Advances in Computer, Signals and Systems (2025)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2025.090107
ISSN 2371-8838 Vol. 9 Num. 1

Visual Semantic SLAM with Compensation of Edge Feature Points in Dynamic Scenes

Tianwei Liu*

Yunnan Normal University, Kunming, China
774920289@qq.com
**Corresponding author*

Keywords: Dynamic Scenes, Mobile Robots, Semantic SLAM

Abstract: Mobile robots operating in dynamic environments are easily affected by moving objects, which results in significant discrepancies between the localization and mapping results provided by their SLAM systems and the actual conditions. To address this issue, this paper proposes a visual semantic SLAM approach for mobile robots in dynamic scenes. First, a lightweight instance segmentation network, Yolact++, is employed to segment and detect objects in the scene, removing feature points from the regions of the a priori dynamic objects. Furthermore, an LBP (Local Binary Pattern) algorithm that incorporates depth information is designed to prevent the misclassification of edge feature points from dynamic objects, thus preserving static features for the system's pose estimation. A series of simulation analyses validate the performance advantages of the proposed method.

1. Introduction

Today, with the widespread application of intelligent robots across various fields, Simultaneous Localization and Mapping (SLAM), a key enabling technology for autonomous navigation, has become a cutting-edge research hotspot. Among these, visual SLAM (vSLAM) using cameras as sensors has garnered significant attention. In recent years, several excellent vSLAM technologies have been proposed, such as direct methods like DTAM [1], LSD-SLAM [2], and DSO [3], feature-based methods like ORB-SLAM2 [4], and point-line feature-based PL-SLAM [5]. These algorithms perform well in static environments, but in real-world applications, they are affected by dynamic objects in the scene, leading to substantial error accumulation during the localization process. To address the challenges posed by dynamic scenes, the main research directions are divided into traditional methods and deep learning-based approaches. Traditional methods, such as multi-view geometry constraints and optical flow, perform well in low-dynamic scenarios and have the advantages of real-time processing with low resource consumption. However, in complex dynamic scenes, traditional methods struggle to achieve optimal performance. On the other hand, deep learning-based semantic segmentation networks can identify prior dynamic objects in the input image, allowing the system to eliminate dynamic feature points in the target regions, significantly improving the system's robustness in dynamic environments.

Based on this, we introduce the instance segmentation network Yolact++ [6] into the system's frontend. By utilizing the semantic information from the detection and segmentation results, we remove feature points from dynamic regions in the image. The image segmentation masks provided by Yolact++ have relatively poor accuracy at the edges, which can lead to edge points being incorrectly classified as static feature points. To compensate for the mask's segmentation accuracy and prevent error accumulation, we design a dynamic mask edge feature point compensation strategy that integrates depth information with the LBP algorithm.

The remainder of this paper is organized as follows. Section 2 provides an overview of related work. Section 3 presents the algorithm implementation process. Section 4 presents simulation studies and experimental results. Section 5 concludes the paper and discusses future research directions.

2. Related Work

Traditional methods to address the impact of dynamic scenes on system robustness often rely on mathematical and multi-view geometry techniques [7], such as Bayesian network models, clustering, and geometric constraints. Cheng et al. [8] designed a sparse motion removal (SMR) model based on the Bayesian framework, utilizing the similarities and differences between different frames. This model reduces the uncertainty in dynamic region detection and significantly improves the system's robustness in dynamic environments. Dai et al. [9] used the Delaunay triangulation algorithm to construct a sparse graph of map points, determining their correlation based on the relative positions between two points. They then classified the map points into different groups, assuming the largest group as the reliable static map points for pose estimation. Additionally, optical flow methods, which represent image motion, have also been favored by researchers. Derome et al. [10] proposed a motion object detection method based on dense stereo matching and optical flow estimation. To reduce the computational cost of optical flow estimation, a fast algorithm based on the Lucas-Kanade paradigm was used. Eppenberger et al. [11] combined dense optical flow with depth information to achieve map mapping and obstacle avoidance in dynamic scenes. However, traditional methods lack high-level understanding of the scene and cannot meet more advanced requirements. As a result, some research has started to introduce deep learning techniques.

