-SLAM	Ours
	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
fwx	0.0148	0.0248	0.0253	0.0182	0.0427	0.0131
fwh	0.0201	0.0573	0.0324	0.0263	0.0607	0.0231
fws	0.0061	0.0125	0.0079	0.0091	0.0106	0.0061

4.2. Dynamic Feature Suppression Method Effect Experiments

This paper combines semantic information obtained from target detection and epipolar geometry constraints to eliminate dynamic feature points and achieve dynamic feature suppression effects. In order to verify the effectiveness of the dynamic feature suppression method, comparative experiments will be carried out in this section.

First, we set up a control group consisting of the ORB-SLAM2 system and an improved system using only epipolar geometry constraints, and an experimental group consisting of our system. The experimental results, as shown in Fig. 9, clearly demonstrate that the ORB-SLAM2 system has minimal effect on suppressing dynamic features, while the improved system using only epipolar geometry constraints mistakenly detects many feature points, as shown in Fig. 9 (c) where the feature points of the monitor are ignored by the system. Our system almost eliminates all dynamic feature points while retaining as many static feature points as possible, such as the monitor feature points mistakenly removed by the epipolar geometry constraints. This experiment demonstrates the effectiveness of our method for suppressing dynamic features.

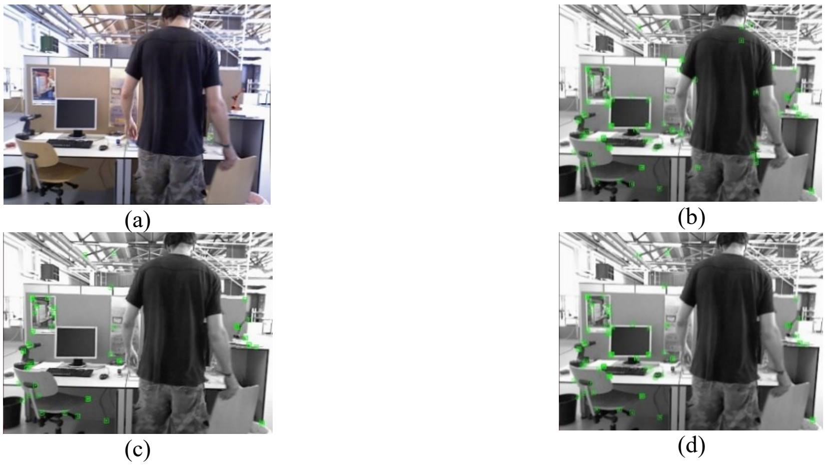


Fig. 9. Comparison of dynamic feature suppression effects (a). RGB, (b). ROB-SLAM2, (c). Epipolar geometric constraints, (d). Ours

4.3. System Real-Time Performance Evaluation

The real-time performance of the SLAM system is crucial for many application scenarios, ensuring that the system can respond and adapt to changes in dynamic environments in a timely manner, thereby providing accurate, reliable and efficient localization, mapping and navigation capabilities.

This section evaluates the system's real-time performance. The evaluation indicator is the average time required for the system to process each frame. Comparative experiments were also conducted with other mainstream systems. The results are shown in Table 8. The average time for DynaSLAM to process each frame is very large, almost unable to meet the requirements of real-time operation. The real-time performance of system [55], a semantic SLAM system based on target detection, on embedded platforms is also worrying. Compared with ORB-SLAM2, although the average time is slightly increased, it can still fully meet the real-time requirements, and the accuracy is much higher than ORB-SLAM2.

Table 8. Average frame processing time comparison of systems

System	Average Frame Processing Time
ORB-SLAM2	60.43
DynaSLAM	232.51
System [55]	688.56
Ours	66.46

4.4. Indoor Real-time 3D Reconstruction Experiments

Semantic visual SLAM plays an important role and value in indoor 3D reconstruction. It not only helps to construct accurate and structured indoor maps, but also provides rich semantic information for subsequent applications to improve the intelligence, adaptability and user experience of the system.

