Journal of Computing and Electronic Information Management
ISSN: 2413-1660 | Vol. 20, No. 3, 2026

A Method for Localization and Dense Mapping in Indoor Dynamic Environments Based on an Improved ORB-SLAM2 Algorithm

Wanlei Liu, Yao Zhao*

School of Physics & Electronic Information Engineering, Henan Polytechnic University, Jiaozuo, 454003, China.
*** Corresponding author:** Yao Zhao (Email: zhaoyao@hpu.edu.cn)

Abstract: To address the difficulty of ORB-SLAM2 in constructing dense maps in complex indoor dynamic environments, this paper proposes an improved method for localization and dense mapping in dynamic indoor scenes. The proposed approach integrates a semantic segmentation module based on DeepLabV3+ and introduces an additional semantic thread to obtain scene semantic information. In the tracking process, semantic segmentation results are combined with a dual depth-consistency checking strategy to identify and remove dynamic feature points while preserving reliable static features. Furthermore, the original keyframe selection mechanism is enhanced by incorporating relative pose variations between consecutive frames as an additional criterion, which helps reduce redundant keyframes and improve localization accuracy. A dense mapping thread is also introduced to fuse multi-frame point cloud data and generate a dense three-dimensional map of indoor environments. Experiments conducted on the TUM RGB-D Dataset demonstrate that the proposed method significantly improves trajectory accuracy. The average localization error is reduced by more than 95% in highly dynamic sequences and by 39.60% in low-dynamic sequences. The results show that the proposed approach achieves more accurate pose estimation and dense map reconstruction, providing effective support for mobile robot localization and navigation in indoor dynamic environments.

Keywords: Dense mapping; Semantic segmentation; Depth consistency detection; Keyframe selection; Point cloud fusion.

1. Introduction

Currently, simultaneous localization and mapping (SLAM) technologies are primarily divided into two categories: laser-based SLAM, which relies on LiDAR sensors, and vision-based SLAM, which utilizes cameras. Researchers both domestically and internationally have conducted extensive studies on SLAM technology to address the challenges of autonomous localization and navigation for robots and related equipment. Laser SLAM has achieved relatively mature development in both theoretical research and practical applications, offering strong environmental adaptability and system stability. Therefore, it is widely used in fields such as autonomous driving and robotic navigation. However, this technology still has certain limitations. For example, LiDAR devices are relatively expensive, and their field of view constraints limit the scanning range, thereby affecting the capability to acquire environmental information. Moreover, LiDAR primarily captures spatial geometric information and lacks the ability to provide visual features such as color and texture, which makes it insufficient for application scenarios requiring richer environmental information. In contrast, vision-based SLAM demonstrates stronger adaptability, as it can be applied to various visual sensors, including monocular, stereo, and RGB-D cameras, with overall lower hardware costs. By capturing images, vision-based SLAM can acquire diverse information such as color, texture, and shape, thereby providing strong environmental perception capabilities. In complex indoor and outdoor environments, this technology typically enables stable autonomous localization and map construction, making it highly valuable for applications in robotic navigation and intelligent perception.

After years of development, the field of visual SLAM has produced a series of representative classic algorithms, such as MonoSLAM [1], Parallel Tracking and Mapping (PTAM) [2], ORB-SLAM3 [3], and VINS-Fusion [4]. These systems generally achieve high-precision camera localization and map construction in static environments. However, their performance is often significantly affected when the scene contains a large number of dynamic objects. Since traditional visual SLAM algorithms are mostly based on the assumption of a static world, they are prone to introducing erroneous feature points under complex dynamic interference, leading to accumulated localization errors or even tracking failure. This issue, to some extent, limits the application of visual SLAM technology in real-world complex environments.

In recent years, with the rapid advancement of deep learning technologies, semantic segmentation and object detection methods have achieved remarkable breakthroughs in the field of computer vision and have been gradually integrated into visual SLAM systems for identifying and processing dynamic objects. For example, DynaSLAM [5] builds upon ORB-SLAM2 by combining multi-view geometric constraints with a semantic segmentation network to detect and eliminate dynamic objects, while also employing a background inpainting method to restore static regions occluded by dynamic objects. DS-SLAM [6] introduces an independent semantic processing thread and incorporates optical flow methods to detect moving objects, subsequently constructing a semantic octree map. On the other hand, RDS-SLAM [7] adopts a keyframe-based semantic segmentation strategy to obtain the latest semantic information, enabling the tracking thread to proceed without waiting for semantic segmentation results, thereby improving the systems real -time performance to some extent, though its semantic segmentation module still incurs high computational costs. Additionally, Fu et al. [8] enhanced localization accuracy in dynamic scenes by integrating the

YOLO object detection network with a depth consistency detection mechanism, further generating a dense point cloud map suitable for robotic navigation. However, as object detection methods typically provide only bounding box-level detection results, lacking pixel-level accuracy, they may inadvertently remove some static features during the dynamic feature elimination process, resulting in the loss of valid information.

To address the issues of false detection and missed detection in identifying potential dynamic features in existing dynamic SLAM methods, which consequently affect the systems localization accuracy, this paper proposes a DeS-SLAM method that integrates semantic segmentation with depth consistency detection. The method constructs a semantic segmentation module based on DeepLabV3+ and employs MobileNetV2 as its backbone network. First, MobileNetV2 significantly reduces the computational complexity of the model through its depthwise separable convolutional structure, making the system more suitable for deployment on platforms with limited computational resources. Second, its linear bottleneck structure effectively enhances feature information transmission and improves the networks expressive efficienc y. Additionally, by incorporating the atrous convolution module from DeepLabV3+, the model can capture multi-scale contextual information while maintaining high segmentation accuracy, thereby providing more reliable semantic support for dynamic object recognition and feature filtering in visual SLAM systems. Based on the above methods, the proposed system can more effectively identify and eliminate potential dynamic features in dynamic environments, thereby improving localization stability and system robustness. The main contributions of this paper are as follows:

(1) To address the degradation in localization accuracy and the accumulation of pose estimation errors caused by interference from moving objects in dynamic scenes, this paper introduces semantic segmentation information for prior identification of potential dynamic regions in the scene. On this basis, depth consistency constraints are combined to further evaluate feature points, thereby effectively filtering out dynamic features, reducing interference from moving objects in the tracking and localization processes of the visual SLAM system, and enhancing the systems stability and localization accuracy in dynamic environments. This method is based on the semantic information provided by DeepLabV3+ and utilizes a depth consistency detection mechanism to further filter dynamic features.

(2) By improving the ORB-SLAM2 system, this paper adjusts the feature point extraction threshold, redesigns the keyframe determination strategy, and introduces a dense mapping thread, thereby enhancing the capabilities of localization and dense map construction. This method aims to improve the environmental perception capability of SLAM systems in complex and dynamic indoor environments, providing more reliable technical support for the autonomous navigation and task execution of mobile robots.