Semantic segmentation is performed on the input image stream. After obtaining semantic information from the scene, it helps the system better classify dynamic and static features. Yu et al. [12] proposed DS-SLAM based on ORB-SLAM2, using SegNet for semantic segmentation of the input image. The segmentation results were then combined with an optical flow-based motion consistency check to remove dynamic features and retain static features for tracking. Dyna SLAM [13] used semantic information extracted by Mask R-CNN to detect prior dynamic objects and employed multi-view geometry methods to detect non-prior dynamic objects. Although deep learning-based semantic segmentation and object detection networks have been evolving, they still struggle to balance speed and accuracy, with detection misses or suboptimal segmentation precision being a significant challenge. Jin et al. [14] designed a segmentation missing compensation algorithm by detecting the difference in masks between consecutive frames. Wang et al. [15] proposed a mask refinement compensation algorithm based on a constant-speed model and a depth-based region-growing algorithm to improve Yolact’s segmentation accuracy. Wei et al. [16] used a depth-based region-growing algorithm to optimize point cloud maps and compensate for over-segmentation when constructing semantic maps, with the threshold for the region-growing algorithm obtained by random sampling within the mask region. Zheng et al. [17] introduced RLD-SLAM, which corrects images using an IMU or keyframes when the camera rotates, optimizing the performance of the YOLOv5 object detection network.

3. Framework and Methods

The visual semantic SLAM algorithm with edge feature point compensation for dynamic scenes developed in this paper is an improvement based on the ORB-SLAM2 algorithm, as shown in Figure 1. The framework primarily includes modules such as semantic segmentation, feature point classification, pose estimation, keyframe selection, loop closure detection, Bundle Adjustment (BA) optimization, object tracking, and map building.

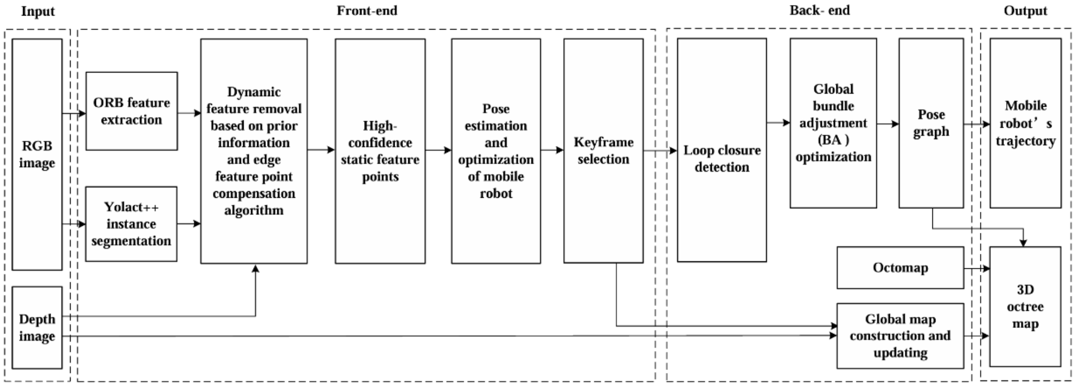
Input
Front-end
Back- end
Output
RGB image
Depth image
ORB feature extraction
Yolact++ instance segmentation
Dynamic feature removal based on prior information and edge feature point compensation algorithm
High-confidence static feature points
Pose estimation and optimization of mobile robot
Keyframe selection
Loop closure detection
Global bundle adjustment (BA) optimization
Pose graph
Octomap
Global map construction and updating
Mobile robot' s trajectory
3D octree map

Figure 1: Framework of Visual Semantic SLAM Algorithm with Edge Feature Point Compensation for Dynamic Scenes.