In order to verify the actual 3D reconstruction effect, this paper verifies the algorithm on the TUM dataset and performs real-time 3D reconstruction experiments on the high-dynamic sequence fr3_walking, the low-dynamic sequence fr3_sitting, and the large-scale static sequence fr3_long_office. Establish global Octo-maps and semantic metric maps, as shown in Fig. 10, Fig. 11, and Fig. 12. The coordinates displayed on the maps are converted based on the origin where the semantic visual SLAM system runs.

Furthermore, experiments were conducted on the challenging lifelong SLAM dataset OpenLORIS dynamic dataset [56], as shown in Fig. 13. This dataset closely resembles real-world complex dynamic scenes and presents significant challenges. By performing real-time 3D reconstruction experiments on three different environments from the TUM dataset and the OpenLORIS dataset, this study validates that our algorithm can establish maps with low overlap, good readability, and high-level semantic information in indoor 3D reconstruction tasks. These maps can assist mobile robots in localization, obstacle avoidance, and enable them to perform higher-level tasks in more complex environments.

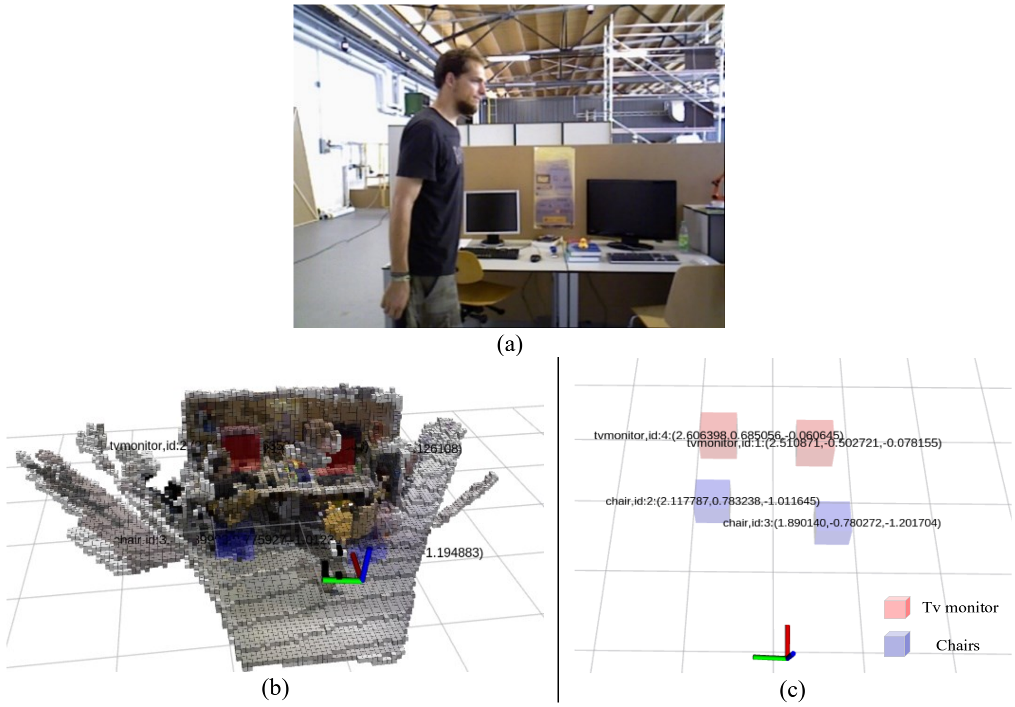

(a)
(b)
(c)
tvmonitor,id:4:(2.606398,0.685056,-0.060645)
tvmonitor,id:1:(2.510871,-0.502721,-0.078155)
chair,id:2:(2.117787,0.783238,-1.011645)
chair,id:3:(1.890140,-0.780272,-1.201704)
Tv monitor
Chairs

Fig. 10. Real-time 3D reconstruction in high dynamic scene (a). fr3_walking sequence, (b). Global Octo-map, (c). Semantic metric map