2. Related works

Current research on dynamic object detection and removal can generally be categorized into three technical approaches: geometry-based methods, semantics-based methods, and methods that integrate geometry and semantics. Geometry-based methods primarily rely on spatial structural information in the scene, identifying dynamic objects by analyzing feature point motion consistency, depth variation, or geometric constraints. Semantics-based methods utilize deep learning models to recognize semantic categories in images, determining potential dynamic targets based on prior knowledge of object categories, thereby achieving identification and separation of dynamic regions. In contrast, methods that integrate geometry and semantics leverage the strengths of both types of information, improving the accuracy of dynamic object detection and enhancing system stability in complex environments through collaborative analysis and fusion of multi-source data.

2.1. Geometry-based methods

Geometry-based dynamic object detection methods primarily rely on geometric constraints between images to identify moving targets in a scene. Such methods typically analyze the motion consistency of feature points across consecutive frames to determine their attributes: if a feature point satisfies predefined geometric model constraints, it is considered to originate from the static background; otherwise, it may belong to a dynamic object.

Kim et al. proposed a non-parametric model based on background subtraction, which extracts and filters moving objects by utilizing depth information in the scene [9]. However, this method is sensitive to the quality of depth data and is prone to misjudgment when depth information is uncertain. Subsequently, Wang et al. [10] introduced a dynamic object detection method combining epipolar constraints and depth clustering, which first removes outlier matching points through epipolar geometric constraints and then clusters depth maps to segment moving objects. However, as it assumes a small disparity change between adjacent frames, it tends to produce relatively large localization errors in highly dynamic scenes.

In addition, Dai et al. [11] employed Delaunay triangulation to establish spatial relationships among feature points, achieving the separation of dynamic and static points by analyzing the correlation between them. Nevertheless, the detection performance of this method significantly declines when moving objects occupy a large proportion of the scene. In 2022, Song et al. [12] proposed the DynaVINS method, which incorporates an improved Bundle Adjustment (BA) and leverages attitude prior information provided by an Inertial Measurement Unit (IMU) to reduce the impact of dynamic features on the optimization process. Meanwhile, keyframe grouping and multi-hypothesis constraints are used to mitigate interference from dynamic objects in loop closure detection. In general, geometry-constraint-based methods do not rely on pre-trained models and are computationally efficient, yet their performance depends heavily on the quality of feature points and often struggles to maintain stability in highly dynamic environments.

On the other hand, some studies utilize optical flow information to detect dynamic regions. Meng et al. [13] proposed a method that identifies dynamic regions by analyzing pixel-level motion changes in consecutive RGB images and subsequently removes dynamic feature points accordingly. This method demonstrates certain effectiveness in dynamic object detection but is sensitive to illumination variations. Moreover, Fang et al. [14] applied optical flow to detect and filter dynamic regions in images, thereby reducing errors in inter-frame pose estimation. However, due to inherent errors in optical flow estimation, its overall contribution to system performance improvement remains

limited. The FlowFusion method [15] highlights dynamic semantic features in RGB-D point clouds by leveraging optical flow residuals, thereby providing a more accurate distinction between dynamic and static regions for system tracking and mapping, though at a relatively high computational cost. Finally, M. C. Bakkay et al. [16] proposed a dynamic object detection method based on scene flow, which employs a region-growing segmentation algorithm to separate dynamic and static regions, effectively reducing the impact of feature matching errors.

Geometry-based methods offer certain advantages for dynamic object detection, yet their performance often relies on rich texture information in the scene. When environmental textures are sparse or uniformly distributed, the stability of feature extraction and matching is compromised, which in turn reduces the accuracy of dynamic object identification. Therefore, researchers typically mitigate the interference of dynamic regions on SLAM system tracking and mapping by localizing and eliminating them. With the advancement of deep learning technologies, object detection and semantic segmentation methods have achieved significant improvements in both accuracy and application scope. Integrating deep learning methods with SLAM systems and leveraging semantic prior information to identify and remove features in dynamic regions has gradually become an important research direction for enhancing system robustness and localization accuracy.

2.2. Semantics-based methods

Semantics-based dynamic object detection methods primarily rely on deep learning models to semantically interpret image content, determining the motion attributes of objects by obtaining their category information or semantic labels in the scene. In such methods, the system first recognizes semantic objects in the image, then assesses whether the object category is potentially dynamic, and removes feature points within dynamic regions to reduce the interference of dynamic objects on pose estimation, thereby enhancing the localization accuracy of the SLAM system.

Gao R et al. [17] proposed a real-time dynamic visual SLAM method based on the YOLOv5 object detection network, aiming to improve system robustness and camera localization accuracy in indoor dynamic scenes affected by moving objects. Chang Z et al. [18] introduced an improved lightweight object detection network, YOLOv4-tiny, to detect dynamic regions during the tracking stage and subsequently eliminate dynamic feature points within these regions. Experimental results show that this method can improve system localization accuracy to some extent. However, since object detection typically outputs results in the form of bounding boxes, some static feature points inside the bounding box may be mistakenly removed. Similarly, Xiao et al. proposed a semantic-enhanced simultaneous localization and mapping framework, Dynamic SLAM [19]. This method incorporates a prior knowledge-based SSD object detector to identify potential dynamic targets at the semantic level in the newly added detection thread. Nevertheless, this approach also adopts the strategy of directly removing all feature points inside the detection bounding boxes, which may still lead to the erroneous removal of some valid static features. Liu and Miura employed semantic segmentation techniques to identify dynamic objects in images and filter out abnormal feature points in keyframes [20]. While this method partially mitigates the misjudgment issues associated with bounding box-based detection through pixel-level semantic information, it may still result in the loss of useful information due to excessive removal of feature points in practical applications.

Semantics-based methods endow SLAM systems with higher-level scene understanding capabilities through deep learning models, enabling more accurate identification of dynamic objects and thereby enhancing system robustness in dynamic environments to a certain extent. However, such methods often heavily depend on the performance of neural network models and struggle to balance real-time requirements with detection accuracy. Moreover, when relying solely on semantic information to process dynamic regions, some static feature points within detection bounding boxes or segmented areas may be misclassified as dynamic and completely removed. This reduces the number of static features available to the system and may, in some cases, lead to insufficient feature points, thereby affecting the stability and localization accuracy of the SLAM system.