3.1. Dynamic Feature Detection and Removal

To accurately detect and identify prior objects in the input images while ensuring good real-time performance, we introduced the lightweight instance segmentation network Yolact++ into the system's front-end. This model has slightly lower segmentation accuracy compared to networks like Mask-RCNN, but it offers better real-time performance. Additionally, the Yolact++ model has been trained on the MS COCO dataset [18], which contains 80 categories, including common dynamic objects, and achieves a speed of 33.5 fps with a mAP of 34.1 on MS COCO.

First, Yolact++ preprocesses the input RGB image by resizing it to 550×550 pixels before passing it through the backbone network for convolutional processing. The backbone network uses ResNet101, specifically from Conv1_x to Conv5_x, with the convolutional layers from Conv3_x to Conv5_x replaced by 3×3 deformable convolutions. The processed image is then passed to the Feature Pyramid Network (FPN) for convolution and downsampling. After processing by the FPN, the data is sent to both the prediction network and the prototype network. The prediction network outputs three results: class confidence, location offsets, and mask confidence. Based on this, the Fast NMS algorithm is applied to select the optimal ROI (Region of Interest) for each object. The mask for each class is obtained through convolution in the Protonet. Finally, the images from NMS and Protonet are assembled, cropped, and filtered with a threshold to produce the final output.

To balance the system's real-time performance, we designed a semantic segmentation thread that operates synchronously with the tracking thread. After Yolact++ processes the current image, the semantic segmentation thread provides the semantic information of the current frame to the tracking thread. Based on the semantic information in the image, we classify objects that are typically in motion, such as people, as prior dynamic objects. The feature points in the regions of these objects are removed to avoid the introduction of dynamic feature points. For the i-th prior dynamic object, the feature points within its mask region form a feature point set FP_{object}^{i}, which is treated as a dynamic feature point set during pose estimation and excluded from tracking.

It should be noted that when classifying feature points according to the mask of prior dynamic objects, most prior dynamic objects are non-rigid targets. The masks provided by Yolact++ do not perfectly align with these non-rigid targets, leading to some edge feature points of prior dynamic objects not being correctly classified. To address this, we have designed an edge feature point compensation strategy specifically for prior dynamic objects.

3.2. Dynamic Mask Edge Feature Point Compensation Strategy Using Depth Information and LBP Algorithm

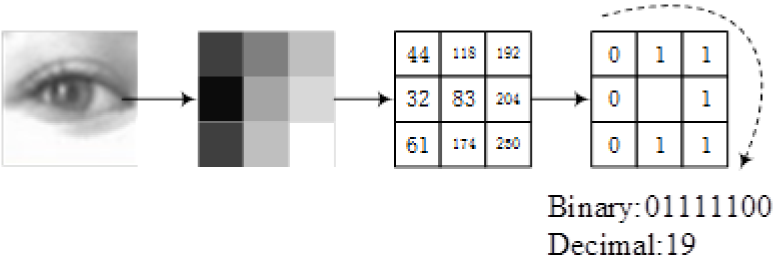

Figure 2: LBP Calculation Process.

Local Binary Pattern (LBP) is an operator used to describe local features of an image, with advantages such as gray-scale invariance and rotational invariance. The original LBP operator is defined in a 3×3 neighborhood, where the gray value of the center pixel in the neighborhood serves as the threshold, and it is compared with the gray values of the eight neighboring pixels. If the gray value of a surrounding pixel is greater than the threshold, the pixel is marked as 1; otherwise, it is marked as 0. This process generates an 8-bit binary number. These 8 binary digits are arranged in a specific order to form a binary number, which represents the LBP value of the center pixel, as shown in Figure 2.