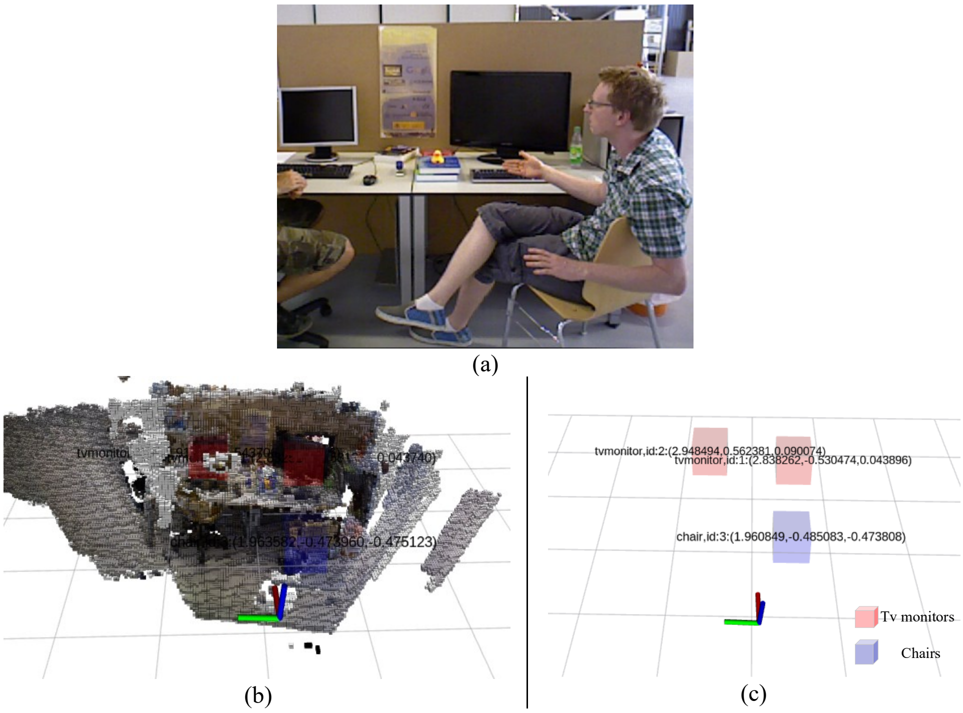
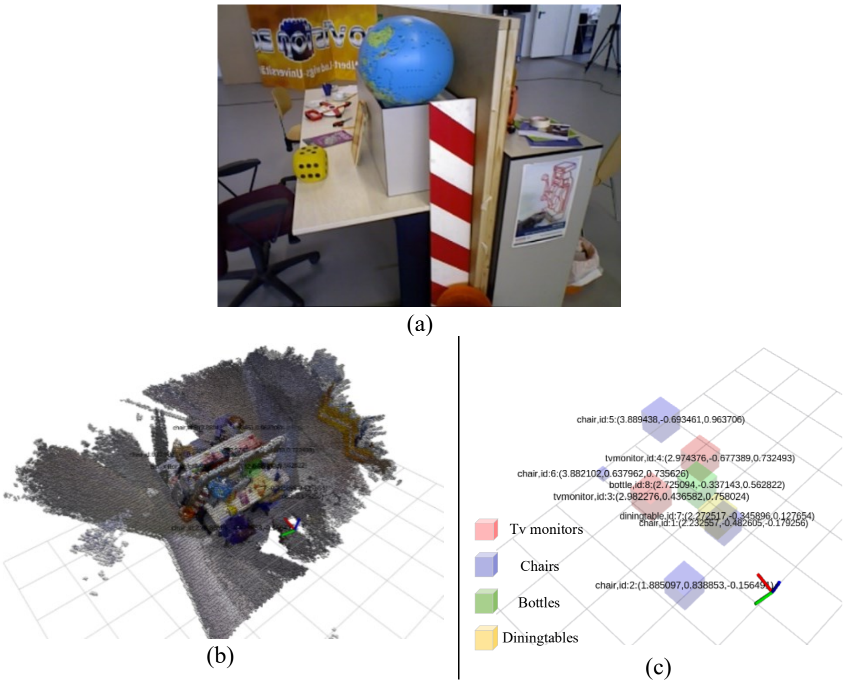