2.3. Methods combining geometric and semantic information

In recent years, to enhance the stability and localization accuracy of visual SLAM in dynamic environments, researchers have increasingly explored the integration of geometric constraints with semantic information, proposing various SLAM methods tailored for dynamic scenes. Such approaches generally adopt a "deep learning semantic analysis + geometric verification" framework: first, deep learning models are used to obtain semantic categories or dynamic prior information of objects in images; then, multi-view geometric constraints are applied for further confirmation, thereby identifying and removing dynamic feature points. By fusing semantic and geometric information, dynamic targets can be detected more accurately, reducing the interference of dynamic objects with system localization and mapping.

Building on the ORB-SLAM2 framework, Yu et al. [6] proposed the DS-SLAM system, which introduces the SegNet semantic segmentation network and combines semantic segmentation results with motion consistency detection to identify and filter dynamic regions. Additionally, the system establishes an independent semantic mapping thread for constructing a 3D semantic octree map. Bescos et al. [5] proposed DynaSLAM, which integrates the Mask R-CCNN instance segmentation network with multi-view geometry methods. It first identifies potential dynamic objects (e.g., movable chairs, pedestrians) and then uses geometric relationships to filter out dynamic feature points, while incorporating a background inpainting module to restore occluded static regions. DOT-SLAM [21] utilizes keyframe semantic segmentation and dynamic deviation metrics, combining static and dynamic feature points for pose estimation, thereby enhancing the system's robustness in dynamic environments. SOF-SLAM [22] employs SegNet to obtain motion prior information, which is used as a mask to exclude dynamic or potentially dynamic feature correspondences when computing optical flow between consecutive frames, retaining only static features for tracking and optimization. TwistSLAM proposed by Gonzalez et al. [23] incorporates object motion parameters into the backend optimization process through semantic point cloud clustering and mechanical joint constraints. DyOb SLAM [24] employs neural networks and dense optical flow methods to separately model static and dynamic objects, and estimates the motion

speed of dynamic objects in real-time, achieving stable inter-frame tracking.

Furthermore, Chang et al. [25] used the YOLACT instance segmentation network to detect dynamic targets in images and further filtered dynamic feature points with geometric constraints. He et al. [26] proposed OVD-SLAM, an online visual SLAM system for dynamic environments, which distinguishes foreground and background feature points through object detection and depth information, and uses optical flow to quickly identify dynamic features. In the same year, You et al. [27] employed the YOLACT++ instance segmentation method and proposed a robust tracking strategy based on semantic information, detecting and removing dynamic features to suppress the impact of dynamic objects on the system. Han et al. proposed PSPNet-SLAM [28], which introduces a PSPNet semantic segmentation thread into ORB-SLAM2 and combines semantic segmentation results with optical flow to remove dynamic feature points, ultimately constructing a more environmentally aware SLAM map.

In summary, the interference of moving objects in dynamic environments has become a critical factor limiting the performance improvement of visual SLAM systems. The presence of dynamic targets leads to feature matching errors and pose estimation deviations, thereby affecting the localization accuracy and map quality of the system. Therefore, how to effectively identify and handle dynamic objects remains a significant challenge in visual SLAM research. In recent years, object detection algorithms have made remarkable progress in semantic understanding and real-time performance, enabling relatively accurate identification of target information in complex scenes. Integrating object detection technology with visual SLAM systems can not only reduce mismatches and tracking errors caused by dynamic objects but also enhance the localization accuracy and robustness of the system in dynamic environments, while improving map construction outcomes. Thus, accurate identification and effective removal of dynamic targets are of great importance for enhancing the overall performance of visual SLAM systems in complex environments.

3. System Overview

3.1. System Architecture

This paper proposes improvements based on the ORB-SLAM2 framework. The overall system architecture, as illustrated in Figure 1, consists of four threads: tracking, local mapping, dense mapping, and loop closing. The tracking thread serves as the core module, responsible for receiving camera images and estimating camera poses in real-time. For each input RGB frame, the system simultaneously performs ORB feature extraction and matching, and incorporates a DeepLabv3+ semantic segmentation network to obtain semantic masks, which identify dynamic object regions in the image.

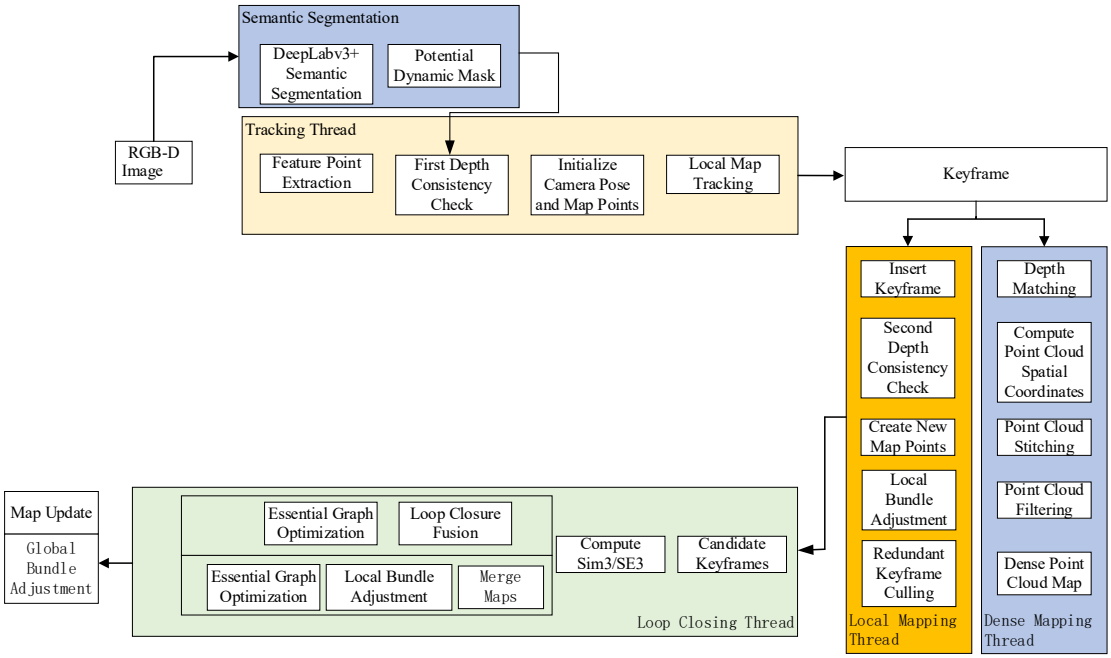


Figure 1. System Framework Diagram

On this basis, depth consistency detection is applied to potential dynamic regions to initially remove dynamic feature points. The local mapping thread handles keyframe insertion and local map optimization. Upon receiving a keyframe, this thread integrates it into the map and optimizes the keyframe poses and map point coordinates through local Bundle Adjustment (BA). Simultaneously, it performs triangulation of new map points, culls redundant keyframes, and filters new and old map points to enhance the accuracy and stability of the local map. Additionally, based on the long-term stability of keyframes, a second depth consistency detection is conducted to further eliminate residual dynamic map points. The dense mapping thread, after acquiring a new keyframe, combines the corresponding depth map and semantic segmentation mask to generate dense point clouds through projection and fuses them into the global point cloud map. This thread also employs point cloud filtering methods to remove outliers, ensuring the accuracy and smoothness of the global dense map. The loop closing thread is designed to identify loop closure relationships between historical keyframes and the current keyframe. Once a similar keyframe is detected, the system aligns the poses by solving a Sim(3) similarity transformation, then constructs loop closure constraints and performs global optimization to effectively

eliminate accumulated drift and enhance the global consistency of the system.