Based on this, we integrate depth information into the LBP calculation. Typically, depth images provided by depth cameras reflect depth information in grayscale values. Therefore, when calculating the LBP value, we no longer convert the RGB image to grayscale but instead use the depth image directly. In this case, the LBP value of a pixel reflects the depth features around that point. The LBP feature that combines depth information is referred to as D-LBP. For the i-th prior dynamic object in the image, we randomly select several feature points from the object's FP_{object}^{i} and read their depth values X_{depth} in the depth image to calculate their respective D-LBP values X_{D-LBP}. The outliers are removed, thus determining the depth range $[a,b]_{depth}$ and D-LBP value range $[a,b]_{D-LBP}$ of the feature points on that object. In practical experiments, we found that to save computational resources and avoid interference from outliers, we only sample from FP_{object}^{i}, which may not provide an accurate $[a,b]_{depth}$ and $[a,b]_{D-LBP}$. Therefore, in actual applications, we slightly expand the ranges $[a,b]_{depth}$ and $[a,b]_{D-LBP}$ to improve accuracy. Yolact++ not only provides the object's mask but also the bounding box. Although the accuracy of the mask may not perfectly match the actual object region, the bounding box fully encompasses the object. Therefore, the edge feature points of non-rigid objects will certainly lie within the bounding box. For the feature points within the bounding box of the i-th prior dynamic object, if the point is not in FP_{object}^{i} and satisfies equation (1), it is considered to also belong to the FP_{object}^{i}, i.e.

$$X_{depth}^{i} \in [a,b]_{depth} \ and \ X_{D-LBP}^{i} \in [a,b]_{D-LBP} \tag{1}$$

4. Simulation Analysis

This section presentss the simulation analysis of the algorithm on different datasets. All experiments were conducted on a laptop with the following configuration: CPU: i7-13700H, 16GB RAM, GPU: NVIDIA RTX4060, and the operating system: Ubuntu 20.04.

To validate the performance advantages of the proposed algorithm, we used the TUM dataset [19] and the BONN dataset [20] to perform a comparative analysis of the simulation results of different algorithms. In the simulation, we used Absolute Trajectory Error (ATE/m) and Relative Pose Error (RPE/m) as the main metrics to evaluate the accuracy of different algorithms, where RPE is further divided into Relative Translation Error (RTE/m) and Relative Rotation Error (RRE/ °). The Absolute Trajectory Error (ATE) represents the direct difference between the estimated pose and the ground truth pose, providing an intuitive reflection of the algorithm's accuracy and the global consistency of the trajectory. The Relative Pose Error (RPE) mainly describes the accuracy of the pose difference between two frames separated by a fixed time gap (relative to the ground truth pose), which is equivalent to directly measuring the odometry error. We then selected Root Mean Square Error (RMSE) to evaluate these errors.

From Figure 3, we could clearly observe that on the TUM *fr3/walking_xyz* dataset and the BONN *crowd3* dataset, our algorithm was able to effectively eliminate feature points in the dynamic object areas, thus preserving the static feature points in the scene and preventing the introduction of errors.