(a)
(b)
(c)
tvmonitor,id:2:(2.948494,0.562381,0.090074)
tvmonitor,id:1:(2.838262,-0.530474,0.043896)
chair,id:3:(1.960849,-0.485083,-0.473808)
Tv monitors
Chairs

Fig. 11. Real-time 3D reconstruction in low dynamic scene (a). fr3_sitting sequence, (b). Global Octo-map, (c). Semantic metric map

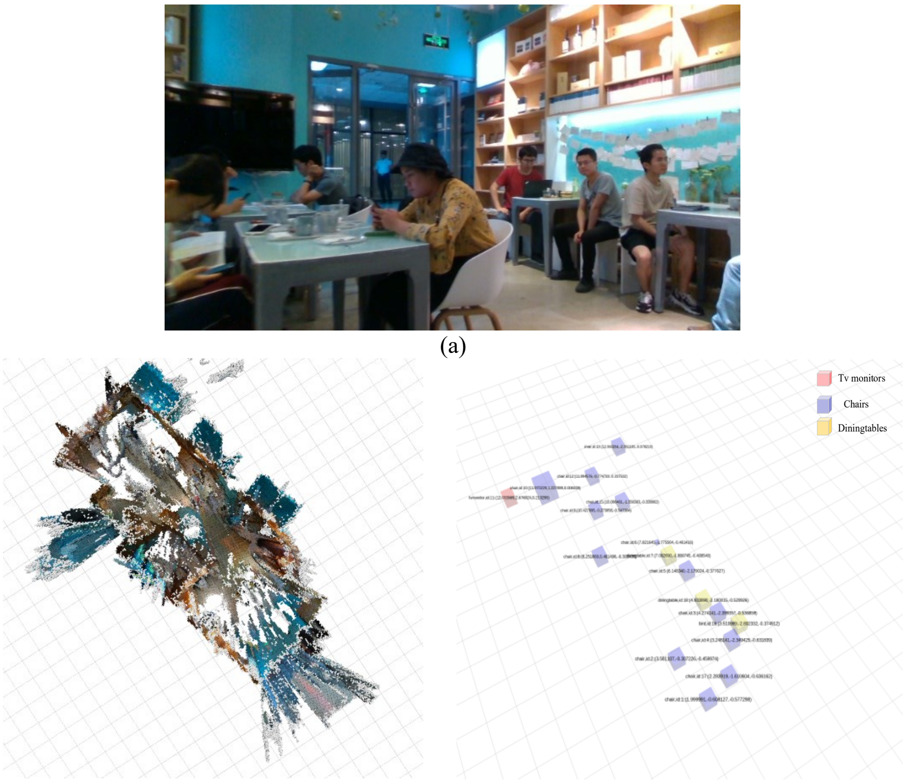

chair,id:5:(3.889438,-0.693461,0.963706)
tvmonitor,id:4:(2.974376,-0.677389,0.732493)
chair,id:6:(3.882102,0.637962,0.735626)
bottle,id:8:(2.725094,-0.337143,0.562822)
tvmonitor,id:3:(2.982276,0.436582,0.758024)
diningtable,id:7:(2.272517,-0.345896,0.127654)
chair,id:1:(2.232557,-0.482605,-0.179256)
chair,id:2:(1.885097,0.838853,-0.15649)
Tv monitors
Chairs
Bottles
Diningtables

(a)

(b)

(c)

Fig. 12. Real-time 3D reconstruction in static scene (a). fr3_sitting sequence, (b). Global Octo-map, (c). Semantic metric map



(a)

(b)

Fig. 13. Real-time 3D reconstruction in OpenLORIS datasets (a). OpenLORIS cafe1-2 sequence, (b). Global Octo-map, (c). Semantic metric map

5. Conclusion

The paper proposes a real-time indoor 3D reconstruction system based on semantic visual SLAM, primarily addressing the challenge of 3D reconstruction in dynamic indoor environments.