3.2. Dynamic Feature Point Removal Method

In dynamic scenes, moving objects can significantly interfere with camera tracking and map updating in visual SLAM systems. To address this problem, this paper proposes a novel dynamic feature point filtering algorithm that combines semantic information and depth consistency constraints. The algorithm is implemented in the tracking thread: it first extracts potential dynamic regions in the image using a semantic segmentation network, and then analyzes the depth fluctuation of feature points within these regions. Feature points whose depth deviation exceeds the predefined threshold are marked as dynamic points and removed from map construction and pose optimization, thereby reducing the negative impact of dynamic targets on system localization accuracy.

Considering the measurement noise inherent in RGB-D cameras and the cumulative error caused by inter-frame pose estimation, directly using a fixed threshold may lead to incorrect feature classification. Therefore, this paper comprehensively considers dynamic point removal and static feature preservation. Through extensive parameter tuning and experimental verification, the depth deviation threshold is finally set to 12%. This setting can accurately eliminate dynamic feature points while retaining as many valid static key points as possible, thus maintaining the integrity and consistency of dense mapping.

To verify the rationality and generalization ability of this threshold, multiple indoor dynamic test scenes are constructed with different proportions of moving objects and various camera motion speeds for controlled experiments. The results demonstrate that an excessively small threshold (e.g., 6%) causes extensive misclassification of static features, which reduces point cloud density and degrades tracking stability. In contrast, an excessively large threshold (e.g., 18%) fails to fully remove dynamic points, resulting in trajectory drift. Based on comprehensive experimental results, the 12% depth deviation threshold achieves the best balance between dynamic interference suppression and static feature preservation, effectively improving system mapping quality and localization accuracy.

Based on the above method, the preliminary screening rules for dynamic feature points can be expressed by Formula 1 and Formula 2.

Depth Consistency Deviation:

$$\varepsilon_d = \left| 1 - \frac{d_{k-1}^{proj}}{z_k} \right| \times 100\% \tag{1}$$

Dynamic Point Removal Criterion：

$$p_k \in \mathsf{S}_{dn} \left(M_k(u_k, v_k) = 1 \right) \wedge \left(\varepsilon_d > \tau_d \right), \tau_d = 12\% \tag{2}$$

where z_k is the observed depth of the feature point in the current frame k, d_{k-1}^{proj} denotes the projected depth of the corresponding 3D point from the previous frame $k-1$, S_{dn} is the set of dynamic feature points to be eliminated, $M_k(u_k, v_k)$ is the semantic segmentation mask indicating whether the pixel (u_k, v_k) belongs to a potential dynamic region, and τ_d is the predefined depth deviation threshold. The metric ε_d quantifies the relative deviation between the predicted and observed depth, and a feature point is classified as dynamic and excluded from tracking and mapping only when both the semantic mask condition and the depth deviation threshold condition are satisfied simultaneously, which serves as the core criterion for dynamic feature identification.

After the initial screening of dynamic feature points, this paper further introduces a secondary verification mechanism based on keyframe depth consistency to remove the remaining dynamic points that were not filtered out in the first stage. According to the relative pose relationship between the current frame and its co-visible keyframes, the reserved feature points are projected into the keyframe coordinate system, and the corresponding reprojection operation is represented by Equations (3) and (4). Then, the depth obtained by projection is compared with the optimized depth of the corresponding map point in the keyframe, and the relative depth difference is computed. As the map points in keyframes have been fully optimized via multi-view bundle adjustment, their depth values are stable and credible, which can be used as an effective reference to judge whether a feature point is dynamic. When the depth difference of a feature point is larger than the given threshold (e.g., 8%), this point is recognized as a residual dynamic feature and removed from the processes of map update and pose optimization.

This two-stage verification mechanism significantly improves the robustness of the system in complex dynamic environments and ensures the accuracy of trajectory estimation and pose computation. The corresponding mathematical formulation is given in Equations (5) and (6).

Coordinate transformation equation:

$$X_r = T_{r \leftarrow k} X_k \tag{3}$$

Pixel reprojection expression:

$$u_r^{proj} = \mathsf{P}(X_r, K) \tag{4}$$

Depth consistency deviation calculation:

$$\delta_{rel} = \frac{|z_r - d_r|}{z_r} \times 100\% \tag{5}$$

Secondary filtering criterion:

$$p_k \in \mathsf{D}_{res} \text{ if } \delta_{rel} > \tau_d, \tau_d = 8\% \tag{6}$$

In the above formulas, X_k and X_r denote the 3D coordinates of the feature point in the current frame and reference keyframe, respectively. $T_{r \leftarrow k}$ represents the relative pose transformation matrix between frames. $\mathsf{P}(\cdot)$ is the camera projection function, and K is the intrinsic parameter matrix. z_r is the optimized depth of map points, d_r is the observed depth, δ_{rel} is the relative depth residual, and τ_d is the secondary filtering threshold.

3.3. Dense Map Construction

To address the issues in ORB-SLAM2 caused by moving objects in dynamic environments—such as point cloud noise, artifacts, loss of environmental details, and map contamination due to sparse features—this study introduces a dense mapping thread into its architecture. This thread takes the existing keyframes and their corresponding accurate poses as input, aiming to perform high-precision and high-

completeness 3D dense reconstruction of the surrounding indoor dynamic scenes. Considering that dynamic objects (e.g., pedestrians or moving devices) may change their poses during continuous observation, directly incorporating them into mapping would inevitably introduce inconsistent point cloud structures, severely affecting map stability and the reliability of subsequent applications. Therefore, this module is specifically designed with robust handling mechanisms for dynamic objects.

The dense mapping process primarily relies on the color images and depth data provided by the RGB-D camera. The basic workflow is as follows: first, the image pixel coordinates are back-projected into 3D space according to the camera model and the depth values to obtain the initial point cloud; then, point cloud recovery and filtering algorithms are applied to remove outliers and noise caused by dynamic objects, generating an accurate and clean local dense point cloud. The specific formula for converting pixel coordinates to 3D points is as follows:

$$x = (u - q_x).\frac{z}{p_x} \tag{7}$$

$$y = (v - q_y).\frac{z}{p_y} \tag{8}$$

$$z = \frac{d}{b} \tag{9}$$

Here, p_x and p_y represent the effective focal lengths of the image sensor in the horizontal and vertical directions, respectively (in pixels), while q_x and q_y denote the coordinates of the principal point in the pixel coordinate system. d is the raw observation obtained from the depth sensor, and b is the scaling factor used to convert the raw measurement into an actual physical distance. z represents the resulting actual depth of the corresponding point in 3D space.

