Robotica (2025), **43**, pp. 3366–3394
doi:10.1017/S0263574725102506

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

A robust visual simultaneous localization and mapping system for dynamic environments without predefined dynamic labels and weighted features

Shuai Xiang[1,2], Chaoyi Dong[1,2,3], Kang Zhang[1,2], Ge Tai[1,2], Tianyu Yuan[1,2], Haoda Yan[1,2] and Xiaoyan Chen[1,2,3]

[1]College of Electric Power, Inner Mongolia University of Technology, Hohhot, 010080, China
[2]Intelligent Energy Technology and Equipment Engineering Research Centre of Colleges, Universities in Inner Mongolia Autonomous Region, Hohhot, 010051, China
[3]Engineering Research Center of Large Energy Storage Technology, Ministry of Education, Hohhot, 010010, China
Corresponding author: Chaoyi Dong; Email: dongchaoyi@imut.edu.cn

Received: 13 October 2024; **Revised:** 28 April 2025; **Accepted:** 19 August 2025; **First published online:** 17 September 2025

Keywords: vSLAM; dynamic environment; weighted features; semantic information; neural radiation field

Abstract

Visual Simultaneous Localization and Mapping (vSLAM) is essentially limited by the static world assumption, which makes its application in dynamic environments challenging. This paper proposes a robust vSLAM system, RFN-SLAM, which is based on ORB-SLAM3 and does not require preset dynamic labels and weighted features to process dynamic scenes. In the feature extraction stage, an enhanced efficient binary image BAD descriptor is used to improve the accuracy of static feature point matching. Through the improved RT-DETR target detection network and FAST-SAM instance segmentation network, RFN-SLAM obtains semantic information and uses a novel dynamic box detection algorithm to identify and eliminate the feature points of dynamic objects. When optimizing the pose, the static feature points are weighted according to the dynamic information, which significantly reduces the mismatch and improves the accuracy of positioning. Meanwhile, 3D rendering of the neural radiation field is used to remove dynamic objects and render them. Experiments were conducted on the TUM RGB-D dataset, Bonn dataset, and self-collected dataset. The results show that in terms of positioning accuracy, RFN-SLAM significantly outperforms ORB-SLAM3 in dynamic environments. It also achieves more accurate positioning than other advanced dynamic SLAM methods and successfully realizes accurate 3D reconstruction of static scenes. In addition, on the premise of ensuring accuracy, the real-time performance of RFN-SLAM is effectively guaranteed.

1. Introduction

Simultaneous localization and mapping (SLAM) is a key technology that provides accurate position estimation for robots, autonomous vehicles, and other mobile devices in unknown space in real time by integrating multi-sensor data such as cameras, lidars, and IMUs, and generates maps that reflect the surrounding environment [1]. According to the different types of sensors, SLAM technology can be divided into laser SLAM and visual SLAM. Laser SLAM has higher positioning accuracy than visual SLAM, and its theory, technology and product maturity have been widely recognized in industrial applications. However, its sensor cost is relatively high. On the contrary, visual SLAM uses cameras to capture image information such as color and texture in the environment. Its low cost and rich perceptual ability make it gradually attract the attention of researchers and become an important branch of mobile robot research [2]. The framework of visual SLAM mainly includes four parts: front-end visual odometry (VO), back-end (nonlinear) optimization, loop closure detection and mapping.

The continuous advancement of vSLAM technology has promoted the vigorous development of camera-based (monocular, binocular and RGB-D) visual SLAM systems, and many frameworks

© The Author(s), 2025. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

with excellent performance have emerged. For example, PTAM [3], LSD-SLAM [4], SVO [5] and ORB-SLAM 1-3 [6–8] are typical algorithms in the field of visual SLAM. They capture two-dimensional image feature points through the camera, and then use the camera pose calculation to project these feature points into three-dimensional space, so as to obtain their accurate three-dimensional position in the world coordinate system. In this way, positioning and mapping in an unknown environment are realized.

In order to simplify the problem description, these algorithms assume that the working environment is static, rigid body, stable lighting conditions and no obvious human interference. However, the real environment is full of various dynamic objects, such as moving cars, non-stationary people and potential moving objects [9]. The interference of dynamic objects will lead to feature positioning errors and inaccurate association, which will lead to matching failure and affect the accuracy of camera pose estimation, thus reducing the overall positioning accuracy of the system [10, 11]. Therefore, it is very important to improve the SLAM positioning accuracy in dynamic scenes. Many SLAM systems have proposed solutions for dynamic environments, in which dynamic object prediction is performed through geometric methods or deep learning to reduce the impact of dynamic objects on the SLAM system, thereby significantly improving positioning accuracy and achieving robust SLAM in complex environments.

Geometric methods use epipolar relationship, motion consistency and depth difference constraints to identify and exclude dynamic objects. Without prior knowledge, accurate segmentation can be achieved by manipulating the pixels of moving objects, but the processing of complex scenes and motion patterns may not be stable enough [12, 13].

In deep learning, semantic segmentation of moving objects and deletion of dynamic pixels with abnormal values are implemented [14, 15]. Mask R-CNN and other models can be trained using labeled data to detect known moving objects. However, these high-precision methods often require powerful GPUs, which are not suitable for the computing power of small real-time robots. They also cannot handle unknown or unlabeled dynamic objects because of the lack of training samples in such cases [17]. Therefore, in recent years, a SLAM scheme combining deep learning and geometric methods has emerged, which improves the recognition accuracy of dynamic objects by fuzing semantic and geometric information. However, these methods still need to be improved when dealing with static objects before motion.

In order to accurately identify and eliminate dynamic feature points, this paper proposes a dynamic environment semantic SLAM system RFN-SLAM, based on target detection and instance segmentation fusion. RFN-SLAM is an RGB-D camera mode based on ORB-SLAM3, which skillfully integrates dynamic detection function and semantic segmentation module, and is realized by two independent but cooperative threads. First, the target detection obtains the detection box of each object. Then, the dynamic state of the points in each detection box is judged by motion consistency analysis. Then, the dynamic box decision algorithm is used to determine its dynamic attributes. The dynamic box is passed to the instance segmentation thread to obtain the mask of a specific object. Finally, in the Tracking thread, the feature points covered by the mask are excluded. In addition, in the camera pose estimation stage, the pose weight optimization is introduced. Even after the dynamic points are excluded, the static points can be used to accurately estimate the camera position. In the back-end mapping thread, the neural radiance fields (NeRF) [16] technology built by the instant-ngp [17] framework is used to generate 3D dense maps without dynamic objects. The main contributions of this paper are summarized as follows:

- A SLAM algorithm combining RT-DETR [18] improved target detection and FAST-SAM [19] network instance segmentation is proposed to automatically identify potential moving objects without pre-setting object motion characteristics. This fusion scheme shows good robustness in various environments.
- A novel dynamic box decision algorithm is introduced, and the BAD [20] descriptor is used in the feature description stage. In the camera pose estimation, the pose weight optimization is introduced, which helps to improve the accuracy of static point matching after excluding dynamic points.

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

- Combining VO and NeRF framework, 3D dense reconstruction is performed in a dynamic environment using static points. It is particularly emphasized that this method effectively reconstructs background regions occluded by dynamic objects.
- In the experiments of TUM dataset and Bonn dataset, the proposed algorithm performs well in comparison with complex dynamic algorithms, especially in most dataset sequences.

The rest of this paper is constructed as follows: The second chapter summarizes the improvement research of related dynamic SLAM. The third chapter details the improved strategy of RFN-SLAM. Chapters 4 and 5 show the experimental results and the discussions and conclusions based on these results, respectively.

2. The related work

2.1. Dynamic SLAM improved by geometric method

Kim [21] proposed the fusion of RGB color data and depth information to generate three-dimensional feature points. The dynamic object is divided into interior points and abnormal points according to its moving characteristics, and the abnormal detection uses RANSAC algorithm to eliminate outliers. Wang et al [22]. used the fundamental matrix to detect the inconsistency of feature points. The depth image is clustered to form several clusters, and then the basic matrix method is used to eliminate outliers. Then, a moving object recognition model is constructed based on these statistical features. When the contour in the clustering region exceeds the preset threshold, all features of the region are cleared. Song et al [23]. used the optical flow method to eliminate dynamic objects based on the consistency of geometric features. Dai et al [24]. used point correlation technology to identify the static and dynamic attributes of map points.

Geometric methods usually rely on accurate mathematical functions and pixel information, which has the advantage of high speed. However, compared with deep learning, their limitations lie in the lack of advanced understanding ability and the inability to use prior knowledge to analyze the scene, resulting in low robustness. Especially when dynamic objects dominate the scene, the performance of geometric methods is relatively weak.

2.2. Dynamic SLAM improved by deep learning method

The core of the deep learning method is to use the deep network to extract semantic information, which helps to generate prior knowledge. Semantic segmentation or object detection provides clues to dynamic objects (such as people and cars) for SLAM systems, provided that they presuppose that certain categories may move. The system processes this dynamic information by labeling dynamic objects and removing associated feature points. For example, Bescos et al [12]. proposed Dyna-SLAM, which identifies and removes feature points marked as potential dynamic objects in 20 categories through Mask R-CNN without verifying their motion state. For unknown dynamic objects, management relies on multi-view geometric methods. Soares et al [25]. proposed Crowd-SLAM, which uses YOLOv3 technology to directly detect and remove key points in categories (such as people) without additional processing. Ji et al [26]. used semantic segmentation technology to remove objects defined as highly dynamic in 20 categories. For unknown dynamics, it is identified by the reprojection error of the depth image clustering. Zhang et al [27]. proposed Sem-VSLAM based on improved Mask R-CNN, introduced a semantic segmentation thread, combined with motion consistency detection to eliminate dynamic feature points, repaired segmentation missing, and constructed a 3D semantic point cloud map. However, this algorithm has limited coverage of dynamic objects and poor robustness in extreme scenarios. Yu et al [28]. proposed the CPR-SLAM method, which generates sub-point clouds through depth image clustering, uses the structural invariance and relative relationships of static sub-point clouds to detect dynamic regions, and optimizes the dynamic regions of key frames. However, the threshold parameters for sub-point cloud

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

division need to be tuned for specific scenarios, and the generalization ability is limited. Kenye et al [29]. proposed a semantic-assisted segmentation method, which uses a deep neural network to detect dynamic objects and generate binary masks, separates the features of the static background and dynamic objects, performs SLAM only using static features, and simultaneously tracks dynamic features to maintain mask consistency. However, the accuracy of dynamic object detection and mask generation is limited by the model's generalization ability, and it has insufficient adaptability to unlabeled categories or complex occlusion scenarios. Li et al [30]. proposed a lightweight multi-modal semantic framework, which can detect dynamic objects in real-time through an improved lightweight instance segmentation network, effectively suppress dynamic interference, and improve the real-time performance of SLAM. However, multi-modal fusion increases the computational burden of embedded deployment.

In addition, some researchers have developed a classic SLAM framework to identify the motion state of the segmented object. For example, Yu et al [13]. proposed DS-SLAM, which uses SegNet and optical flow technology to eliminate human feature points in images. Fan et al [31]. use BlitzNet to segment objects and remove outliers in several categories marked as movable in the dynamic mask region through epipolar constraints. These methods have limited robustness to unknown objects, and fixed thresholds may lead to over-sensitivity to some objects and insufficient sensitivity to other objects. He et al [32]. proposed the OVD-SLAM algorithm that integrates semantic detection with a dynamic point weighting optimization mechanism, achieving robust camera pose estimation in dynamic environments. However, it struggles to address partial object deformation or subtle motion, exhibiting accumulated localization errors in low-dynamic scenarios. Du et al [33].'s LC-SLAM combines GC-RANSAC-based initial pose estimation with conditional random field-based long-term consistency analysis, enabling more accurate camera tracking in dynamic environments. Nevertheless, its geometric rules prove inadequate for effective dynamic point identification due to the absence of semantic scene understanding, ultimately compromising pose estimation accuracy. Zhang et al [34]. proposed the MR - SLAM system. By combining YOLOX object detection with an adaptive threshold adjustment algorithm, it achieves more robust camera pose estimation and scene reconstruction in dynamic environments. However, this method lacks robustness in local human observation scenarios. The adaptive threshold adjustment mechanism may fail under extreme lighting or complex occlusion conditions, leading to the accumulation of pose estimation errors.