Improvements over the ORB-SLAM2 system include the addition of two new parallel threads—object detection and semantic map reconstruction. The core idea of dynamic feature suppression in this paper is the integration of semantic and geometric information, and experimental results demonstrate significant effectiveness in suppressing dynamic features. The system evaluation on high-dynamic sequences from the TUM dataset shows an average improvement of 97.56% over ORB-SLAM2, and on the Bonn dataset, an average improvement of 96.67%, while still maintaining good real-time performance. Compared to some mainstream semantic SLAM open-source systems, the real-time performance is improved by 3.5-10.5 times while maintaining accuracy. During 3D reconstruction of indoor scenes from the TUM dataset and the more challenging OpenLORIS dataset, the system can establish semantic metric maps and global Octo-maps suitable for mobile robots to perform high-level tasks.

The algorithm also proves that appropriately integrating deep learning methods can make traditional visual SLAM systems perform better. Compared to semantic segmentation methods, object detection has stronger real-time capabilities, leading to positive effects in practical engineering applications.

However, the system also has some limitations, such as insufficient reconstruction accuracy in low-texture and large-scale scenes, ineffective dynamic feature suppression methods in low-dynamic scenes, and limited ability to detect small objects, resulting in lower-quality semantic metric maps. Future considerations include improving deep learning methods by replacing them with YOLOV5s or other methods and conducting training to enhance small object detection capabilities, or integrating IMU and other sensors to improve pose estimation performance, to adapt to more complex and high-speed environments.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This work was Supported by Chinese Universities Industry-Academia-Research Innovation Fund Project (2021ZYA11002) Sichuan University of Science & Engineering Graduate Innovation Fund (No. Y2022172).

Acknowledgment: This work was Supported by Chinese Universities Industry-Academia-Research Innovation Fund Project (2021ZYA11002), Sichuan University of Science & Engineering Graduate Innovation Fund (No. Y2022172).

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] B. Gong, Z. Zhu, C. Yan, Z. Shi, and F. Xu, "PlaneFusion: Real-Time Indoor Scene Reconstruction With Planar Prior," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, pp. 4671-4684, 2022, https://doi.org/10.1109/TVCG.2021.3099480.

[2] Z. Xi, M. Rui, G. Rui, and H. Qi, "Phase-SLAM: Phase Based Simultaneous Localization and Mapping for Mobile Structured Light Illumination Systems," *IEEE Robotics and Automation Letters*, vol. 7, pp. 6203-6210, 2022, https://doi.org/10.1109/LRA.2022.3162024.

[3] S. Hajira, M. Reza, and M. Hussan, "Neural Network-Based Recent Research Developments in SLAM for Autonomous Ground Vehicles: A Review," *IEEE Sensors Journal*, vol. 23, pp. 13829-13858, 2023, https://doi.org/10.1109/JSEN.2023.3273913.

[4] H. Y. Chia and H. L. Min, "Robust 3D Reconstruction Using HDR-Based SLAM," *IEEE Access*, vol. 9, pp. 16568-16581, 2021, https://doi.org/10.1109/ACCESS.2021.3051257.

[5] M. Francisco, C. J. M., M. Magdalena, and F. Camino, "Augmented Reality Based on SLAM to Assess Spatial Short-Term Memory," *IEEE Access*, vol. 7, pp. 2453-2466, 2019, https://doi.org/10.1109/ACCESS.2018.2886627.

[6] P. M. C. Joao, C. S. Antonio, and R. R. S. Silvio, "A mapping of visual SLAM algorithms and their applications in augmented reality," *in 2020 22nd Symposium on Virtual and Augmented Reality (SVR)*, pp. 20-29, 2020, https://doi.org/10.1109/SVR51698.2020.00019.

[7] J. Fuentes-Pacheco, J. Ascencio, and J. Rendon-Mancha, "Visual Simultaneous Localization and Mapping: A Survey," *Artificial Intelligence Review*, vol. 43, pp. 55-81, 2015, https://doi.org/10.1007/s10462-012-9365-8.

[8] G. Jeremias, O. Eugenio, R. Francisco, and S. Carlos, "Mapping the Landscape of SLAM Research: A Review," *IEEE Latin America Transactions*, vol. 21, pp. 1313-1336, 2023, https://doi.org/10.1109/TLA.2023.10305240.