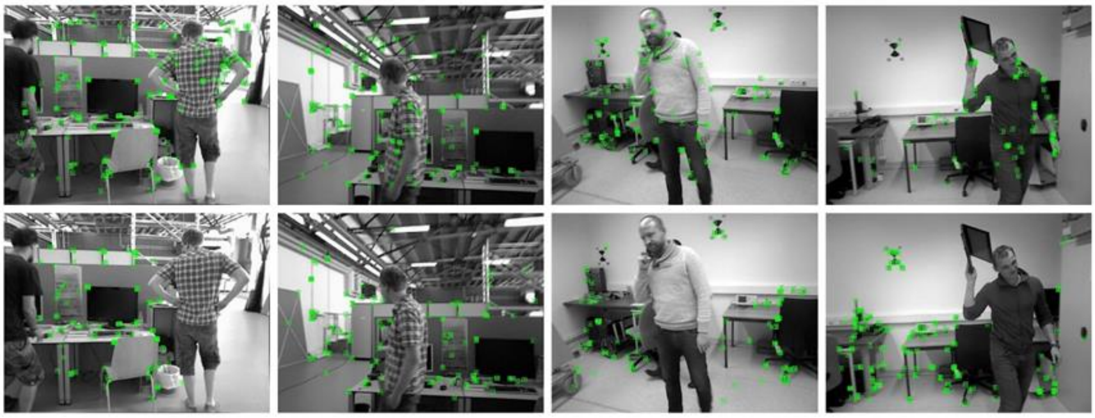

Figure 3: Dynamic removal effects of different algorithms on the TUM and BONN datasets. The first row shows ORB-SLAM2, and the second row shows our algorithm.

In addition, we selected four dynamic datasets from the TUM dataset: *fr3/walking_xyz*, *fr3/walking_half*, *fr3/walking_rpy*, and *fr3/walking_static*, and quantitatively compared the RMSE of ATE and RPE for different algorithms on these four sequences in Tables 1 to 3. It can be observed that our algorithm performs similarly to the other algorithms. This is because our algorithm only removes dynamic features based on the masks of prior dynamic objects, without considering some non-prior dynamic scenarios. However, with the edge feature point compensation algorithm we designed, we effectively removed all dynamic feature points generated by prior dynamic objects, ensuring that the algorithm performs well. DS-SLAM [12] combines SegNet with motion consistency detection methods to determine the motion status of prior dynamic objects, but its mask accuracy is poor. DynaSLAM [13] combines Mask R-CNN with multi-view geometry methods, but there are cases where fewer usable feature points are available. DN-SLAM [21] uses optical flow to optimize the masks obtained from coarse segmentation, but the optimization method of the masks is affected by both the coarse segmentation network and the optical flow. DynaTM-SLAM [22] matches the detection boxes of the same prior dynamic object and determines whether the object is moving based on the similarity within the detection box, but this method is overly reliant on the accuracy of the object detection network. Additionally, compared to ORB-SLAM2, the proposed algorithm showed improvement in ATE on the TUM dataset. The degree of improvement is calculated using Equation (2), i.e.

$$\eta = \frac{\gamma - \phi}{\gamma} \times 100 \tag{2}$$

where, η denotes the improvement in RMSE, γ and ϕ denoting the RMSE values for ORB-SLAM2 and our algorithm, respectively. From Table 1, it can be observed that for these four high-dynamic scene datasets, the improvement of our algorithm is over 94%.

Table 1: RMSE comparison of Absolute Trajectory Error (ATE/m) for different algorithms on the TUM series datasets.

Sequences	ORB-SLAM2	DS-SLAM	Dyna SLAM	DN-SLAM	DynaTM-SLAM	Ours	Improvement (%)
	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
fr3_w_xyz	0.7521	0.0246	0.0155	0.015	**0.0149**	0.0151	98.00%
fr3_w_static	0.39	0.0082	0.0069	0.008	**0.0067**	0.0079	97.97%
fr3_w_rpy	0.8705	0.4438	0.0378	0.032	**0.0287**	0.0316	96.37%
fr3_w_half	0.4563	0.0911	**0.0257**	0.026	0.0291	0.027	94.08%

Table 2: RMSE comparison of Relative Trajectory Error (Translation part) (RPE/m) for different algorithms on the TUM series datasets.

Sequences	ORB-SLAM2	DS-SLAM	Dyna SLAM	DN-SLAM	DynaTM-SLAM	Ours
	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
fr3_w_xyz	0.4124	0.0333	0.0254	0.024	0.0191	**0.0189**
fr3_w_static	0.2162	0.0102	0.0133	0.011	**0.0088**	0.0097
fr3_w_rpy	0.4249	0.1503	0.0415	0.065	**0.0356**	0.0395
fr3_w_half	0.355	0.0297	0.0394	0.035	**0.0281**	0.0289

Table 3: RMSE comparison of Relative Trajectory Error (Rotation part) (RPE/ °) for different algorithms on the TUM series datasets.