Additionally, Li et al [35]. proposed a static weighting method for depth edge points in key frames. This method assigns static weights to depth edge points obtained from depth discontinuities to reduce the impact of dynamic information. However, this method is weak in distinguishing small and fast-moving objects, and its effect depends on the quality of the dense point cloud. Zhong et al [36]. proposed WF-SLAM. Based on the feature point weighting mechanism, it assigns higher pose optimization weights to static feature points and dynamically adjusts the weights through BA joint optimization, significantly reducing dynamic interference. Zhang et al [37]. proposed VDO-SLAM. It adopts a dynamic feature point weighting mechanism. By detecting the confidence level, it assigns high weights to static points and reduces the weights of dynamic points, strengthening the static constraints. However, this method is prone to mis-weighting in detection blind areas and when facing small objects.

Deep learning networks can improve the performance of SLAM systems in semantic understanding, but they mainly provide semantic information about the possibility of object motion, rather than the actual motion state. This leads to the inability of deep learning methods to accurately determine whether the object is moving. In order to improve, it can be considered to combine other sensor data or geometric information to enhance the recognition of the motion state of the object.

2.3. Dense visual reconstruction method

The technology of 3D dense reconstruction based on computer vision aims to recover the complete 3D structure of a scene from a sequence of 2D images. Its development can be divided into two major stages: traditional methods and deep learning-based methods. Traditional methods, grounded in multi-view geometry theory, first extract and match image feature points using Structure from Motion (SFM)

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

algorithms, optimize to recover sparse point clouds and camera poses through bundle adjustment (BA), and then achieve dense reconstruction via Multi-View Stereo (MVS) techniques. MVS methods can be categorized by scene representation into voxel-based methods, surface evolution methods, and depth map fusion methods, with core challenges including view selection, occlusion handling, and cost function design. Although traditional methods are relatively mature in theory, they exhibit weaker adaptability to complex environments such as those with textureless regions or dynamic scenes.

With the rise of deep learning, multi-view learning methods based on convolutional neural networks (CNNs) have made breakthroughs. MVSNet [38] constructs a cost volume through differentiable homography mapping and infers depth maps using 3D CNNs. R-MVSNet [39] incorporates recurrent neural networks to reduce memory footprint. OctNet optimizes storage efficiency through an octree structure. Another category, neural radiance field-based methods (such as NeRF), employs implicit representations, inputting 5D coordinates into an MLP network to predict volume density and color values, and achieving highly realistic novel view synthesis through volume rendering equations. Additionally, the recently developed 3D Gaussian Splatting discretizes NeRF into Gaussian point clouds [40], enabling efficient training and real-time rendering, albeit with limitations in handling complex geometric details.

This paper selects NeRF as the dense reconstruction method because its implicit representation is more suitable for incremental optimization in SLAM and can generate high-fidelity maps. Although 3D Gaussian Splatting offers fast rendering speeds, it requires explicit storage of Gaussian parameters, leading to inefficient memory usage and difficulty in adapting to real-time updates in dynamic scenes.

In summary, compared with the visual SLAM method applied in the static environment, the performance improvement in the dynamic environment still has great research potential and room for further exploration. Therefore, this paper proposes a dynamic environment semantic SLAM system RFN-SLAM, based on target detection and instance segmentation fusion. The system uses an improved lightweight RT-DETR network and an instance segmentation FAST-SAM network, which are designed for dynamic environments and can accurately detect potential moving targets. Through the dynamic box determination algorithm and pose weight optimization, the dynamic target elimination can be efficiently executed, thereby improving the accuracy of pose estimation. In the SLAM post-processing stage, a 3D dense map reconstruction thread based on the NeRF framework is added, which makes the generated map not contain dynamic objects and achieves more accurate dense map construction.

3. System overview

In this paper, semantic and geometric strategies are used to filter dynamic feature points in images. Firstly, in the feature description stage, the enhanced binary descriptor BAD algorithm is used to improve the matching accuracy. Then, the improved lightweight RT-DETR network is used to obtain the bounding box of the target object. Then, the abnormal points are obtained by motion consistency test. Then, the acquired bounding box motion attributes are determined by the proposed novel dynamic box determination algorithm, and the dynamically attributed bounding boxes are input to the FAST-SAM instance segmentation network, so as to obtain the mask image of the target object, which improves the accuracy and efficiency. In the Tracking task, feature points in the mask image are removed to enhance stability. During pose optimization, camera localization accuracy is further enhanced by weighted optimization of static feature points. In the mapping stage of post-processing, a NeRF-based mapping technology is introduced, which removes dynamic feature points from 3D maps to generate more accurate static maps. The proposed RFN-SLAM system framework is shown in Figure 1.

3.1. Enhanced efficient binary image descriptor: BAD algorithm

In dynamic environments, a dynamic feature point culling strategy is adopted to enhance the accuracy and robustness of system localization. At the same time, static points are retained for feature matching to estimate camera pose. However, the number of feature points extracted based on the ORB-SLAM3

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

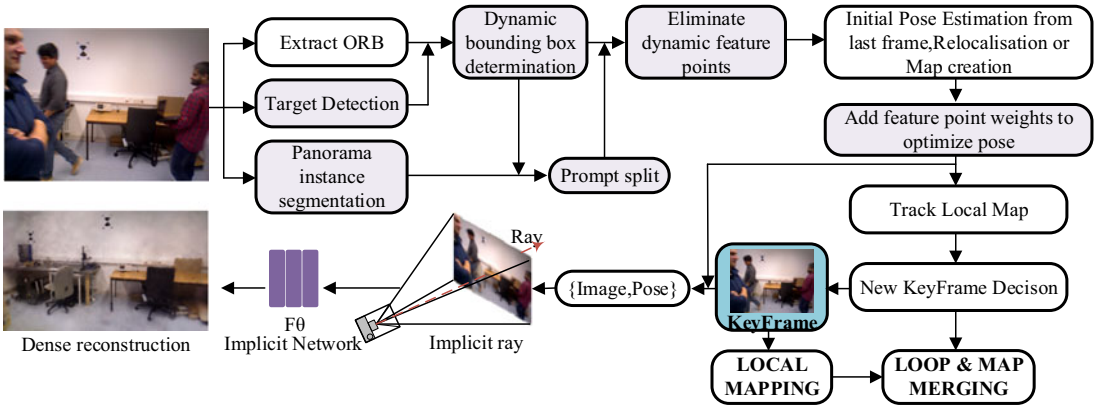


Figure 1. *RFN-SLAM system framework.*

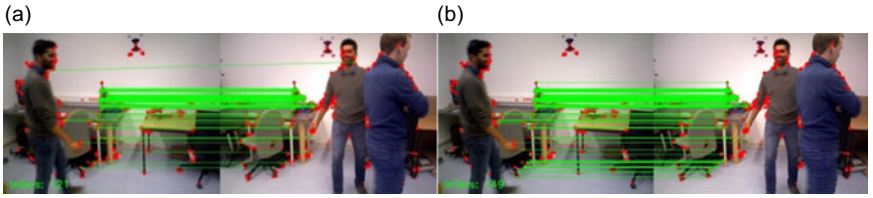


Figure 2. *The matching effects of the two descriptors are compared: (a) the application of the brief descriptor is demonstrated, and (b) the application of the BAD descriptor is demonstrated.*

system is fixed, and the removal of dynamic points may result in a significant reduction in the available feature points for matching. This is particularly obvious when dynamic objects occupy a large imaging area, resulting in a decrease in matching accuracy and even loss of camera tracking. In order to make up for the impact of the reduction of dynamic feature points, the feature description stage is optimized, and more accurate descriptors are selected. Even if the number of feature points is limited, the accuracy of matching can be ensured, which can improve the overall matching performance. Therefore, this paper uses the enhanced efficient binary image descriptor BAD algorithm.

The BAD algorithm introduces a greedy process to select a set of unrelated pixel differences to distinguish similar and different image blocks. The binary weak descriptor $h(x; f, \theta)$ is a decision tree on $f(x, u_1, u_2, s)$, as shown in Formula (1). The feature is obtained by calculating the difference of the average gray value in two boxes of size s, which are centered on the pixel u_i of the image block x. It approximates the gradient of the image in a specific direction and scale, and can be effectively calculated by integrating the image. Learning the size of the box enables the feature to adapt to the optimal scale to describe the image block. By selecting the optimal threshold θ, the average eigenvalue tends to zero. By combining with threshold regression, the descriptor can be binarized with only zero threshold processing.

$$h(x; f, \theta) = \begin{cases} +1\ , \ if\ f(x) \leq \theta \\ -1\ , \ otherwise \end{cases} \tag{1}$$

The Brief descriptor in the original ORB is replaced to significantly improve the accuracy of feature point matching. As shown in Figure 2, Figure (a) shows 121 pairs of feature points matched using the Brief descriptor, and Figure (b) shows 149 pairs of feature points matched using the BAD descriptor. Obviously, the BAD descriptor can capture the details around the feature points more effectively, so as to achieve accurate matching. In addition, even in the case of eliminating some dynamic feature points, it can ensure stable inter-frame matching and avoid mismatching and camera tracking loss.

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

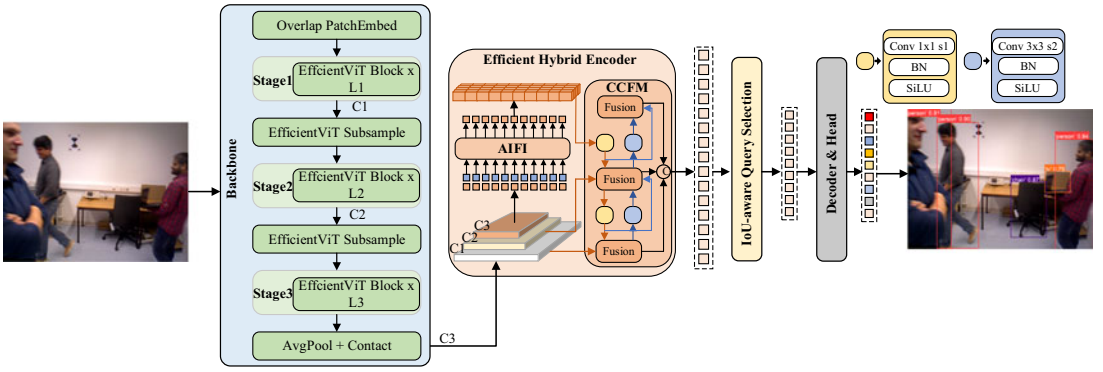


Figure 3. *RT-DETR network structure diagram based on efficientViT backbone network.*

3.2. Semantic information fusion

Traditional dynamic feature elimination methods rely on a clear distinction between dynamic objects (such as high dynamic, low dynamic and static). During detection, moving object features are discarded once detected. However, this approach can lead to false positives, such as stationary cars, resting people, and objects in their hands, which might be mistakenly identified as moving rather than static. In this paper, a culling strategy combining object detection network and instance segmentation is proposed. It does not require preset dynamic information, is suitable for changing environments, and significantly enhances the robustness of the system. Firstly, object detection and instance segmentation are processed by parallel threads to generate semantic information. Then, the outlier detection and dynamic box determination algorithm identifies the bounding box of the dynamic object. These boxes are passed to the instance segmentation network thread, and the mask of the dynamic object is obtained through the network. Then, the feature points in the mask are removed, and only the static points are retained for subsequent calculation. This processing improves efficiency and ensures the separation of dynamic and static information.

3.2.1. Improved lightweight target detection network RT-DETR

RT-DETR is the first real-time end-to-end object detector. This model designs an efficient hybrid encoder. Different from the general DETR series algorithm, the encoder improves the running speed by decoupling intra-scale interaction and cross-scale fusion. Under the same model scale, compared with the YOLO series target detection algorithm characterized by real-time performance, RT-DETR still has advantages in accuracy and speed, and can better meet the real-time and accuracy requirements of SLAM algorithm. This paper uses the EfficientViT [41] network to replace the original backbone network in RT-DETR. The network designs a memory-efficient sandwich layout construction module, using a single memory-constrained MHSA layer between efficient FFN layers. The cascade group attention module is introduced to provide different segmentations of complete features to the attention head to save computing costs and improve attention diversity. Optimize parameter usage by extending the channel width of key network components and narrowing the channel width of those components that are less important.