[9] G. Yang, Y. Wang, J. Zhi, W. Liu, Y. Shao, and P. Peng, "A Review of Visual Odometry in SLAM Techniques," *in 2020 International Conference on Artificial Intelligence and Electromechanical Automation (AIEA)*, pp. 332-336, 2020, https://doi.org/10.1109/AIEA51086.2020.00075.

[10] J. Wang and F. Yang, "A Review of Vision SLAM-based Closed-loop Inspection," *in 2023 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 507-512, 2023, https://doi.org/10.1109/ICMA57826.2023.10215583.

[11] Y. Ying, Z. Wei, H. Shang, and S. Liang, "Dense Scene 3D Reconstruction Based on Semantic Information of Indoor Environment: A Review," *in 2021 International Conference on Networking Systems of AI (INSAI)*, pp. 110-117, 2021, https://doi.org/10.1109/INSAI54028.2021.00030.

[12] X. Lin, Y. Huang, D. Sun, Y. Lin, E. Brendan, M. E. Ryan, and G. Manni, "A Robust Keyframe-Based Visual SLAM for RGB-D Cameras in Challenging Scenarios," *IEEE Access*, vol. 11, pp. 97239-97249, 2023, https://doi.org/10.1109/ACCESS.2023.3312062.

[13] C. Campos, E. Richard, J. G. R. Juan, M. M. M. Jose, and D. T. Juan, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," *IEEE Transactions on Robotics*, vol. 37, pp. 1874-1890, 2021, https://doi.org/10.1109/TRO.2021.3075644.

[14] M. Raul and D. T. Juan, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, pp. 1255-1262, 2017, https://doi.org/10.1109/TRO.2017.2705103.

[15] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, pp. 99-110, 2006, https://doi.org/10.1109/MRA.2006.1638022.

[16] L. Zhao, B. Wei, L. Li, and L. Xu, "A Review of Visual SLAM for Dynamic Objects," *in 2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA)*, pp. 1080-1085, 2022, https://doi.org/10.1109/ICIEA54703.2022.10006191.

[17] S. Chen, C. Sun, S. Zhang, and D. Zhang, "SG-SLAM: A Real-Time RGB-D Visual SLAM Toward Dynamic Scenes With Semantic and Geometric Information," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1-12, 2023, https://doi.org/10.1109/TIM.2022.3228006.

[18] K. Chen, J. Liu, Q. Chen, Z. Wang, and J. Zhang, "Accurate Object Association and Pose Updating for Semantic SLAM," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 25169-25179, 2022, https://doi.org/10.1109/TITS.2021.3136918.

[19] Y. Liu and M. Jun, "RDS-SLAM: Real-Time Dynamic SLAM Using Semantic Segmentation Methods," *IEEE Access*, vol. 9, pp. 23772-23785, 2021, https://doi.org/10.1109/ACCESS.2021.3050617.

[20] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," *Robotics and Autonomous Systems*, vol. 66, pp. 86-103, 2015, https://doi.org/10.1016/j.robot.2014.12.006.

[21] R. Teng, Y. Liang, J. Zhang, D. Tang, and H. Li, "RS-SLAM: A Robust Semantic SLAM in Dynamic Environments Based on RGB-D Sensor," *IEEE Sensors Journal*, vol. 21, pp. 20657-20664, 2021, https://doi.org/10.1109/JSEN.2021.3099511.

[22] K. Chen, J. Zhang, J. Liu, Q. Tong, R. Liu, and S. Chen, "Semantic Visual Simultaneous Localization and Mapping: A Survey," *ARXIV*, 2022, https://arxiv.org/abs/2209.06428v1.

[23] H. Pu, J. Luo, G. Wang, T. Huang, H. Liu, and J. Luo, "Visual SLAM Integration With Semantic Segmentation and Deep Learning: A Review," *IEEE Sensors Journal*, vol. 23, pp. 22119-22138, 2023, https://doi.org/10.1109/JSEN.2023.3306371.

[24] A. P. Julio, S. Jared, C. Henry, A. Nikolay, I. Vadim, C. Luca, and A. C. Jose, "A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers," *IEEE Transactions on Robotics*, vol. 39, pp. 1686-1705, 2023, https://doi.org/10.1109/TRO.2023.3248510.