Sequences	ORB-SLAM2	DS-SLAM	Dyna SLAM	DN-SLAM	DynaTM-SLAM	Ours
	RMSE	RMSE	RMSE	RMSE	RMSE	RMSE
fr3_w_xyz	7.7432	0.8266	0.6252	-	**0.6006**	0.6202
fr3_w_static	3.8958	0.269	0.3	-	**0.2510**	0.2573
fr3_w_rpy	8.0802	3.0042	0.9047	-	**0.8228**	0.9469
fr3_w_half	7.3744	0.8142	0.8933	-	**0.7443**	0.8151

Figure 4 demonstrates the effect of our edge feature point compensation strategy. As shown in Figure 4(a), although feature points on the moving pedestrian were removed based on the prior dynamic object mask, some feature points at the edges of the pedestrian were still retained. After incorporating the edge feature point compensation strategy, Figure 4(b) clearly shows that the feature points at the edges of the pedestrian were also correctly removed.

To evaluate the individual contribution of each component in the dynamic removal algorithm, Table 4 compared the RMSE of Absolute Trajectory Error (ATE/m) using different methods on the TUM high-dynamic scene dataset. Here, "Y" indicates that only Yolact++ was used for image segmentation, while "Y+D" indicates the application of the dynamic mask edge feature point compensation strategy that integrates depth information and the LBP algorithm on the Yolact++ segmentation results. It can be observed that relying solely on Yolact++, when the segmentation result is incomplete, affects the accuracy of the algorithm. The use of the edge feature point compensation strategy effectively removes the impact of dynamic object edge points on the algorithm's accuracy.

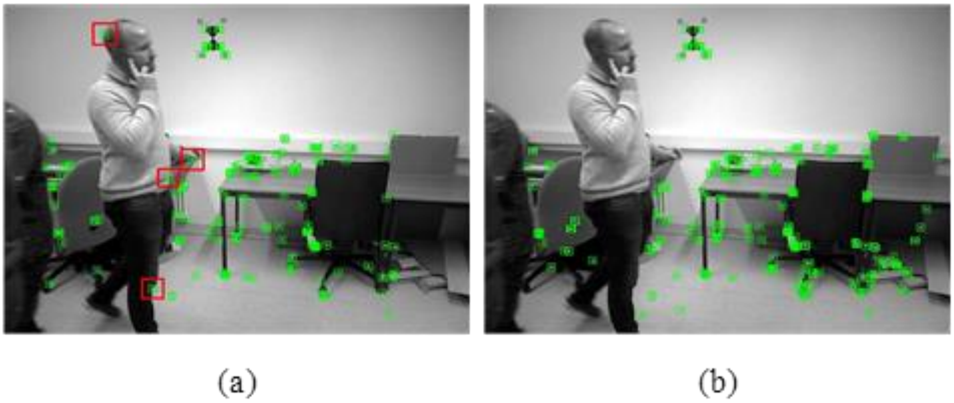

Figure 4: Demonstration of the Edge Feature Point Compensation Strategy.

Table 4: Ablation Study Results Comparison of Different Algorithms on the TUM Dataset.

equences Sequences	Y	Y+D
	RMSE(/m)	RMSE(/m)
fr3/walking_xyz	0.015793	0.015144
fr3/walking_rpy	0.03124	0.030661
fr3/walking_half	0.027543	0.026522

5. Summary and Outlook

This paper presents a visual semantic vSLAM algorithm for dynamic scenes based on ORB-SLAM2. By introducing the lightweight semantic segmentation network Yolact++, the algorithm provides prior dynamic object masks to remove dynamic features in the scene and retain high-confidence static feature points. Additionally, to address the issue of incorrect classification of edge feature points due to insufficient mask accuracy, a dynamic mask edge feature point compensation strategy combining depth information and the LBP algorithm was designed. The feasibility and effectiveness of the proposed algorithm were validated through simulation studies with various datasets.