The schematic structure of RT-DETR after replacing the backbone network with the EfficientViT network is shown in Figure 3. In this paper, we extract three scale outputs (marked as C1, C2, C3) from the EfficientViT backbone network. The output of C3 is processed by AIFI (Intrascale Feature Interaction) module for scale interaction, and then input into CCFM (Cross Scale Feature Fusion Module) module with C1 and C2 to complete cross-scale fusion. Then the output passes through the IoU-Aware query selection layer, the decoder and the auxiliary prediction head layer to obtain the target detection result. On the COCO2017 dataset, the model was trained and validated to be able to detect 80 categories of objects with a comparison of training metrics, as shown in Table I.

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

Table I. *Comparison of key metrics for object detection algorithms.*

Algorithm	Number of parameters (M)	Calculation volume (GFLOPs)	Weight size (MB)	FPS	mAP(%)
RT-DETR-ResNET18	19.97	57.3	38.8	236	41.8
Improvement of RT-DETR	14.99	49.8	21.7	284	42.0

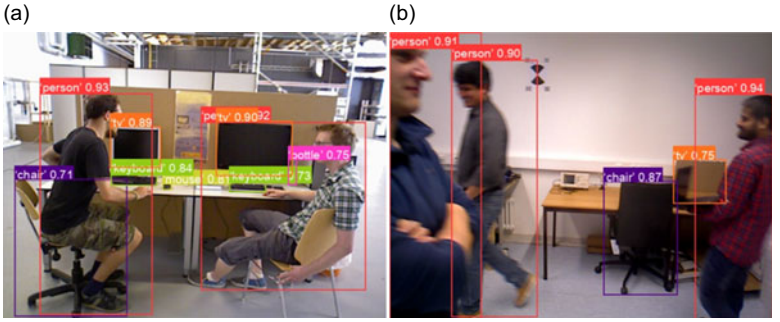


Figure 4. *The improved detection effect of RT-DETR network is demonstrated. (a) presents the detection in the multi-target complex scene, (b) shows the detection in the handheld object scene.*

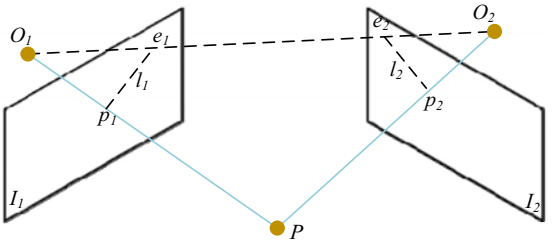


Figure 5. *Contrapolar geometry diagram. By matching the correspondence of feature points between adjacent frames, the epipolar geometric relationship between adjacent frames is established.*

The improved RT-DETR network can accurately detect the bounding box of the object, as shown in Figure 4. Especially for small objects, such as mouse, water cup and keyboard in Fig. (a), and notebook in Fig. (b), they can be effectively identified, and the recognition effect is remarkable. At the same time, the detection time of a single image is maintained at about 4 ms.

The improved RT-DETR network is used to obtain the bounding boxes of the object, and then the key step is to determine whether these bounding boxes are moving. It is necessary to obtain the abnormal points in the current image frame through the pole constraint, and then obtain the motion attributes of the bounding box through the dynamic box determination algorithm. As shown in Figure 5, the projections of P points are obtained from the camera angles O_1 and O_2, respectively, and mapped onto the imaging plane. After eliminating the outliers by RANSAC algorithm and calculating the basic matrix F by optical flow method, the points of the previous frame are mapped to the search area of the current frame, which corresponds to the current epipolar line.

Assume that p_1 and p_2 represent the correct matching points of the previous frame I_1 and the current frame I_2, and P_1 and P_2 represent the corresponding homogeneous coordinates :

$$\begin{aligned} P_1 &= [u_1, v_1, 1]\,, P_2 = [u_2, v_2, 1]\,, \\ p_1 &= [u_1, v_1]\,, p_2 = [u_2, v_2] \end{aligned} \tag{2}$$

where u, v are the coordinate values of the corresponding points in the image. l_2 represents the polar line, and its calculation method is as follows:

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

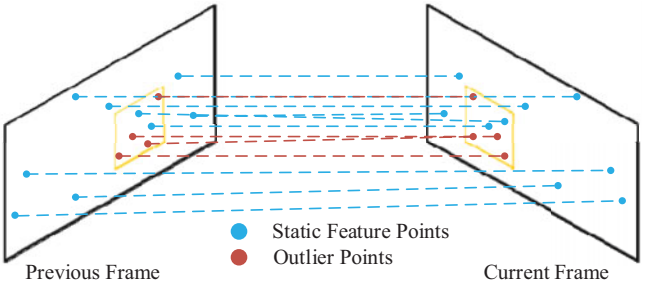


Figure 6. *The feature points in the previous frame are projected to the current frame through the basic matrix. The blue dotted line indicates that the projection between feature points satisfies the basic matrix transformation, and the red dotted line indicates that the projection between feature points does not satisfy the basic matrix transformation.*

$$l_2 = \begin{vmatrix} X \\ Y \\ Z \end{vmatrix} = FP_1 = F \begin{vmatrix} u_1 \\ v_1 \\ 1 \end{vmatrix} \tag{3}$$

where X, Y, Z are line vectors, and F is the basis matrix. When the matching relationship between p_1 and p_2 between two frames is known, the distance between the actual position of p_2 and the ideal epipolar line l_2 is defined as follows:

$$D = \frac{|P_2^T F P_1|}{\sqrt{\|X^2\| + \|Y^2\|}} \tag{4}$$

where D represents the distance. In this paper, the abnormal points are screened from the feature point set by setting the threshold. If the value of D exceeds the preset threshold, the feature point is regarded as an abnormal point, as shown in Figure 6.

Due to the different motion characteristics of people and objects, abnormal values are detected in the bounding box of people, that is, an outlier appears, which is regarded as dynamic, and the box is marked at this time. For other objects, the dynamic ratio relationship is used to judge whether it is dynamic. The ratio of the abnormal value contained in the object box to the total number of optical flow tracking feature points in the box is calculated, as shown in Formula (5). When the ratio of outliers and feature points in the box exceeds the preset threshold, it is judged to be dynamic, otherwise regarded as static.

$$\frac{o}{s} > \alpha \tag{5}$$

where o represents the total number of outliers in the box, s represents the total number of feature points tracked by optical flow in the box, and α represents the preset threshold. After many experiments, setting α to 0.4 shows the best performance in this paper.

3.2.2. Instance segmentation network FAST-SAM

The instance segmentation network adopts FAST-SAM, which is a fast instance segmentation model based on YOLOv8-seg. It can segment panoramic instances of 'zero shot' samples and obtain high-quality segmentation results. Features are extracted from the image through the backbone network and the feature pyramid, and multi-scale features are generated. These features contain rich semantic information. As shown in Figure 7, the segment anything task is divided into two continuous stages: first, the whole instance segmentation is performed, and then the prompt guidance selection is performed. The first stage relies on the implementation of a detector based on CNN, which generates segmentation masks for all instances in the image. Then in the second stage, it outputs the region of interest corresponding to the prompt.

In parallel threads, panoramic segmentation performs segmentation of all objects, accurately segmenting each object or region, regardless of the category of the object. As shown in Figure 8, all objects

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

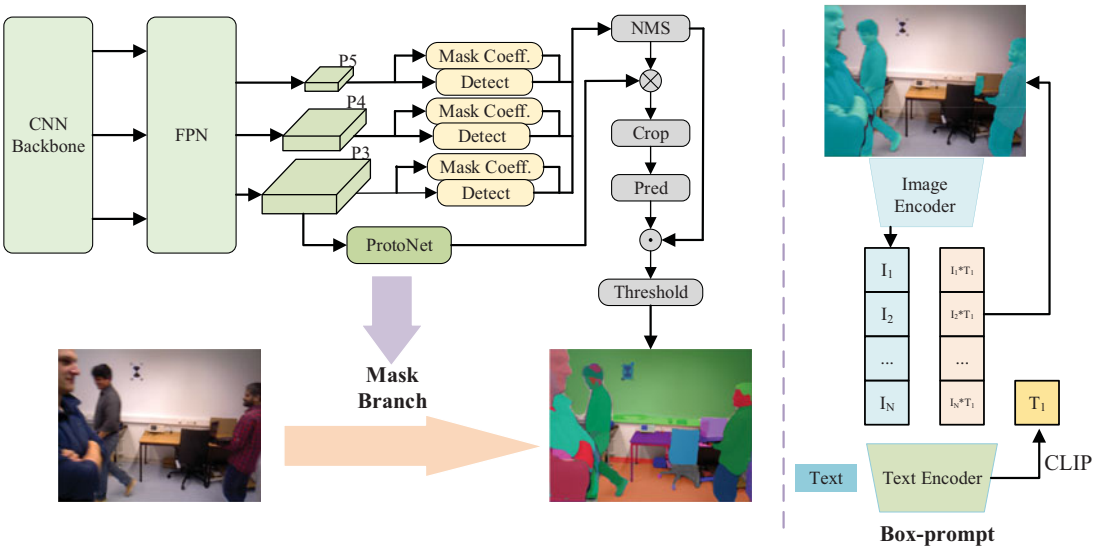


Figure 7. *FAST-SAM instance segmentation network structure diagram.*

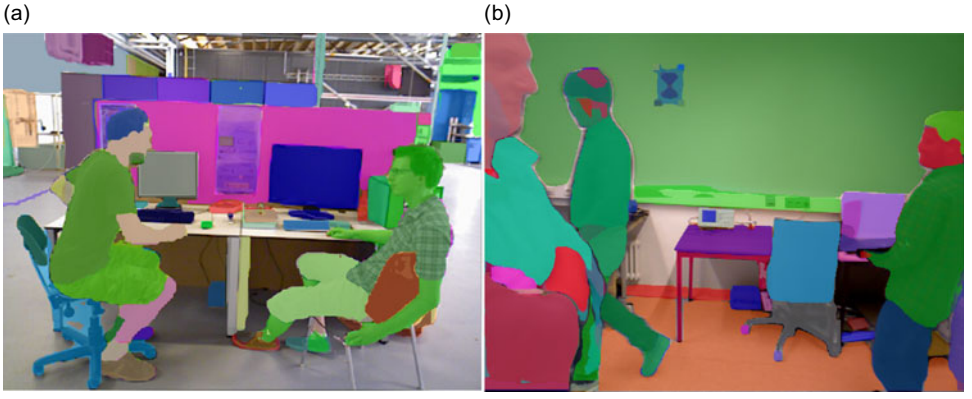


Figure 8. *The FAST-SAM network realizes the panoramic segmentation of the first stage.*

in the image are accurately segmented, providing an accurate segmentation mask for the subsequent prompt segmentation stage.

After completing the panoramic segmentation, the Tracking thread receives the dynamic bounding box input. By prompting segmentation, the mask of a specific object can be obtained, and the feature points that fall within the mask can be removed. This processing can ensure that the feature points on the dynamic object are accurately removed and will not be missed, while avoiding the false recognition of the static feature points, as shown in Figure 9. Figure (a) shows the accuracy of segmentation in the standing scene. It can accurately generate a mask for the individual who is ready to get up without wrongly segmenting the sedentary person. Using the dynamic object mask, the dynamic feature points are effectively removed, and the feature points on the static object are retained. Figure (b) can not only obtain the mask of the obvious area of the dynamic target, but also includes small objects. According to the generated dynamic mask, the corresponding dynamic feature points are excluded.