[25] F. Min, Z. Wu, D. Li, G. Wang, and N. Liu, "COEB-SLAM: A Robust VSLAM in Dynamic Environments Combined Object Detection, Epipolar Geometry Constraint, and Blur Filtering," *IEEE Sensors Journal*, vol. 23, pp. 26279-26291, 2023, https://doi.org/10.1109/JSEN.2023.3317056.

[26] K. Wang, S. Ma, J. Chen, F. Ren, and J. Lu, "Approaches, Challenges, and Applications for Deep Visual Odometry: Toward Complicated and Emerging Areas," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, pp. 35-49, 2022, https://doi.org/10.1109/TCDS.2020.3038898.

[27] K. Liu, H. Zhang, Y. Liu, and Y. Wang, "Dynamic Object Removal based on Deep Learning and Multi-view Geometry," *in 2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*, pp. 863-868, 2022, https://doi.org/10.1109/ICFTIC57696.2022.10075301.

[28] Y. Wang, B. Zhang, P. Li, T. Cao, and B. Zhang, "Dynamic Object Separation and Removal in 3D Point Cloud Map Building," *in 2022 6th International Conference on Robotics and Automation Sciences (ICRAS)*, pp. 247-252, 2022, https://doi.org/10.1109/ICRAS55217.2022.9842041.

[29] B. Berta, M. F. Jose, C. Javier, and N. Jose, "DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes," *IEEE Robotics and Automation Letters*, vol. 3, pp. 4076-4083, 2018, https://doi.org/10.1109/LRA.2018.2860039.

[30] C. Yu, Z. Liu, X. Liu, F. Xie, Y. Yi, Q. Wei, and Q. Fei, "DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments," *in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1168-1174, 2018, https://doi.org/10.1109/IROS.2018.8593691.

[31] J. Chang, N. Dong, and D. Li, "A Real-Time Dynamic Object Segmentation Framework for SLAM System in Dynamic Scenes," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1-9, 2021, https://doi.org/10.1109/TIM.2021.3109718.

[32] X. Cui, C. Liu, and J. Wang, "3D Semantic Map Construction Using Improved ORB-SLAM2 for Mobile Robot in Edge Computing Environment," *IEEE Access*, vol. 8, pp. 67179-67191, 2020, https://doi.org/10.1109/ACCESS.2020.2983488.

[33] K. He, G. Gkioxari, D. Piotr, and G. Ross, "Mask R-CNN," *in 2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2980-2988, 2017, https://doi.org/10.1109/ICCV.2017.322.

[34] B. Vijay, K. Alex, and C. Roberto, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481-2495, 2017, https://doi.org/10.1109/TPAMI.2016.2644615.

[35] N. Yoshikatsu and S. Hideo, "Efficient Object-Oriented Semantic Mapping With Object Detector," *IEEE Access*, vol. 7, pp. 3206-3213, 2019, https://doi.org/10.1109/ACCESS.2018.2887022.

[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *In Computer Vision–ECCV 2016: 14th European Conference*, pp. 21-37, 2016, https://doi.org/10.1007/978-3-319-46448-0_2.

[37] H. Andrew *et al.*, "Searching for MobileNetV3," *in 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314-1324, 2019, https://doi.org/10.1109/ICCV.2019.00140.

[38] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: an efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, pp. 189-206, 2013, http://dx.doi.org/10.1007/s10514-012-9321-0.

[39] D. Li, S. Liu, W. Xiang, Q. Tan, K. Yuan, Z. Zhang, and Y. Hu, "A SLAM System Based on RGBD Image and Point-Line Feature," *IEEE Access*, vol. 9, pp. 9012-9025, 2021, https://doi.org/10.1109/ACCESS.2021.3049467.

[40] K. Liu, Z. Fan, M. Li, and S. Zhang, "Object-aware Semantic Mapping of Indoor Scenes using Octomap," *in 2019 Chinese Control Conference (CCC)*, pp. 8671-8676, 2019, https://doi.org/10.23919/ChiCC.2019.8865848.