In future work, we will further explore the use of semantic information combined with traditional methods to assist in better determining the true motion status in the scene, thus preserving as many usable static feature points as possible to improve the system’s accuracy.

References

[1] R. A. Newcombe, S. J. Lovegrove and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," 2011 International Conference on Computer Vision, Barcelona, Spain, 2011, pp. 2320-2327, doi: 10.1109/ICCV. 2011.6126513.

[2] Engel, Jakob, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-scale direct monocular SLAM." European conference on computer vision. Cham: Springer International Publishing, 2014.

[3] ENGEL J, KOLTUN V, CREMERS D. Direct sparse odometry [J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(3): 611-625.

[4] MUR-ARTAL R, TARDÓS J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras [J]. IEEE transactions on robotics, 2017, 33(5): 1255-1262.

[5] PUMAROLA A, VAKHITOV A, AGUDO A, et al. PL-SLAM: Real-time monocular visual SLAM with points and lines; proceedings of the 2017 IEEE international conference on robotics and automation (ICRA), F, 2017 [C]. IEEE.

[6] ZHOU C. Yolact++ Better Real-Time Instance Segmentation [M]. University of California, Davis, 2020.

[7] WANG K, MA S, CHEN J, et al. Approaches, challenges, and applications for deep visual odometry: Toward complicated and emerging areas [J]. IEEE Transactions on Cognitive and Developmental Systems, 2020, 14(1): 35-49.

[8] 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.

[9] 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.

[10] DEROME M, PLYER A, SANFOURCHE M, et al. Moving object detection in real-time using stereo from a mobile platform [J]. Unmanned Systems, 2015, 3(04): 253-266.

[11] EPPENBERGER T, CESARI G, DYMCZYK M, et al. Leveraging stereo-camera data for real-time dynamic obstacle detection and tracking; proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), F, 2020 [C]. IEEE.

[12] YU C, LIU Z, LIU X-J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments; proceedings of the 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), F, 2018 [C]. IEEE.

[13] BESCOS B, F áCIL 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.

[14] JIN J, JIANG X, YU C, et al. Dynamic visual simultaneous localization and mapping based on semantic segmentation module [J]. Applied Intelligence, 2023, 53(16): 19418-19432.

[15] WANG X, ZHENG S, LIN X, et al. Improving RGB-D SLAM accuracy in dynamic environments based on semantic and geometric constraints [J]. Measurement, 2023, 217: 113084.

[16] WEI S, WANG S, LI H, et al. A Semantic Information-Based Optimized vSLAM in Indoor Dynamic Environments [J]. Applied Sciences, 2023, 13(15): 8790.

[17] ZHENG Z, LIN S, YANG C. RLD-SLAM: A Robust Lightweight VI-SLAM for Dynamic Environments Leveraging Semantics and Motion Information [J]. IEEE Transactions on Industrial Electronics, 2024.

[18] LIN T-Y, MAIRE M, BELONGIE S, et al. Microsoft coco: Common objects in context; proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, F, 2014 [C]. Springer.

[19] STURM J, ENGELHARD N, ENDRES F, et al. A benchmark for the evaluation of RGB-D SLAM systems; proceedings of the 2012 IEEE/RSJ international conference on intelligent robots and systems, F, 2012 [C]. IEEE.

[20] PALAZZOLO E, BEHLEY J, LOTTES P, et al. ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals; proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), F, 2019 [C]. IEEE.

[21] RUAN C, ZANG Q, ZHANG K, et al. DN-SLAM: A Visual SLAM with ORB Features and NeRF Mapping in Dynamic Environments [J]. IEEE Sensors Journal, 2023.

[22] ZHONG M, HONG C, JIA Z, et al. DynaTM-SLAM: Fast filtering of dynamic feature points and object-based localization in dynamic indoor environments [J]. Robotics and Autonomous Systems, 2024, 174: 104634.