3.2.3. Semantic information fusion effects

The fusion process based on RT-DETR, FAST-SAM, and epipolar constraints algorithms and the final results are shown in Figure 10. Firstly, ORB feature point extraction, object detection frame extraction,

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press



Figure 9. *The prompt segmentation output and dynamic point elimination in two scenarios are shown. (a) Accurately capture moving objects, accurately identify and eliminate their dynamic feature points, and focus on the identification of motion attributes. (b) shows the excellent recognition of passive moving objects and achieves accurate segmentation, so as to achieve the purpose of dynamic point elimination.*

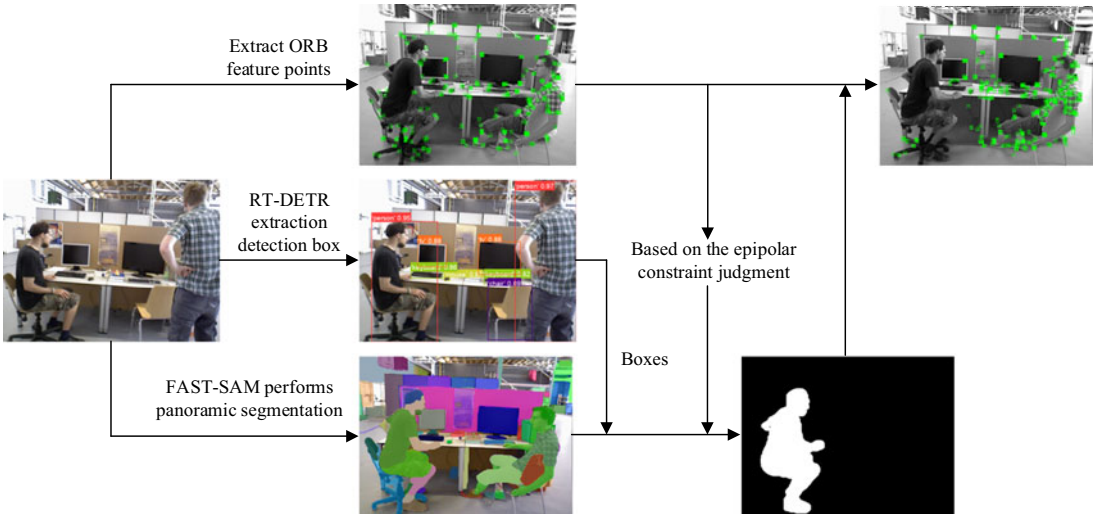


Figure 10. *Fusion process based on RT-DETR, FAST-SAM and epipolar constraint determination and final results.*

and panorama instance segmentation are carried out simultaneously. Subsequently, the extracted feature points are analyzed according to the pairwise geometric decision algorithm to determine the motion attributes of the object detection frames and mark them. If it is labeled as a dynamic frame, it is fed into the second stage of FAST-SAM as a hint. Finally, FAST-SAM outputs a dynamic object mask. Based on this mask, the dynamic feature points are eliminated from the original ORB feature point distribution image, thus preserving the final static feature points.

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

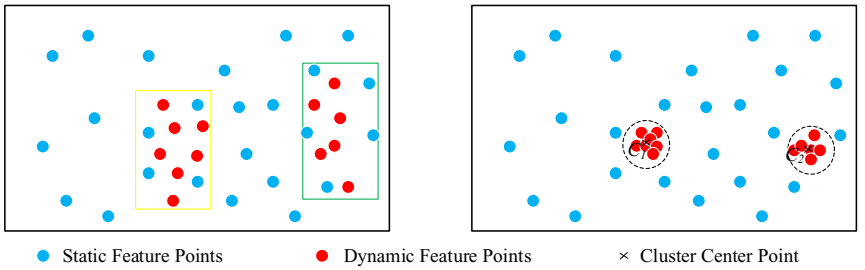


Figure 11. *The k-means clustering algorithm is used to analyze the dynamic feature points in the image frame. According to the number of dynamic objects, the corresponding clustering operation is performed. Finally, each cluster will have a center point as a representative.*

3.3. Pose weight optimization

The tracking module of ORB-SLAM3 first extracts features and matches them, and then initializes the camera pose according to the motion model. Then, the matching feature points are used for BA optimization, and the camera pose is estimated by minimizing the reprojection error of the 3D point X_i in the world coordinate system and the key point x_i in the image frame. The camera rotation R and translation t are optimized through this process. The reprojection error is calculated by the following formula:

$$\{R^*, t^*\} = \underset{R,t}{\operatorname{argmin}} \frac{1}{2} \sum_{i=1}^{n} \| x_i - \pi \left(RX_i + t \right) \|_2^2 \tag{6}$$

where π represents the projection of the three-dimensional coordinate to the pixel coordinate system, and R^* and t^* represent the optimized pose.

In ORB-SLAM3, 3D map points are usually added to the objective function without distinguishing between static and dynamic. However, in a dynamic environment, the matching feature points are inevitably mixed with dynamic objects. If the optimization objective function contains dynamic points, it will affect convergence and lead to obvious attitude deviation. After eliminating the dynamic feature points, the remaining static points are used to optimize the pose. However, this may lead to a lack of features in some areas, while the features in other areas are dense, and it is difficult to take into account the global effect when optimizing. When dealing with feature points, considering the balance of spatial distribution, the weight value of each static feature point is given to optimize the overall performance.

In order to improve the accuracy and robustness of the whole SLAM system, a weighted feature point based on dynamic point elimination is introduced. By jointly optimizing the pose and weight, the BA method is used to estimate and optimize. In the optimization process, the weight Q_i is assigned to the feature point $p_i(u, v)$, and Q_i represents the reliability of $p_i(u, v)$. The larger the value, the stronger the reliability. Based on this static characteristic, the weight of feature points is calculated.

Firstly, after extracting the dynamic feature points from the dynamic mask, the k-means clustering algorithm is used to cluster these points, and the corresponding number of clusters is performed according to the number of moving objects in the image. The center point of each cluster is represented as C_i, as shown in Figure 11.

Next, the Euclidean distance S_i from each static feature point to each cluster center C_i is calculated. Then, the average distance S from the static feature points to all cluster centers is calculated. Finally, according to the preset segmentation threshold, the weight value Q_i of each feature point is determined, as shown in Figure 12.

The static feature points close to the dynamic points have low output values and relatively small contribution to pose optimization, which helps to reduce the fluctuations caused by dynamic objects. In contrast, static feature points away from dynamic points have higher output values and are more important for pose optimization. These points are more stable and reliable.

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

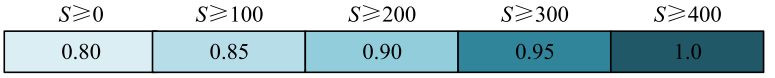


Figure 12. *The weight distribution of static feature points is shown. The weight of each static feature point is calculated according to its average distance to the clustering center of each dynamic object, and then these weights are integrated into the pose optimization thread.*

For optimization, the reprojection error function is multiplied by the weight values of the feature points; feature points with larger weights are more reliable and contribute more to the objective function, and feature points with smaller weights contribute less. The optimization variable has been changed from the original camera pose to camera pose and the weights of feature points. The optimized feature point weight can more accurately characterize the reliability of the point, and its objective function is updated according to the weighted projection error:

$$\{R^*, t^*, Q^*\} = \underset{R,t,w_i}{\operatorname{argmin}} \frac{1}{2} Q_i \sum_{i=1}^{n} \| x_i - \pi \ (RX_i + t) \|_2^2 \tag{7}$$

where Q_i is the weight of map point x_i.

In pose estimation, the optimization is performed jointly for the rotation matrix R, translation vector t, and the weights Q_i of the map points X_i. The Levenberg–Marquardt algorithm in the g2o library is used to deal with the joint optimization task. At the same time, it is necessary to manually calculate the partial derivative of the error term to the optimization variable, so the partial derivative of the feature point weight is:

$$\frac{\partial e}{\partial Q_i} = \begin{bmatrix} u_i - \left(\frac{X}{Z} f_x + c_x \right) \\ v_i - \left(\frac{Y}{Z} f_y + c_y \right) \\ u_i - \left(\frac{X}{Z} f_x + c_x - \frac{b_f}{Z} \right) \end{bmatrix} \tag{8}$$

where f_x, f_y, c_x, c_y and b_f are camera parameters, $P = (X \ Y \ Z)^T$ is the spatial point coordinates in the camera coordinate system, and (u_i, v_i) is the corresponding pixel coordinate point.

3.4. NeRF dense map

NeRF reconstructs 3D scenes by training a series of 2D images from different viewpoints. The continuous scene can be expressed as a function $F_\Theta : (\mathbf{x}, \mathbf{d}) \rightarrow (\mathbf{c}, \sigma)$, where $\mathbf{x}$ is the 3D coordinate position of a spatial point, and $\mathbf{d} = (dx, dy, dz)$ is the 3D Cartesian unit vector representing the direction (θ, ϕ). $\mathbf{c} = (r, g, b)$ is the 3D point color of the point related to the perspective, and σ is the density corresponding to the 3D position.

By using a multi-layer perceptron (MLP) network model to approximate the mapping relationship between input and output, an 'implicit representation' of the 3D scene is created. According to the above definition, for any light $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$, its light color $C(r)$ in the near boundary and far boundary interval $[t_n, t_f]$ can be expressed by integral:

$$C\ (r) = \int_{t_n}^{t_f} T\ (t) \cdot \sigma\ (\mathbf{r}\ (t)) \cdot \mathbf{c}\ (\mathbf{r}\ (t)\ , \mathbf{d})\ dt \tag{9}$$

where $T(t)$ is the cumulative transparency of the light from t_n to t:

$$T\ (t) = \exp\left(- \int_{t_n}^{t} \sigma\ (\mathbf{r}\ (s))\ ds \right) \tag{10}$$

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

In the NeRF rendering step, the color and voxel density of the three-dimensional points obtained during the reconstruction process are used to integrate along the ray and obtain the final two-dimensional image pixel value.

In practice, it is challenging for NeRF to accurately estimate continuous 3D information, and the corresponding implementation code cannot perform continuous integration. Therefore, it is necessary to discretize on the basis of Eq. (12). N points are sampled uniformly in the area where the quadrature is needed for approximate calculation. The i-th sampling point can be expressed as:

$$t_i = \mathbf{U}\left[t_n + \frac{i-1}{N}\left(t_f - t_n\right), t_n + \frac{i}{N}\left(t_f - t_n\right)\right] \tag{11}$$

based on these sampling points, $C(r)$ can be simplified as follows:

$$\hat{C}\left(\mathbf{r}\right) = \sum_{i=1}^{N} T_i \cdot \left(1 - \exp\left(-\sigma_i \cdot \delta_i\right)\right) \cdot \mathbf{c}_i \tag{12}$$

where $\delta_i = t_{i+1} - t_i$ is the distance between two adjacent sampling points. Therefore, $T(t)$ can be expressed as follows:

$$T_i = \exp\left(-\sum_{j=1}^{i-1} \sigma_j \delta_j\right) \tag{13}$$

Based on this rendering method, the NeRF function can be used to render the image from any angle. After processing the dynamic environment, NeRF rendering is performed on the image frame. Firstly, the position needs to be passed to NeRF as input. By minimizing the reprojection error, BA is used for pose optimization. After inputting the camera position and key frame into NeRF, the photometric error is minimized by the regression of the image color. The ray can be determined by giving the key frame $(K, [R|t])$ and the pixel coordinate (u, v):

$$r\left(d\right) = \left(-R^T t + dR^T K^{-1}\left[u, v, 1\right]^T\right), d \in \mathbb{R} \tag{14}$$

Optical sampling is performed near the surface of the ray, but only selective sampling is performed in the static region. Finally, the NeRF luminosity loss function L is expressed as the L2 norm between the predicted color $\hat{C}(\mathbf{r}_{ij})$ and the pixel color $C(\mathbf{r})$:

$$L = \sum_{ij} \left\| C\left(\mathbf{r}_{ij}\right) - \hat{C}\left(\mathbf{r}_{ij}\right) \right\|_2 \tag{15}$$

When performing NeRF rendering, the mask obtained by segmenting the dynamic region is used to determine whether to render, and the image is generated by light sampling for each point in the scene to calculate the radiance of the light passing through the point. Because the dynamic part is beyond the sampling range, the dynamic part will appear as artifacts in the rendering process. When the dynamic object leaves the scene, the background occluded by the dynamic object in the previous rendering is repaired in the subsequent rendering to achieve the effect of background repair.