[41] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303-338, 2010, http://dx.doi.org/10.1007/s11263-009-0275-4.

[42] A. Pranav, R. Pratibha, and K. Manoj, "YOLO v3-Tiny: Object Detection and Recognition using one stage improved model," *in 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 687-694, 2020, https://doi.org/10.1109/ICACCS48705.2020.9074315.

[43] W. Fang, L. Wang, and P. Ren, "Tinier-YOLO: A Real-Time Object Detection Method for Constrained Environments," *IEEE Access*, vol. 8, pp. 1935-1944, 2020, https://doi.org/10.1109/ACCESS.2019.2961959.

[44] L. Chen *et al*., "Deep Neural Network Based Vehicle and Pedestrian Detection for Autonomous Driving: A Survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, pp. 3234-3246, 2021, https://doi.org/10.1109/TITS.2020.2993926.

[45] R. Gao, Z. Li, J. Li, B. Li, J. Zhang, and J. Liu, "Real-Time SLAM Based on Dynamic Feature Point Elimination in Dynamic Environment," *IEEE Access*, vol. 11, pp. 113952-113964, 2023, https://doi.org/10.1109/ACCESS.2023.3324146.

[46] H. Liu, G. Liu, G. Tian, S. Xin, and Z. Ji, "Visual SLAM Based on Dynamic Object Removal," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 596-601, 2019, https://doi.org/10.1109/ROBIO49542.2019.8961397.

[47] H. Thorsten and A. Ayoub, "Pixel-Wise Motion Segmentation for SLAM in Dynamic Environments," *IEEE Access*, vol. 8, pp. 164521-164528, 2020, https://doi.org/10.1109/ACCESS.2020.3022506.

[48] C Shao, L. Zhang, and W. Pan, "Faster R-CNN Learning-Based Semantic Filter for Geometry Estimation and Its Application in vSLAM Systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 5257-5266, 2022, https://doi.org/10.1109/TITS.2021.3052812.

[49] J. Cheng, C. Wang. and Q. H. M. Max, "Robust Visual Localization in Dynamic Environments Based on Sparse Motion Removal," *IEEE Transactions on Automation Science and Engineering*, vol. 17, pp. 658-669, 2020, https://doi.org/10.1109/TASE.2019.2940543.

[50] M. Quigley *et al*., "ROS: an open-source Robot Operating System," *in ICRA Workshop Open Source Softw*, pp. 5, 2009, http://dx.doi.org/10.13140/RG.2.2.28424.93446.

[51] S. Jurgen, E. Nikolas, E. Felix, B. Wolfram, and C. Daniel, "A benchmark for the evaluation of RGB-D SLAM systems," *in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573-580, 2012, https://doi.org/10.1109/IROS.2012.6385773.

[52] P. Emanuele, B. Jens, L. Philipp, G. Philippe, and S. Cyrill, "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals," *in 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7855-7862, 2019, https://doi.org/10.1109/IROS40897.2019.8967590.

[53] F. Zhong, S. Wang, Z. Zhang, C. Chen, and Y. Wang, "Detect-SLAM: Making Object Detection and SLAM Mutually Beneficial," *in 2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1001-1010, 2018, https://doi.org/10.1109/WACV.2018.00115.

[54] D. Wu, B. Xie, and C. Tao, "3D Semantic VSLAM of Dynamic Environment Based on YOLACT," *Mathematical Problems in Engineering*, vol. 2022, pp. 1-12, 2022, http://dx.doi.org/10.1155/2022/7307783.

[55] W. Wu, L. Guo, H. Gao, Z. You, Y. Liu, and Z. Chen, "YOLO-SLAM: A semantic SLAM system towards dynamic environment with geometric constraint," *Neural Computing and Applications*, vol. 34, pp. 6011-6026, 2022, https://doi.org/10.1007/s00521-021-06764-3.

[56] X. Shi *et al*., "Are We Ready for Service Robots? The OpenLORIS-Scene Datasets for Lifelong SLAM," *in 2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3139-3145, 2020, https://doi.org/10.1109/ICRA40945.2020.9196638.