Subsequently, the system maintains a pose queue to record the camera poses corresponding to the keyframes. Using a point cloud registration algorithm, the local point clouds generated at different times and from different viewpoints are accurately aligned and fused according to their poses, gradually constructing a complete and consistent global dense point cloud map. The specific computation formula for point cloud fusion is as follows:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R_{wc} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + t_{wc} \tag{10}$$

Here, R_{wc} is a 3×3 rotation matrix, and t_{wc} is a 3×1 translation vector. Together, they describe the spatial transformation between the camera coordinate system and the world coordinate system.

4. Experimental Validation

To ensure the reliability, reproducibility, and generalization of the experimental results, all experiments in this study were conducted in a controlled environment. The hardware platform consists of an NVIDIA GeForce RTX 3090 GPU, an Intel Core i7-12700 CPU, and 32 GB of RAM. The software environment runs on Ubuntu 20.04 LTS, with the deep learning framework implemented in Python 3.9 and accelerated using PyTorch 2.0 and CUDA 12.1. All datasets used in the experiments are publicly available, ensuring the objectivity and comparability of the results.

4.1. Semantic–Depth Consistency Fusion Method and Its Performance

To validate the effectiveness of the proposed dynamic feature point processing method, which integrates semantic segmentation with depth consistency detection in dynamic visual SLAM, we conducted comparative experiments and visualized the results of feature point filtering. Figure 2 shows the comparison of dynamic point processing, where (a) illustrates the feature point distribution extracted by the ORB-SLAM2 system, and (b) shows the feature point distribution after applying the proposed method. The comparison indicates that, after applying our approach, feature points on moving objects are effectively suppressed while static scene features are well preserved.

To avoid the potential mis-removal of static features caused by relying solely on semantic segmentation, the method further refines the semantic segmentation results using depth consistency detection, achieving precise identification of dynamic features. As shown in (c), in scenes containing objects with different motion states, the proposed method accurately identifies and removes only the feature points corresponding to actual moving entities (e.g., the person walking on the left), while retaining the feature points of static objects (e.g., the person standing on the right) intact.

These results demonstrate that the proposed fusion strategy enables selective filtering of feature points in complex dynamic scenes, effectively suppressing dynamic interference while preserving the structural integrity of static environments, thereby providing a reliable feature basis for subsequent robust pose estimation.

(a) (b) (c)

Figure 2. Dynamic Point Removal Results

4.2. SLAM Algorithm Error Evaluation

To evaluate the performance of the visual SLAM system in dynamic scenes, this paper conducts experiments based on the widely used TUM RGB-D dataset. This dataset contains various camera motion patterns and dynamic interference scenarios, which provides a standard benchmark for the quantitative evaluation of SLAM algorithms. In the experiments, we select five sequences with different dynamic characteristics: the former four sequences represent highly

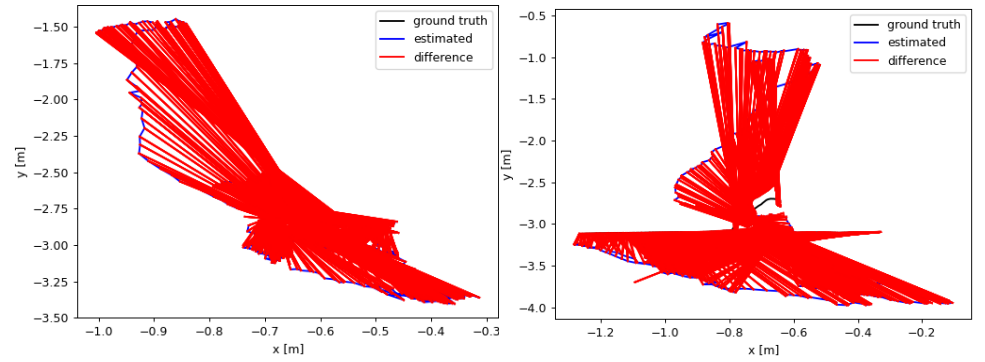
dynamic environments with obvious pedestrian motion, while the last one corresponds to a weakly dynamic and relatively static scene, so as to comprehensively verify the adaptability of the system under different levels of dynamic interference.

For trajectory accuracy evaluation, Absolute Trajectory Error (ATE) and Relative Pose Error (RPE) are adopted as the main metrics. ATE reflects the global deviation between the estimated trajectory and the ground truth, while RPE focuses on the local drift generated in consecutive frame pose estimation. Based on the classic ORB-SLAM2 system as the baseline, the quantitative comparison results are listed in Table 1. It can be observed that the proposed DeS-SLAM system achieves remarkable performance improvements in highly dynamic sequences, where the RMSE of trajectory error is reduced by more than 92%. Even in the low-dynamic sequence fr3_sitting_static with weak environmental changes, the proposed method still obtains an accuracy improvement of 39.60% in terms of RMSE.

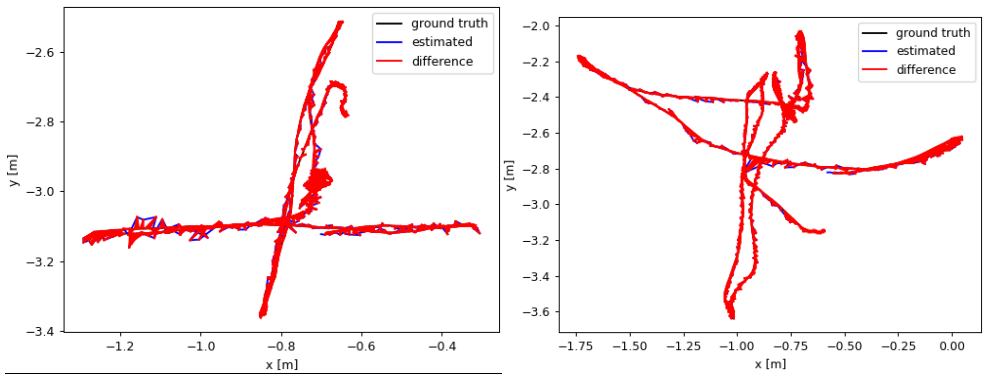
Figure 3 further presents the visualization results of trajectory comparison. It can be clearly seen that, in highly dynamic environments, the trajectory estimated by ORB-SLAM2 (Figure 3(a)) deviates significantly from the ground truth. In contrast, the trajectory obtained by DeS-SLAM (Figure 3(b)) is highly consistent with the ground truth. The results demonstrate that the dynamic feature processing strategy combining semantic segmentation and depth consistency can effectively eliminate the negative impacts of dynamic objects on pose estimation, and thus enables more stable and accurate trajectory reconstruction in complex dynamic environments.