4. Experiment and results

In this section, a series of experiments are carried out to prove the effectiveness of RFN-SLAM. Tested on TUM RGB-D dataset [42], Bonn dataset [43] and self-collected dataset, RNF-SLAM is compared with ORB-SLAM3 and advanced SLAM systems in a dynamic environment. Absolute trajectory error (ATE) and relative attitude error (RPE) are used as indicators. Root Mean Square Error (RMSE) and Standard Deviation (S.D) were used to assess the standard deviation, which reflects the stability of the system, with smaller values indicating that the data is concentrated and stable. Considering that some processes in the algorithm have certain uncertainties, five experiments are performed on each sequence, and the median of each experiment is calculated. All experiments were run on a computing platform

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

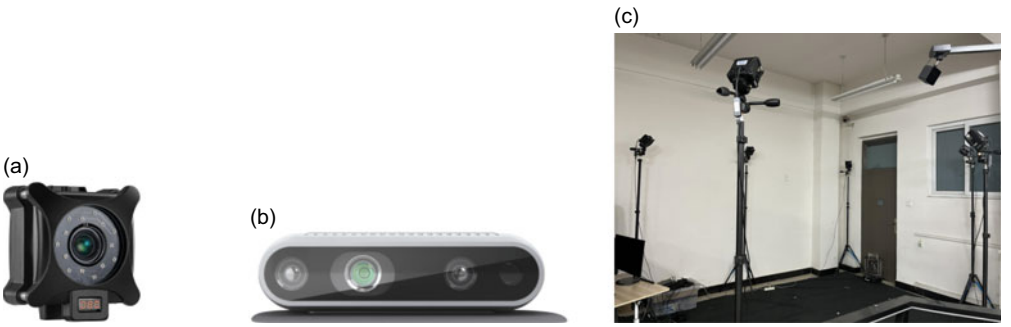


Figure 13. *Dataset acquisition platform. The D435i camera moves within the effective area, captures poses through the Mars1.3H lens, and outputs true motion trajectory values.*

with the Ubuntu 18.04 operating system, with hardware consisting of an Intel Xeon Silver 4214 CPU (2.20 GHz) and an NVIDIA GeForce RTX 3090 graphics card with 24 GB of video memory (CUDA version 11.8).

4.1. TUM dataset

The TUM RGB-D dataset contains many image sequences captured by RGB-D cameras in dynamic environments, as well as accurate ground-truth trajectories and camera internal parameters. Since the main purpose of dynamic SLAM is to improve the SLAM system in dynamic scenes, four groups of f3 sequences in high- and low-dynamic sequences were selected for each, f3-w/half, f3-w/rpy, f3-w/static, f3-w/xyz, f3-s/half, f3-s/rpy, f3-s/static, and f3-s/xyz totaling eight groups (the first four groups are high dynamic and the last four groups are low dynamic). Among them, w and s represent walking and sitting posture sequences, respectively, and half, rpy, static and xyz represent different motion modes of the camera, respectively.

4.2. Bonn dataset

The Bonn dataset uses the ASUS Xtion Pro LIVE camera to capture RGB-D images, including 24 dynamic sequences, such as crowd walking, balloon throwing, and moving boxes. Each scene can access the ground live camera trajectory. A comprehensive evaluation experiment was conducted on the performance of the algorithm proposed in this paper. Seven sequences were selected from the Bonn dataset for experiments, excluding sequences without static points in the entire view.

Among the selected sequences, the ‘balloon’ sequence captures the moment a person throws a balloon, while constantly hitting the balloon to prevent it from falling, while the ‘crowd’ sequence depicts a scene in which five people walk randomly in a room. The ‘person-tracking’ sequence captures the scene where the camera tracks the walking person. These complex scenarios further evaluate the performance of the RNF-SLAM system under various conditions.

4.3. Self-collected dataset

To address the limitations of publicly available datasets, this study constructed a real-scenario data acquisition platform, as shown in Figure 13. The platform was built based on the NOKOV optical system, featuring a multi-view infrared camera array that achieves submillimeter-level accuracy in spatial positioning and full coverage of dynamic areas. Three-dimensional reconstruction was performed using reflective markers to obtain six-degree-of-freedom pose ground truth. Multiple experimental groups were designed: the Static group for static scene acquisition (static); the Single-Person Motion group

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

containing linear, rotational, and compound movements (one_move); the Dynamic Load group to test algorithm stability under non-rigid loading conditions (move_box); and the Multi-Person Interaction groups to verify the algorithm's parsing capability in complex environments(two_move/three_move). These experiments provide reliable support for validating the robustness of the RFN-SLAM algorithm in dynamic environments.

4.4. Evaluating indicator

In the experiment, this paper uses ATE and RPE to measure the experimental results, and RMSE can be used for statistics. It is assumed that the estimated pose is expressed as: $P_1, \ldots, P_n \in SE(3)$, and the real pose is expressed as: $Q_1, \ldots, Q_n \in SE(3)$. ATE directly calculates the difference between the true value of the camera pose and the estimated value of the SLAM system, which can intuitively reflect the accuracy of the algorithm and the global consistency of the trajectory. The ATE of frame i is defined as follows:

$$F_i := Q_i^{-1} S P_i \tag{16}$$

where S is the similarity transformation matrix from the estimated pose to the real pose, and RMSE can be used to count ATE:

$$\mathrm{ATE}_{\mathrm{RMSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||F_i||^2} \tag{17}$$

RPE mainly describes the accuracy of the two-frame pose difference between the fixed time difference, which is equivalent to the error of the direct measurement odometer. Therefore, the RPE of frame i is defined as follows:

$$E_{i:} = \left(Q_i^{-1} Q_{i+\Delta}\right)^{-1} \left(P_i^{-1} P_{i+\Delta}\right) \tag{18}$$

where $\triangle$ is the interval time, RMSE is used to calculate RPE:

$$\mathrm{RPE}_{\mathrm{RMSE}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||E_i||^2} \tag{19}$$

The improvement of ORB-SLAM3 algorithm in this paper is expressed as:

$$Improvement = \frac{\mu - r}{\mu} \times 100\% \tag{20}$$

where μ represents RMSE or S.D of ORB-SLAM3; r represents RMSE or S.D of RFN-SLAM.

4.5. Compared with other advanced SLAM system experimental results

The RNF-SLAM system is compared with four advanced dynamic SLAM systems, including the original ORB-SLAM3, OVD-SLAM [32], LC-SLAM [33], MR-SLAM [34] and the proposed algorithm RFN-SLAM method. In each group of experiments, the RMSE and S.D of the ATE of each system are given to measure its global robustness and stability, and the RMSE and S.D of the RPE are used to measure the local performance.

4.5.1. Analysis of experimental results of TUM dataset

A comprehensive ATE quantitative comparison was performed on the experimental sequences selected in the TUM dataset, and the comparison indicators were RMSE and S.D. The proposed method achieves the best performance in most sequences, as shown in Table II. RFN-SLAM is usually superior to other methods in terms of RMSE of most sequences. For example, in the 'f3/w-xyz' sequence, there are two

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

Table II. *Results of metric ATE on TUM RGB-D dataset.*

	ORB-SLAM3	OVD-SLAM	LC-SLAM	MR-SLAM	RFN-SLAM	Improvements
Sequence	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**
f3/w-xyz	0.6231	0.0164	0.0232	0.0159	0.0144	97.6%
	(0.2978)	(0.0083)	(0.0121)	(0.0075)	(0.0066)	(97.8%)
f3/w-rpy	0.7355	0.0371	0.0481	0.0521	0.0365	95.0%
	(0.3966)	(0.0232)	(0.0291)	(0.0319)	(0.0225)	(94.3%)
f3/w-static	0.1123	0.0070	0.0143	0.0081	0.0079	92.9%
	(0.0518)	(0.0031)	(0.0073)	(0.0039)	(0.0035)	(93.2%)
f3/w-half	0.3123	0.3736	0.0483	0.0345	0.0256	91.8%
	(0.1471)	(0.1606)	(0.0211)	(0.0184)	(0.0133)	(90.9%)
f3/s-xyz	0.0095	0.0143	0.0113	0.0101	0.0108	–
	(0.0048)	(0.0083)	(0.0047)	(0.0053)	(0.0053)	
f3/s-rpy	0.0281	0.0568	0.0227	0.0365	0.0238	15.3%
	(0.0195)	(0.0473)	(0.0145)	(0.0289)	(0.0155)	(20.5%)
f3/s-static	0.0089	0.0070	0.0076	0.0083	0.0072	19.1%
	(0.0041)	(0.0027)	(0.0034)	(0.0036)	(0.0032)	(21.9%)
f3/s-half	0.0266	0.0406	0.0220	0.0142	0.0137	48.5%
	(0.0158)	(0.0296)	(0.0088)	(0.0062)	(0.0058)	(63.2%)

people walking back and forth as moving objects, and they occupy a large area in the current frame. At this moment, RFN-SLAM effectively determines the moving objects, but ORB-SLAM3 does not distinguish them. As a result, the RMSE of RFN-SLAM is the lowest (0.0144), while the RMSE of ORB-SLAM3 is the highest (0.6231). Compared with ORB-SLAM3, RMSE is significantly increased by 97.6%, and S.D is significantly increased by 97.8 %. In the sequences of 'fr3/w-rpy', 'fr3/w-xyz' and 'fr3/w-static', RFN-SLAM continues to show low RMSE, highlighting its excellent accuracy. When evaluating the S.D of the error, RFN-SLAM performs well, and usually the S.D value is the lowest or close to the lowest. For example, in the 'fr3/w-xyz' sequence, the SD of RFN-SLAM is 0.0066, which is significantly lower than that of ORB-SLAM3 (0.2978), which shows the high stability of its performance. In contrast, ORB-SLAM3 showed competitive RMSE in 'fr3/s-xyz'. In this sequence, the motion trend of the moving objects is not obvious, or even nearly stationary. The weighting approach adopted by RFN-SLAM appears to be over-corrected, resulting in inferior performance to ORB-SLAM3. However, in 'fr3/w-rpy', ORB-SLAM3 has the highest RMSE. This indicates that its performance fluctuates greatly in different scenarios. The performance of OVD-SLAM, LC-SLAM and MR-SLAM is relatively balanced. RMSE and S.D are neither the lowest nor the highest, which means that they are consistent in accuracy, but not necessarily optimal. In general, these data strongly support the conclusion that the proposed RFN-SLAM method exhibits better accuracy and stability in the test sequence.

Table III quantitatively analyzes the RPE of various SLAM methods and evaluates the accuracy and stability through RMSE and S.D. Comparing RMSE, the performance differences between different sequences and methods were observed. RFN-SLAM shows competitive or lower RMSE in most cases, indicating its robustness and effectiveness. The S.D value further illustrates the consistency of each method. For example, in 'fr3/w-static', the S.D of RFN-SLAM is 0.0035, showing its consistency. In contrast, ORB-SLAM3 has a higher S.D value (0.0189) in 'fr3/w-static', indicating that its performance is more variable. Compared with ORB-SLAM3, the S.D of RFN-SLAM in this sequence was significantly increased by 81.4%. Overall, the experimental data quantitatively confirm the advantages and disadvantages of each SLAM method in different environments. However, the RFN-SLAM method shows significant advantages, and it can balance accuracy and consistency in most cases.

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

Table III. *Results of metric RPE on TUM RGB-D dataset.*

	ORB-SLAM3	OVD-SLAM	LC-SLAM	MR-SLAM	RFN-SLAM	Improvements
Sequence	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**
f3/w-xyz	0.0253	0.0137	0.0146	0.0122	0.0117	53.7%
	(0.0147)	(0.0081)	(0.0085)	(0.0069)	(0.0060)	(59.2%)
f3/w-rpy	0.0302	0.0213	0.0236	0.0253	0.0249	17.5%
	(0.0191)	(0.0135)	(0.0146)	(0.0181)	(0.0180)	(5.7%)
f3/w-static	0.0220	0.0058	0.0098	0.0067	0.0067	69.5%
	(0.0189)	(0.0028)	(0.0057)	(0.0038)	(0.0035)	(81.4%)
f3/w-half	0.0218	0.0329	0.0155	0.0159	0.0135	38.1%
	(0.0143)	(0.0304)	(0.0086)	(0.0099)	(0.0077)	(46.1%)
f3/s-xyz	0.0099	0.0124	0.0089	0.0093	0.0083	16.2%
	(0.0052)	(0.0082)	(0.0046)	(0.0050)	(0.0042)	(19.2%)
f3/s-rpy	0.0172	0.0239	0.0132	0.0222	0.0145	15.7%
	(0.0117)	(0.0181)	(0.0087)	(0.0190)	(0.0102)	(12.8%)
f3/s-static	0.0063	0.0053	0.0057	0.0053	0.0053	15.9%
	(0.0035)	(0.0026)	(0.0029)	(0.0027)	(0.0025)	(28.5%)
f3/s-half	0.0112	0.0192	0.0106	0.0104	0.0080	28.6%
	(0.0060)	(0.0138)	(0.0062)	(0.0058)	(0.0045)	(25.0%)

Figure 14 shows the error between the estimated trajectory of the ORB-SLAM3 and RFN-SLAM algorithms and the actual trajectory on the ground. Through observation, the original ORB-SLAM3 has obvious errors in the high-dynamic sequence, which is inconsistent with the actual ground conditions. In contrast, the RFN-SLAM algorithm can calculate the estimated trajectory that closely matches the original trajectory, and achieve more accurate positioning in a dynamic environment. Figure 15 shows the operation in the sequence 'fr3/w-static'. By comparing ORB-SLAM3, RFN-SLAM shows significant advantages, which can effectively eliminate the feature points related to dynamic objects, so as to ensure the stable operation of the system.

NeRF rendering is performed on the sequence of the selected experiment on the TUM dataset, such as the effect of rendering the 'fr3/w-xyz' sequence, as shown in Figure 16. Because the proposed algorithm RFN-SLAM is only adopted in the static area, when the dynamic object enters the scene, it will not be adopted. When it is separated from the scene, NeRF will correct the lost part of the previous rendering (i.e., the background occluded by the dynamic object), so as to achieve the effect of background repair, and the map of the static scene will be obtained. RFN-SLAM reduces the influence of dynamic objects on the system at the front end, makes the camera pose estimation more accurate, and provides a more accurate camera position for NeRF rendering to achieve better rendering results in a dynamic environment.

4.5.2. Analysis of experimental results of Bonn dataset

The selected sequences were compared in the Bonn dataset, and a comprehensive ATE quantitative comparison was performed, as shown in Table IV. In crowded scenes (such as 'crowd', 'crowd2' and 'crowd3'), the motion estimation deviation of ORB-SLAM3 is obvious and deviates from the actual trajectory. In the 'crowd' sequence, personnel movement is large and densely distributed, resulting in fewer static feature points to be extracted. RFN-SLAM is able to show better performance advantages by removing dynamic feature points and using static weighting. RFN-SLAM has the lowest RMSE (0.0186), while ORB-SLAM3 has the highest RMSE (0.6001). Compared with ORB-SLAM3, the RMSE of RFN-SLAM is increased by 96.9 %. In other sequences, RFN-SLAM continues to show low RMSE, highlighting its excellent accuracy. RFN-SLAM also performs well when evaluating the

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

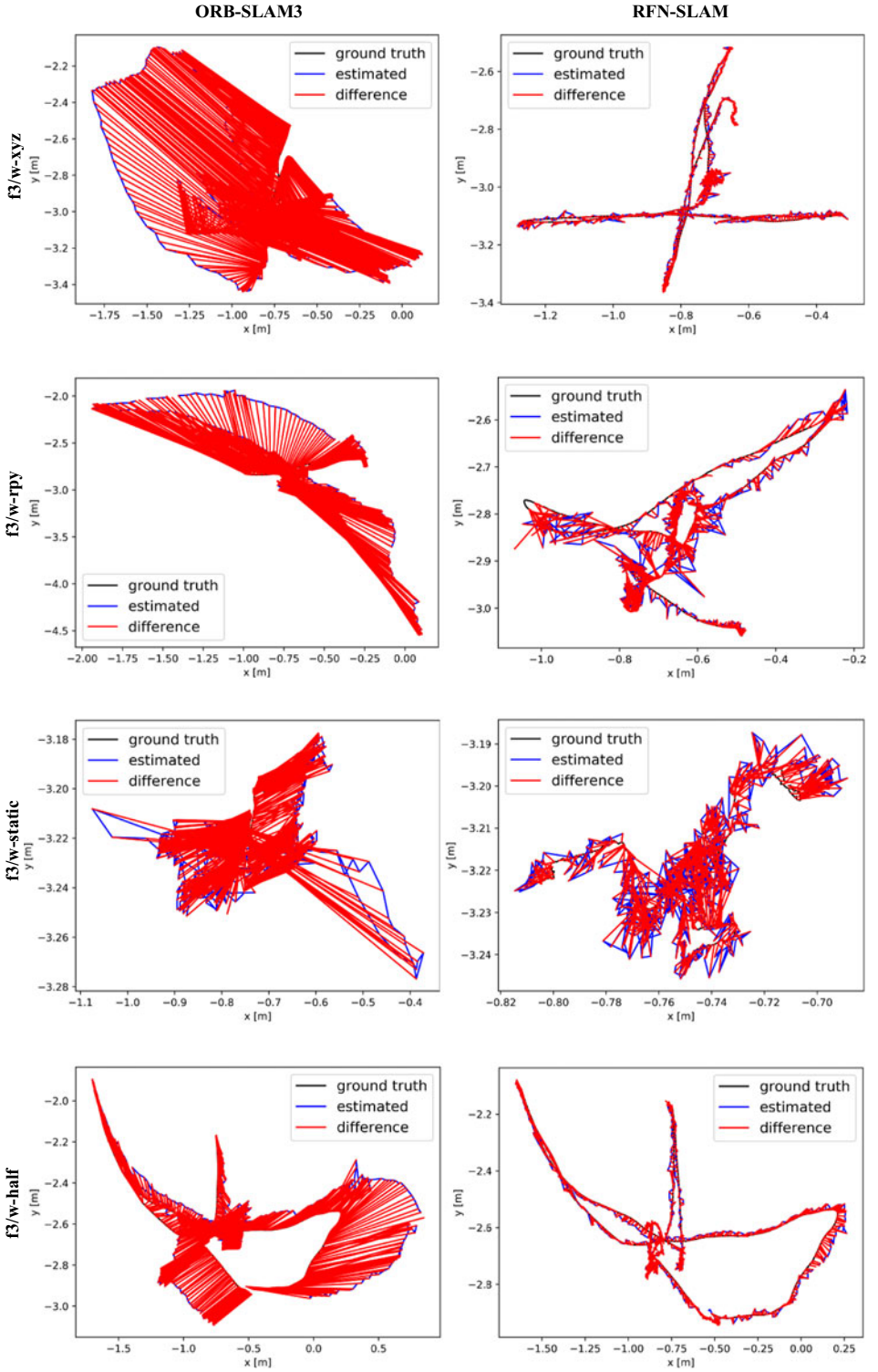


Figure 14. *In the high-dynamic scene of the TUM RGB-D dataset, the ATE of ORB-SLAM3 and RFN-SLAM is compared. The black line represents the real trajectory of the camera motion, the blue line represents the estimated trajectory, and the red line represents the difference between the estimated trajectory and the real trajectory.*

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press



Figure 15. *The feature point extraction results of ORB-SLAM3 and RFN-SLAM in 'fr3/w-static' sequence are compared. RFN-SLAM can effectively detect and eliminate the feature points of dynamic objects.*

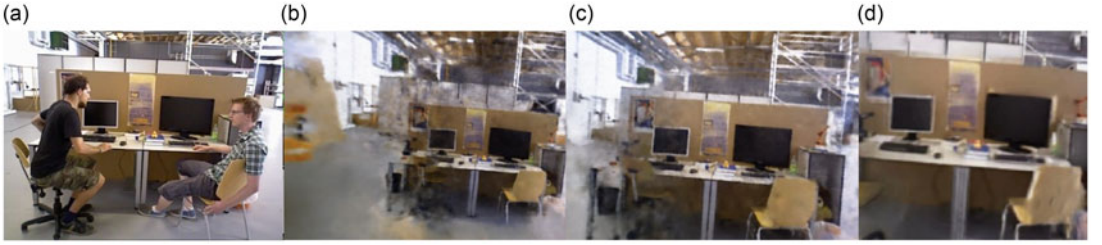


Figure 16. *The results of 3D reconstruction of the 'fr3/w-xyz' sequence of the TUM dataset using RFN-SLAM only reconstruct the static part. (a) input; (b), (c) and (d) reconstruction.*

S.D of the error. For example, in the 'person-tracking2' sequence, the SD of RFN-SLAM is 0.0133, which is significantly lower than that of ORB-SLAM3 (0.4626), fully demonstrating the high stability of its performance. During the experiment, it is found that other advanced algorithms sometimes lose cameras in these sequences. The reason is that dense crowds are the key characteristics of the scene, and the movement of people may occupy a large range. After the removal of dynamic feature points, the remaining static feature points are not enough to support accurate camera pose matching and estimation, and the RFN-SLAM system proposed in this paper can better handle similar situations. Through experiments, it is fully proven that the RFN-SLAM method has significant advantages in accuracy, stability and robustness in complex scenes containing many dynamic objects.

Table V quantitatively analyzes the RPE of various SLAM methods on the Bonn dataset. Through chart data analysis, RFN-SLAM shows lower RMSE in most experimental scenarios. For example, in the sequence 'crowd', the RMSE (0.0222) of RFN-SLAM can obtain better advantages than other SLAM algorithms, indicating its robustness and effectiveness. At the same time, in 'crowd', the S.D of RFN-SLAM is 0.0187, showing its consistency.

Figure 17 clearly shows the gap between the estimated trajectory and the real trajectory of ORB-SLAM3 and RFN-SLAM algorithms in dynamic scenes. The estimation deviation of ORB-SLAM3 is obvious, which is mainly limited by the influence of dynamic objects. In contrast, the RFN-SLAM algorithm can effectively deal with dynamic objects, thus providing more accurate positioning and higher accuracy. Figure 18 shows the actual operation in the sequence crowd2. By comparing ORB-SLAM3, RFN-SLAM shows significant advantages, which can effectively eliminate the feature points related to dynamic objects, so as to ensure the stable operation of the system.

Figure 19 shows the dense reconstruction of RFN-SLAM algorithm under dense human activities (such as 'crowd2' sequence). The figure shows that the crowd is effectively excluded and focuses on the reconstruction of static scenes. By removing dynamic objects, the accuracy of pose estimation is

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

Table IV. *Results of metric ATE on Bonn dataset.*

Sequence	ORB-SLAM3 RMSE (S.D)	OVD-SLAM RMSE (S.D)	LC-SLAM RMSE (S.D)	MR-SLAM RMSE (S.D)	RFN-SLAM RMSE (S.D)	Improvements RMSE (S.D)
balloon	0.1564 (0.0653)	0.0428 (0.0206)	0.0300 (0.0122)	0.0271 (0.0114)	0.0261 (0.0129)	83.3% (80.2%)
balloon2	0.2406 (0.1093)	0.0310 (0.0145)	0.0312 (0.0120)	0.0273 (0.0129)	0.0237 (0.0096)	90.1% (91.2%)
crowd	0.6001 (0.3412)	0.0416 (0.0348)	0.0302 (0.0147)	0.0344 (0.0238)	0.0186 (0.0099)	96.9% (97.1%)
crowd2	0.6514 (0.2530)	0.0262 (0.0147)	0.0292 (0.0163)	0.0687 (0.0560)	0.0220 (0.0114)	96.6% (95.4%)
crowd3	0.9501 (0.4180)	0.0213 (0.0113)	0.0327 (0.0210)	0.0642 (0.0581)	0.0241 (0.0139)	97.5% (96.6%)
person-tracking	0.7194 (0.3433)	0.0537 (0.0144)	0.0423 (0.0118)	0.0470 (0.0150)	0.0354 (0.0140)	95.1% (95.9%)
person-tracking2	0.7761 (0.4626)	0.0585 (0.0158)	0.0426 (0.0137)	0.1492 (0.0474)	0.0341 (0.0133)	95.6% (97.1%)

Table V. *Results of metric RPE on Bonn dataset.*

Sequence	ORB-SLAM3 RMSE (S.D)	OVD-SLAM RMSE (S.D)	LC-SLAM RMSE (S.D)	MR-SLAM RMSE (S.D)	RFN-SLAM RMSE (S.D)	Improvements RMSE (S.D)
balloon	0.0542 (0.0494)	0.0430 (0.0370)	0.0405 (0.0360)	0.0448 (0.0386)	0.0405 (0.0354)	25.3% (28.3%)
balloon2	0.0646 (0.0592)	0.0384 (0.0292)	0.0347 (0.0284)	0.0360 (0.0289)	0.0344 (0.0277)	46.7% (53.2%)
crowd	0.1573 (0.1506)	0.0245 (0.0211)	0.0253 (0.0211)	0.0288 (0.0249)	0.0222 (0.0187)	85.8% ((87.6%)
crowd2	0.1099 (0.1006)	0.0277 (0.0230)	0.0314 (0.0263)	0.0474 (0.0403)	0.0291 (0.0245)	73.5% (75.6%)
crowd3	0.0595 (0.0537)	0.0209 (0.0159)	0.0246 (0.0196)	0.0594 (0.0552)	0.0246 (0.0195)	58.6% (63.7%)
person-tracking	0.1417 (0.1374)	0.0752 (0.0714)	0.0742 (0.0709)	0.0751 (0.0711)	0.0766 (0.0725)	45.9% (47.2%)
person-tracking2	0.1012 (0.0950)	0.0632 (0.0593)	0.0635 (0.0597)	0.0630 (0.0583)	0.0649 (0.0610)	35.9% (35.8%)

improved, which significantly improves the quality of reconstructions with tables, figures may be presented at appropriate locations within the text. Figure captions should be below the figure itself, as shown below.