Table 1. Comparison of absolute trajectory error between DeS-SLAM and ORB-SLAM2

	ORB-SLAM2				DeS-SLAM				Improvement (%)			
Sequence	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD
fr3 walking xyz	0.7360	0.6914	0.5996	0.807	0.0197	0.0177	0.0181	0.0095	97.32	97.44	96.98	98.82
fr3 walking halfsphere	0.6505	0.5925	0.6288	0.2134	0.0461	0.0355	0.0276	0.0251	92.91	94.01	95.61	88.24
fr3 walking rpy	0.6985	0.4587	0.6017	0.3201	0.0285	0.0367	0.0315	0.0297	95.92	92.00	94.76	90.72
fr3 walking static	0.5118	0.4157	0.3976	0.2153	0.0071	0.0062	0.0047	0.0031	98.61	98.51	98.82	98.56
fr3 sitting static	0.0101	0.0067	0.0086	0.0051	0.0061	0.0042	0.0053	0.0037	39.60	37.31	38.37	27.45



(a) ORB-SLAM2



(b) DeS-SLAM

Figure 3. Comparison of System Pose Estimation Experimental Results

In the quantitative analysis of relative pose error, Table 2 and Table 3 present the error statistics for the rotational and translational components, respectively, including root mean square error (RMSE), mean, median, and standard deviation

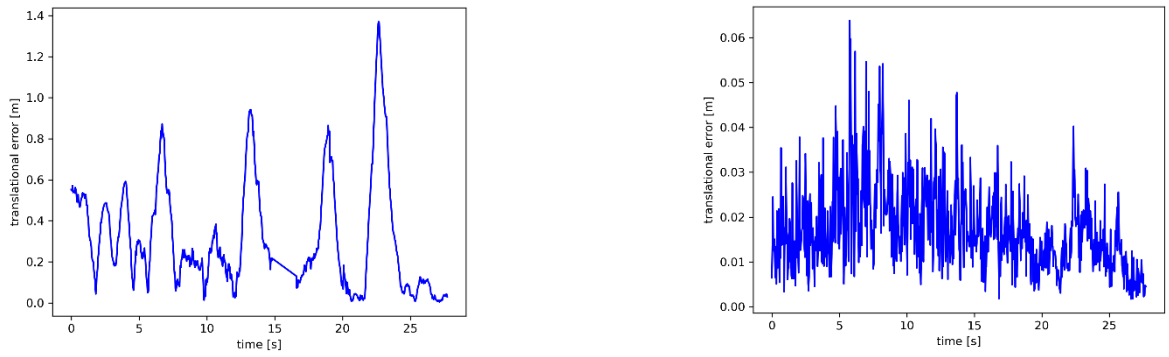
(SD), as well as the performance improvement ratios of the proposed method compared with the baseline system. Experimental data show that in the highly dynamic scene fr3 walking xyz, the translational RPE (RMSE) is reduced by 94.20%, and the rotational RPE (RMSE) is reduced by 90.36%. In the sequences fr3 walking rpy and fr3 walking halfsphere, the key error metrics of translational RPE are improved by more than 92%, where the improvement ratio of translational RPE (RMSE) reaches 94.71% for fr3 walking halfsphere and 92.98% for fr3 walking rpy. Even in the sequence fr3 sitting static with relatively low dynamic interference, the translational RPE (RMSE) is still improved by 27.52% and the rotational RPE (RMSE) by 11.67%. This demonstrates that the proposed dynamic feature processing method exhibits consistent adaptability in scenes with different levels of dynamic disturbance.

Table 2. Comparison of Relative Pose Error (Rotational) between ORB-SLAM2 and DeS-SLAM

	ORB-SLAM2				DeS-SLAM				Improvement (%)			
Sequence	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD
fr3 walking xyz	8.2367	5.1479	3.1683	5.9756	0.7942	0.5531	0.4627	0.4103	90.36	89.26	85.40	93.13
fr3 walking halfsphere	4.1358	2.3410	0.4253	3.9879	0.3987	0.3481	0.3576	0.1988	90.36	85.13	36.47	95.01
fr3 walking rpy	9.4582	6.7014	3.7598	6.7390	0.8227	0.7283	0.6535	0.4048	91.31	89.13	82.51	93.99
fr3 walking static	8.2767	5.9294	2.9405	5.8175	1.5554	1.0603	0.7144	1.1381	81.21	82.13	75.68	80.44
fr3 sitting static	0.3199	0.2885	0.2751	0.1381	0.2826	0.2528	0.2362	0.1264	11.67	12.37	14.13	8.51

Table 3. Comparison of Relative Pose Error (Translational) between ORB-SLAM2 and DeS-SLAM

	ORB-SLAM2				DeS-SLAM				Improvement (%)			
Sequence	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD	RMSE	Mean	Median	SD
fr3 walking xyz	0.3812	0.2564	0.1285	0.2841	0.0221	0.0189	0.0167	0.0110	94.20	92.63	87.01	96.13
fr3 walking halfsphere	0.2305	0.1048	0.0185	0.2063	0.0122	0.0105	0.0093	0.0052	94.71	94.71	49.73	97.48
fr3 walking rpy	0.4527	0.3176	0.1763	0.3229	0.0318	0.0281	0.0256	0.0144	92.98	91.15	85.48	95.54
fr3 walking static	0.4068	0.2917	0.1524	0.2846	0.0731	0.0487	0.0315	0.0532	82.03	83.30	79.33	81.31
fr3 sitting static	0.0109	0.0097	0.0086	0.0051	0.0079	0.0068	0.0062	0.0039	27.52	29.90	27.91	23.53



(a) (b)

Figure 4. Comparison of Relative Pose Error (RPE)

To further investigate the trajectory stability of the proposed system in dynamic environments, Figure 4 depicts the translational relative pose error (RPE) curves of both algorithms on a typical high-dynamic sequence. As shown in

Figure 4(a), the translational RPE curve of the original ORB-SLAM2 system presents obvious violent fluctuations throughout the sequence, and frequent abnormal spikes appear in regions with severe dynamic interference, indicating that the system is seriously disturbed by moving objects. In contrast, the translational RPE curve obtained by DeS-SLAM (Figure 4(b)) remains stable and gentle during the whole sequence, and the error amplitude is always maintained at a low level without large fluctuations or abnormal deviations.This comparison fully demonstrates that the dynamic feature elimination strategy combining semantic information and depth consistency detection can effectively weaken the negative influence of dynamic objects on pose estimation. The above results are consistent with the previous quantitative analysis of RPE, which further proves that the proposed method has stronger robustness and higher positioning accuracy in complex dynamic scenes.