4.5.3. Analysis of experimental results of self-collected dataset

The data in Table VI reveals that RFN-SLAM demonstrates all-round performance enhancement in static environments, dynamic scenarios and load handling scenarios. In the static sequence, RFN-SLAM achieves RMSE = 0.0139, which is 84.5% improvement over ORB-SLAM3, and its S.D. reduction reaches 75.2%, which verifies the optimization effect of the base positioning accuracy. As the

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

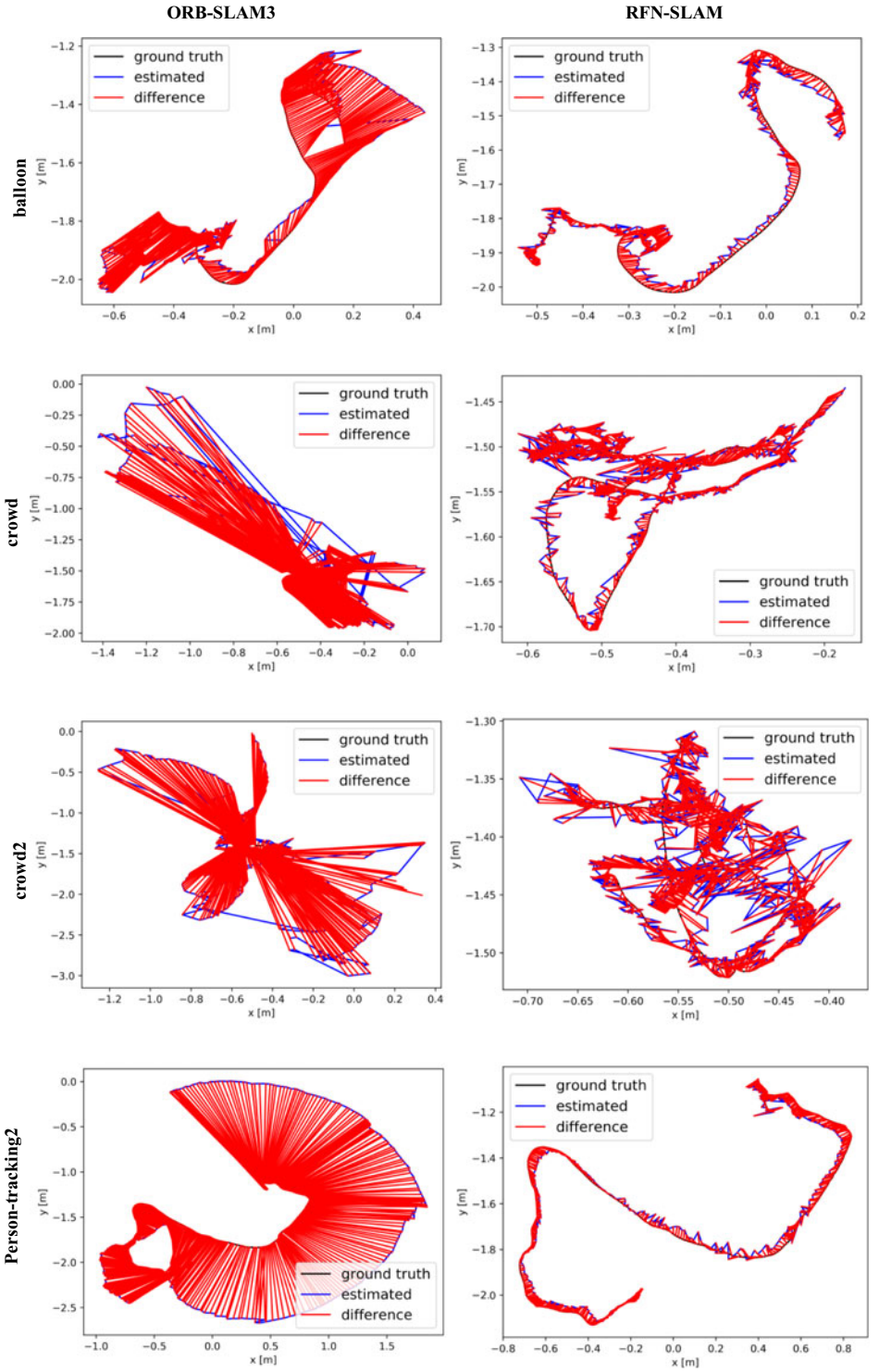


Figure 17. *Comparison of the ATE of ORB-SLAM3 and RFN-SLAM in some dynamic scenes of the Bonn dataset. The black line represents the real trajectory of the camera motion, the blue line represents the estimated trajectory, and the red line represents the difference between the estimated trajectory and the real trajectory.*

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

Table VI. *Results of metric APE on self-collected dataset.*

Sequence	ORB-SLAM3 RMSE (S.D)	OVD-SLAM RMSE (S.D)	LC-SLAM RMSE (S.D)	MR-SLAM RMSE (S.D)	RFN-SLAM RMSE (S.D)	Improvements RMSE (S.D)
static	0.0898	0.0114	0.0176	0.0145	0.0139	84.5%
	(0.0318)	(0.0050)	(0.0076)	(0.0082)	(0.0079)	(75.2%)
one_move	0.1085	0.0360	0.0414	0.0699	0.0178	83.6%
	(0.0366)	(0.0307)	(0.0195)	(0.0610)	(0.0142)	(61.2%)
two_move	0.1068	0.0139	0.0267	0.0133	0.0128	88.0%
	(0.0393)	(0.0073)	(0.0119)	(0.0069)	(0.0061)	(84.5%)
three_move	0.1050	0.0146	0.1641	0.0146	0.0137	87.0%
	(0.0386)	(0.0072)	(0.1154)	(0.0059)	(0.0057)	(85.2%)
move_box	0.0856	0.0099	0.0129	0.0119	0.0083	90.3%
	(0.0312)	(0.0042)	(0.0052)	(0.0610)	(0.0034)	(98.1%)



Figure 18. *The feature point extraction results of ORB-SLAM3 and RFN-SLAM in the 'crowd2' sequence are compared. RFN-SLAM can effectively detect and eliminate the feature points of dynamic objects.*



Figure 19. *The results of 3D reconstruction of the 'crowd2' sequence of the Bonn dataset using RFN-SLAM only reconstruct the static part. (a) input; (b) and (c) reconstruction.*

number of dynamic interference sources increases (two_move to three_move), RFN-SLAM maintains RMSE<0.0137 (S.D.<0.0061), which maintains more than 87.0% error reduction over the baseline algorithm. Particularly in the sequence move_box, RFN-SLAM achieves an RMSE of 0.0083 (IMP 90.3%) with an S.D. of 0.0034, which is 94.4% lower than MR-SLAM. Although OVD-SLAM has a slightly better RMSE in some scenarios (e.g., static), RFN-SLAM maintains the lead in terms of overall dynamic performance.

The RPE data in Table VII further reveals the algorithm characteristics. In the sequence one_move, RFN-SLAM has an RMSE of 0.0146, which is a 38.7% improvement and a 45.0% reduction in S.D.

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

Table VII. *Results of metric RPE on self-collected dataset.*

	ORB-SLAM3	OVD-SLAM	LC-SLAM	MR-SLAM	RFN-SLAM	Improvements
Sequence	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**	**RMSE (S.D)**
static	0.0063 (0.0029)	0.0085 (0.0041)	0.0102 (0.0057)	0.0086 (0.0043)	0.0092 (0.0051)	–
one_move	0.0238 (0.0231)	0.0149 (0.0128)	0.0150 (0.0124)	0.0187 (0.0171)	0.0146 (0.0127)	38.7% (45.0%)
two_move	0.0046 (0.0022)	0.0081 (0.0043)	0.0091 (0.0057)	0.0074 (0.0042)	0.0077 (0.0044)	–
three_move	0.0054 (0.0028)	0.0094 (0.0055)	0.0102 (0.0165)	0.0081 (0.0044)	0.0079 (0.0042)	–
move_box	0.0073 (0.0061)	0.0074 (0.0038)	0.0072 (0.0040)	0.0097 (0.0076)	0.0060 (0.0029)	17.8% (52.5%)

compared to ORB-SLAM3 (0.0238), reflecting its ability to suppress localized motion disturbances. In the sequence move_box, its RMSE is further optimized to 0.0060 (IMP 17.8%) and the S.D. is only 0.0029, which is 52.5% lower than that of ORB-SLAM3 (0.0061). Although its RMSE has a slight disadvantage of 4.3% to 7.8% in the sequences two_move and three_move, its control of error fluctuation is more robust. In addition, RFN-SLAM has an RMSE of 0.0092 in sequence static, which is slightly inferior to ORB-SLAM3 (0.0063), but its S.D. value (0.0051) is close to that of OVD-SLAM (0.0041), and it is still better than LC-SLAM (0.0057). This result shows that RFN-SLAM significantly improves the localization robustness in dynamic environments while ensuring the basic performance in static scenes.

Figure 20 shows the trajectory comparison between ORB-SLAM3 and RFN-SLAM on the self-picked dataset. Through the comparison, the obvious advantages of RFN-SLAM in terms of trajectory accuracy and stability can be visualized. Especially in the dynamic and complex scenario of sequence move_box, RFN-SLAM effectively suppresses the accumulation of errors caused by environmental changes. The dynamic feature point processing mechanism of RFN-SLAM successfully avoids the expansion of errors, showing its excellent dynamic adaptability and robustness. Compared with the ORB-SLAM algorithm, RFN-SLAM is able to maintain high trajectory accuracy when facing complex dynamic scenes, which proves its strong stability and reliability in dynamic environments. This performance validates the superiority of RFN-SLAM in dealing with complex environmental changes, especially its potential to improve system performance and accuracy.

Figure 21 presents the distribution characteristics of dynamic object feature points between ORB-SLAM3 and RFN-SLAM in the real scene sequence ‘one_move’. The experimental results show that RFN-SLAM accurately filters out the feature points on the surface of a dynamic object in a multi-frame sequential image sequence through the dynamic feature discrimination mechanism, thus providing stable and reliable feature information to support the camera pose estimation. In contrast, ORB-SLAM3 lacks the ability to discriminate between dynamic and static features, and its feature point set contains a large amount of dynamic object information. This comparative experimental result further validates the superiority of RFN-SLAM in dynamic scenarios. Figure 22 shows the reconstruction results of RFN-SLAM for the ‘move_box’ sequence without dynamic object dense map. As can be seen from the figure, the algorithm not only successfully removes the moving objects in the scene, but also performs a complete geometric reconstruction of the background region that is occluded by them.