4.3. Dense Mapping Performance

Due to inherent limitations in the continuity and completeness of geometric representation in sparse feature-point maps, it is difficult to finely capture surface morphology and spatial structures of the environment, which imposes significant constraints on applications that rely on high-fidelity scene models, such as 3D reconstruction and augmented reality. Taking the experimental results on the TUM dataset as an example, Figure 5 shows a typical sparse feature-point map generated by ORB-SLAM2, where red points represent extracted features and the green line indicates the camera motion trajectory. It can be observed that the feature points are distributed in an isolated and discrete manner in space, lacking the spatial associations needed to form continuous surfaces or object contours. This leads to visually sparse reconstruction results that cannot provide sufficient geometric detail to support refined visual analysis or further scene understanding. Therefore, developing dense point cloud maps capable of representing the complete geometric form and rich detail of a scene is of clear practical necessity for improving the perception and reconstruction capabilities of systems in dynamic environments.

Through densification processing, discrete sparse feature points can be transformed into a dense point cloud map with high spatial density and strong structural continuity. Such a map not only contains richer geometric details of the scene but can also construct a more complete surface topology through the proximity relationships between point clouds, thereby significantly enhancing the three-dimensional depth and visual realism of scene representation and increasing its practical value in tasks such as 3D interaction and fine-grained reconstruction. Figure 6 shows a static dense point cloud map generated in a dynamic environment using the proposed method. While effectively eliminating interference from dynamic objects, the map well preserves the contour and structural features of various objects in the static scene (such as chairs, monitors, keyboards, etc.), and uses semantic-based coloring to distinguish different objects. The overall point cloud is structurally coherent, with clear details and no obvious motion artifacts, demonstrating the good reconstruction quality of the proposed method in dynamic environments.

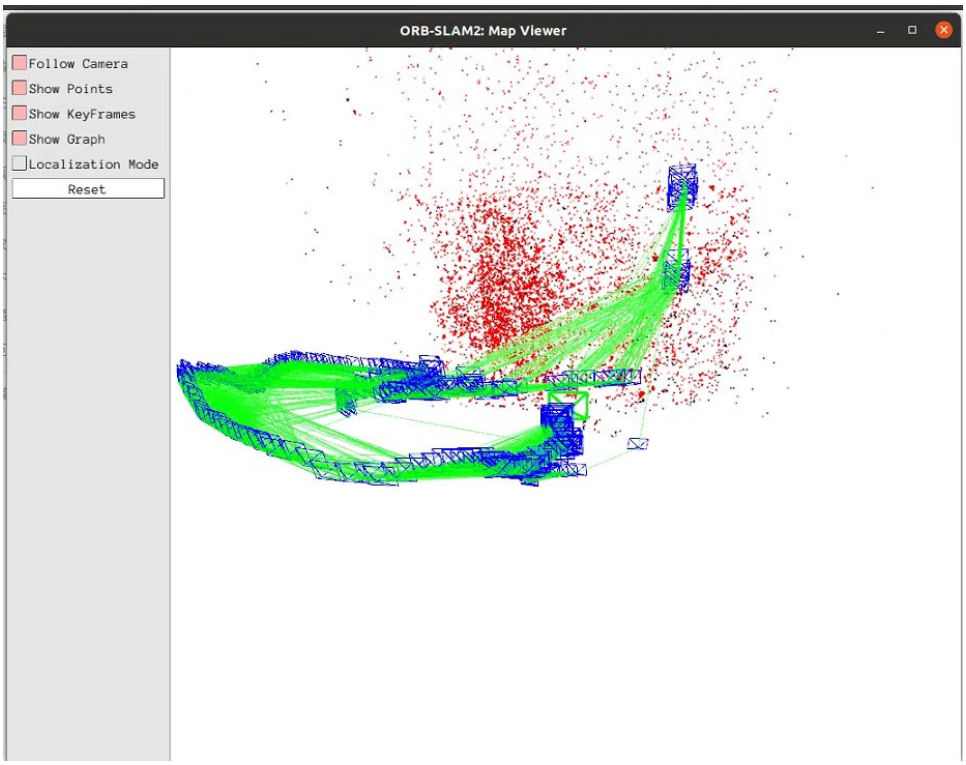


Figure 5. Sparse Point Cloud Map by ORB-SLAM2

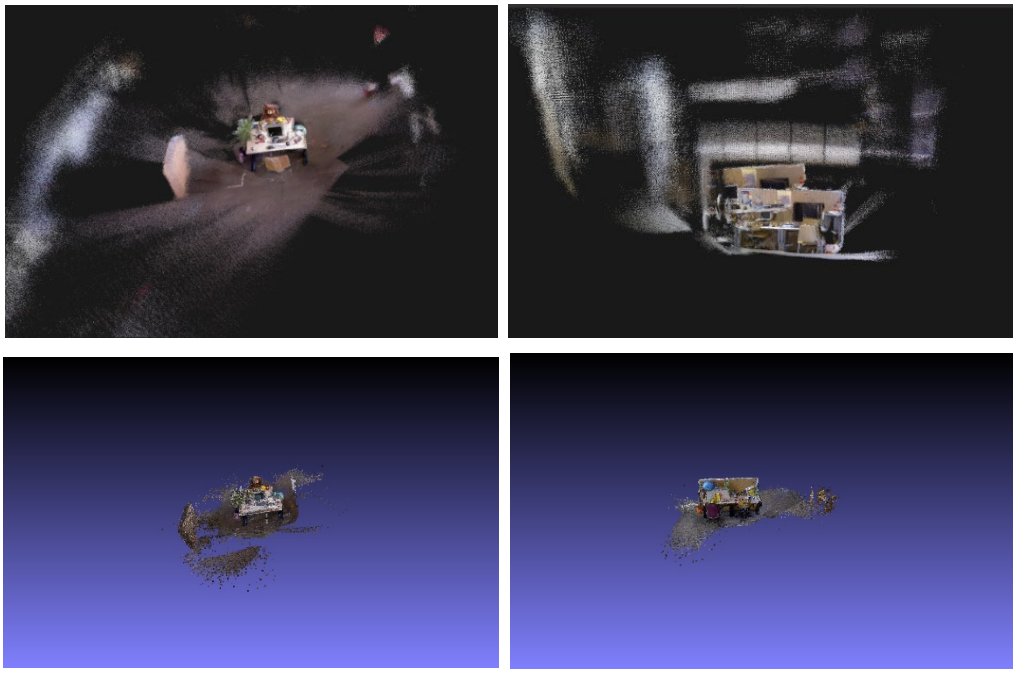


Figure 6. Dense Point Cloud Map

5. Conclusion

To address the degradation in localization accuracy and mapping errors in visual SLAM systems caused by interference from dynamic objects in dynamic environments, this paper proposes and constructs the DeS-SLAM system. The system achieves pixel-level recognition of dynamic objects through an improved DeepLabV3+ semantic segmentation network and innovatively designs a dynamic feature point removal mechanism that integrates semantic information with dual depth-consistency verification. On this basis, an adaptive keyframe selection strategy based on inter-frame motion is introduced to enhance system efficiency, and a dense mapping thread is extended to generate high-quality 3D point cloud maps of static scenes. Comprehensive experiments conducted on the TUM RGB-D dynamic dataset demonstrate that the proposed method can effectively filter out interference from dynamic feature points, significantly improving trajectory estimation accuracy in highly dynamic scenarios (ATE reduced by more than 95%), while maintaining superior robustness and generalization capability in low-dynamic and regularly moving scenes. Its overall performance outperforms the original ORB-SLAM2 and other mainstream dynamic SLAM methods of the same period. This research provides an effective solution for stable localization and fine-grained environment modeling of mobile robots in complex dynamic environments. However, the current study focuses mainly on specific dynamic categories such as pedestrians. Future work will further expand the semantic segmentation model to recognize a wider range of dynamic objects and explore lightweight deployment and deeper multi-sensor fusion to enhance the system's practicality and generalizability in complex real-world scenarios.

References

[1] Davison AJ, Reid ID, Molton ND et al (2007) MonoSLAM: real-time single camera SLAM. IEEE Trans Pattern Anal Mach Intell 29(6):1052–1067

[2] Klein G, Murray D (2007) Parallel tracking and mapping for small AR workspaces. In: 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, pp 225–234.

[3] Campos C, Elvira R, Rodríguez JJG, Montiel, JMM., Tardós, JD (2021) Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. IEEE Trans Robot 37(6):1874–1890.

[4] Qin T, Li P, Shen S (2018) VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator vol 34, pp 1004–1020.

[5] BESCOS B, FÁCIL J M, Civera J, et al. DynaSLAM: tracking, mapping, and inpainting in dynamic scenes[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 4076-4083.

[6] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]//2018 IEEE/RSJ international conference on intelligent robots and systems (IROS). IEEE, 2018: 1168-1174.

[7] LIU Y B, MIURA J. RDS-SLAM: real-Time dynamic SLAM using semantic segmentation methods[J].IEEE Access, 2021, PP(99):1-1.

[8] Fu, Q., Zhong, Z., Ji, Y., Yan, S. (2026). Dynamic Visual SLAM Algorithm Combined with YOLO. In: Hassan, M.H.A., Jamaludin, A.S., Bin Zakaria, M.A., Usman, F., Uchidate, M. (eds) Intelligent Manufacturing and Mechatronics. SympoSIMM 2024. Lecture Notes in Mechanical Engineering. Springer, Singapore. https://doi.org/10.1007/978-981-96-7703-0_8

[9] Kim D H, Kim J H. Effective background model-based RGB-D dense visual odometry in a dynamic environment[J]. IEEE Transactions on Robotics, 2016, 32(6): 1565-1573.

[10] Wang R, Wan W, Wang Y, et al. A new RGB-D SLAM method with moving object detection for dynamic indoor scenes[J]. Remote Sensing, 2019, 11(10): 1143.

[11] Dai W, Zhang Y, Li P, et al. Rgb-d slam in dynamic environments using point correlations[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 44(1): 373-389.

[12] Song S, Lim H, Lee A J, et al. DynaVINS: A visual-inertial SLAM for dynamic environments [J]. IEEE Robotics and Automation Letters, 2022, 7(4): 11523-11530.

[13] Cheng J, Wang C, Meng M Q H. Robust visual localization in dynamic environments based on sparse motion removal[J]. IEEE Transactions on Automation Science and Engineering, 2019, 17(2): 658-669.

[14] Fang Y, Dai B. An improved moving target detecting and tracking based on optical flow technique and kalman filter[C]//2009 4th International Conference on Computer Science & Education. IEEE, 2009: 1197-1202.

[15] Zhang T, Zhang H, Li Y, et al. Flowfusion: Dynamic dense rgb-d slam based on optical flow[C]//2020 IEEE International Conference on Robotics and Automation (ICRA).IEEE, 2020: 7322-7328.

[16] Bakkay M C, Arafa M, Zagrouba E. Dense 3D SLAM in dynamic scenes using Kinect[J].Springer, Cham, 2015: 121-129.

[17] Gao R, Li Z, Li J, et al. Real-time SLAM based on dynamic feature point elimination in dynamic environment[J]. IEEE Access, 2023, 11: 113952-113964.

[18] Chang Z, Wu H, Li C. YOLOv4-tiny-based robust RGB-D SLAM approach with point and surface feature fusion in complex indoor environments[J]. Journal of Field Robotics, 2023, 40(3): 521-534.

[19] Xiao L, Wang J, Qiu X, et al. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment[J]. Robotics and Autonomous Systems, 2019, 117: 1-16.

[20] Liu Y, Miura J. RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods[J]. IEEE Access, 2021, 9: 23772-23785.

[21] Ballester I, Fontán A, Civera J, et al. Dot: Dynamic object tracking for visual slam[C]. 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, 11705-11711.

[22] Cui L, Ma C. SOF-SLAM: A semantic visual SLAM for dynamic environments[J]. IEEE Access, 2019, 7(2): 166528-166539.

[23] Gonzalez M, Marchand E, Kacete A, et al. Twistslam: Constrained slam in dynamic environment[J]. IEEE Robotics andAutomation Letters, 2022, 7(3): 6846-6853.

[24] Wadud, R.A., & Sun, W. DyOb-SLAM: Dynamic Object Tracking SLAM System[J]. arXiv preprint, 2022, arXiv: 2211.01941.

[25] Chang J, Dong N, Li D. A real-time dynamic object segmentation framework for SLAM system in dynamic scenes[J]. IEEE Transactions on Instrumentation and Measurement, 2021, 70: 1-9.

[26] He J, Li M, Wang Y, et al. OVD-SLAM: An online visual SLAM for dynamic environments[J]. IEEE Sensors Journal, 2023, 23(12): 13210-13219.

[27] You Y, Wei P , Cai J, et al. MISD-SLAM: multimodal semantic SLAM for dynamic environments[J]. Wireless Communications and Mobile Computing, 2022, 2022(1): 7600669.

[28] Shuangquan Han, Zhihong Xi. Dynamic scene semantics SLAM based on semantic segmentation[J]. IEEE Access, 2020, 8: 43563-43570.