4.5.4. Time performance analysis

In practical engineering applications, the real-time performance of an algorithm is a key indicator of its applicability. In this study, the temporal performance of RFN-SLAM is verified by comparing its

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

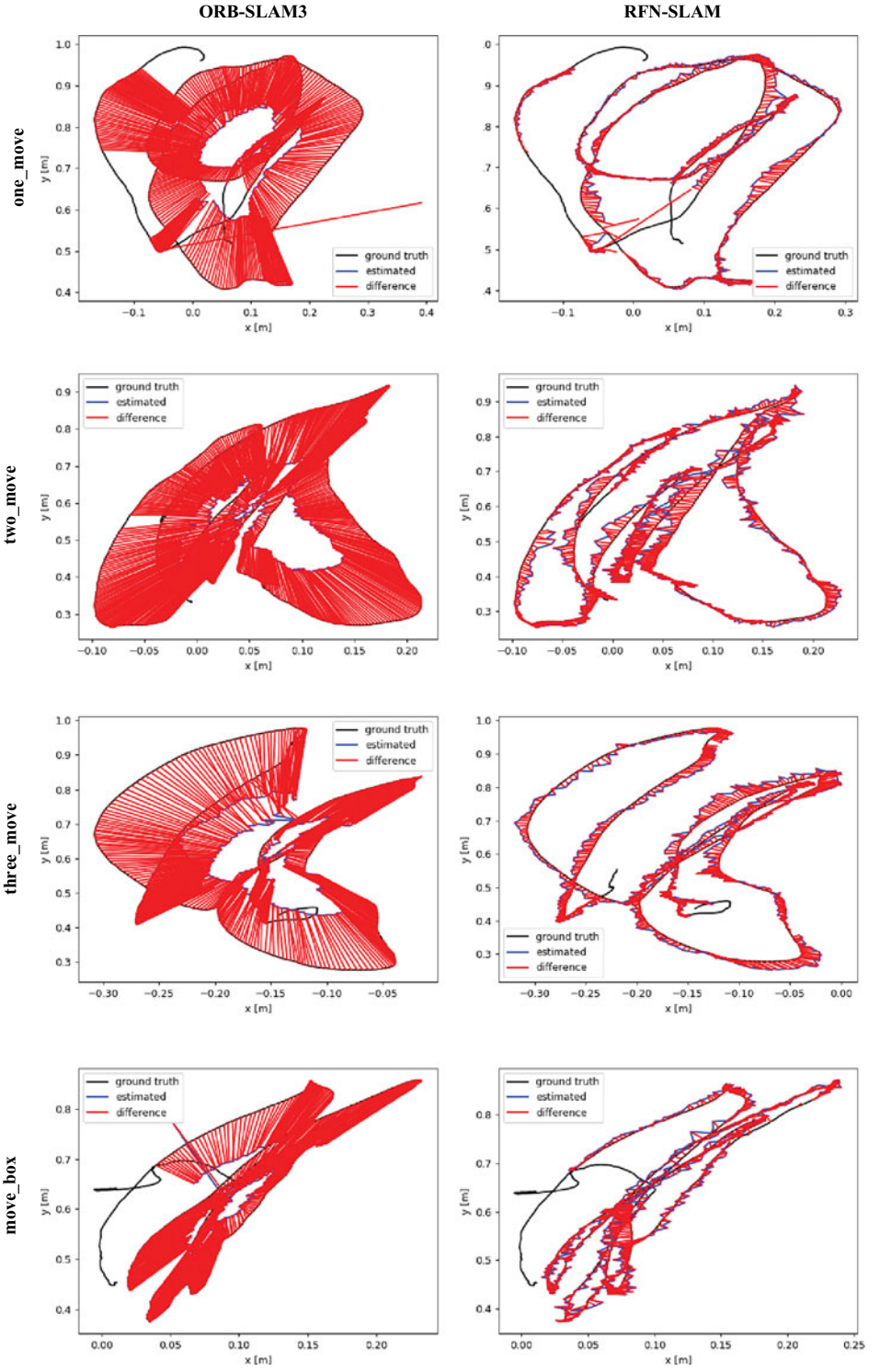


Figure 20. *Comparison of the ATE of ORB-SLAM3 and RFN-SLAM in all dynamic scenes of the self-collected dataset. The black line represents the real trajectory of the camera motion, the blue line represents the estimated trajectory, and the red line represents the difference between the estimated trajectory and the real trajectory.*

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

Table VIII. *Real-time comparison of VSLAM algorithms in dynamic environments.*

VSLAM algorithm	ORB-SLAM3	OVD-SLAM	LC-SLAM	MR-SLAM	RFN-SLAM
Average tracking time/ms	30.75	21.48	77.82	88.72	84.20

Table IX. *Comparison of time consumption of each module of RFN-SLAM algorithm.*

Module	RT-DETR thread	FAST-SAM thread	Tracking thread
Time consumption/ms	5.56	56.74	65.39



Figure 21. *The feature point extraction results of ORB-SLAM3 and RFN-SLAM in the 'one_move' sequence are compared. RFN-SLAM can effectively detect and eliminate the feature points of dynamic objects.*

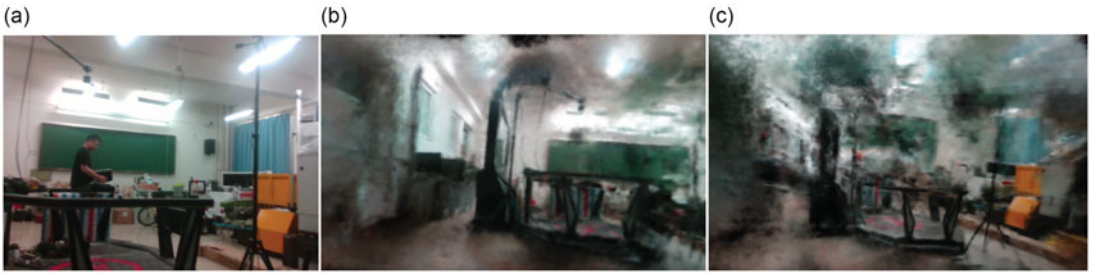


Figure 22. *The results of 3D reconstruction of the 'move_box' sequence of the self-collected dataset using RFN-SLAM only reconstruct the static part. (a) input; (b) and (c) reconstruction.*

computational efficiency with that of existing dynamic visual SLAM algorithms on TUM and Bonn standard datasets. As shown in Table VIII, the average processing time of the RFN-SLAM system for a single-frame image is 84.2 $\pm$ 1.5 ms, which is able to meet the requirements of mobile robot platforms. Compared with the benchmark algorithm ORB-SLAM3, RFN-SLAM keeps the computational elapsed time within 2.8 times of that of ORB-SLAM3 while maintaining a higher localization accuracy, demonstrating the Pareto-optimized characteristics of accuracy and efficiency.

Table IX presents the average computational time distribution of the RFN-SLAM algorithm for each functional module, covering the RT-DETR thread, the FAST-SAM thread, and the Tracking thread as the main thread. Among them, the Tracking thread, as the core control module of the system, takes significantly longer than the two sub-threads, which is determined by the complexity of the main tasks such as global pose estimation and map maintenance. In the FAST-SAM thread, the computational overhead of this module is relatively high due to the two-phase segmentation process involved; while the RT-DETR thread significantly reduces the computational time consumption through algorithm optimization, and demonstrates efficient target detection capability. From the perspective of overall system performance, the comprehensive computation time of RFN-SLAM can meet the real-time requirements of most

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

real-world scenarios, which indicates that the algorithm has good potential for engineering applications while guaranteeing the functional integrity of the algorithm.

5. Conclusion

In this paper, an RFN-SLAM algorithm for dynamic scenes without preset dynamic labels and weighted features is proposed, which combines object detection and instance segmentation networks. The algorithm aims to identify and segment dynamic objects in the environment, including known and limited unknown dynamic objects. In order to improve the matching accuracy, the dynamic feature points, especially the dynamic parts and edges, are eliminated by the mask of the dynamic object. Meanwhile, the enhanced efficient binary BAD image descriptor is used, and the static feature points are weighted according to the dynamic information to optimize the pose, so as to improve the accuracy of camera pose estimation and improve the performance of SLAM system in dynamic environment. In addition, using the fast NeRF framework, only static points are retained to generate dense maps without pre-training, while processing dynamic objects and occluded backgrounds. The experimental results show that compared with the existing advanced SLAM technology, RFN-SLAM shows excellent accuracy, stability and robustness in the indoor scenes of TUM RGB-D dataset and Bonn dataset, which confirms the effectiveness of RFN-SLAM algorithm.

All experiments in this paper are carried out in the indoor environment, and the future plan is to extend to the outdoor scene for algorithm verification. With regard to instance partition networks, we recognize that it is necessary to pay attention to the improvement of real-time performance in future research.

Author contributions. Shuai Xiang: Data curation, writing original draft, software, and validation. Chaoyi Dong: Conceptualization, methodology, and supervision. Kang Zhang: Data curation and supervision. Ge Tai: Software. Tianyu Yuan: Validation. Haoda Yan: Validation. Xiaoyan Chen: Software and validation.

Financial support. This work was supported by the National Natural Science Foundation of China (61863029), Inner Mongolia Scientific and Technological Achievements Transformation Project (CGZH2018129), Industrial Technology Innovation Program of IMAST (2023JSYD01006), Basic Research Funds Project for Directly Affiliated Universities in the Autonomous Region (JY20240076 and RZ2400003491), The Ministry of Industry and Information Technology's Future Industrial Innovation Task List and Leading Project, and Science and Technology Plan Project of Inner Mongolia Autonomous Region (2020GG0268 and 2025YFHH0046).

Competing interests. The authors declare no conflicts of interest.

Ethical approval. None.

References

[1] X. Wang, X. Fan, P. Shi, J. Ni and Z. Zhou, "An overview of key SLAM technologies for underwater scenes," *Remote Sens-BASEL* **15**(10), 2496 (2023).
[2] D. Sharafutdinov, M. Griguletskii, P. Kopanev, M. Kurenkov, G. Ferrer, A. Burkov, A. Gonnochenko and D. Tsetserukou, "Comparison of modern open-source visual SLAM approaches," *J. Intell. Rob. Syst.* **107**(3), 43 (2023).
[3] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," *2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, 13 Nov 2007* (IEEE, 2007) pp. 225–234.
[4] J. Engel, T. Schöps and D. Cremers, "LSD-SLAM: Large-Scale Direct Monocular SLAM," *European Conference on Computer Vision, 6 Sep 2014* (Springer International Publishing, Cham, 2014) pp. 834–849.
[5] Y. Sun, M. Liu and M. Q. Meng, "Improving RGB-D SLAM in dynamic environments: A motion removal approach," *Rob. Auton. Syst.* **1**(89), 110–122 (2017).
[6] R. Mur-Artal, J. M. Montiel and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Rob.* **31**(5), 1147–1163 (2015).
[7] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Trans. Rob.* **33**(5), 1255–1262 (2017).

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

[8] C. Campos, R. Elvira, J. J. Rodríguez, J. M. Montiel and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Trans. Rob.* **37**(6), 1874–1890 (2021).

[9] H. Zhang, F. Ye, Y. Lai, K. Li and J. Xu, "IQ-VIO: Adaptive visual inertial odometry via interference quantization under dynamic environments," *Intell. Serv. Rob.* **16**(5), 565–581 (2023).

[10] X. Zhao, T. Zuo and X. Hu, "OFM-SLAM: A visual semantic SLAM for dynamic indoor environments," *Math. Probl. Eng.* **2021**(1), 5538840 (2021).

[11] M. Gonzalez, E. Marchand, A. Kacete and J. Royan, "Twistslam: Constrained slam in dynamic environment," *IEEE Rob. Autom. Lett.* **7**(3), 6846–6853 (2022).

[12] B. Bescos, J. M. Fácil, J. Civera and J. Neira, "DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes," *IEEE Rob. Autom. Lett.* **3**(4), 4076–4083 (2018).

[13] C. Yu, Z. Liu, X. J. Liu, F. Xie, Y. Yang, Q. Wei and Q. Fei, "DS-SLAM: A Semantic Visual SLAM Towards Dynamic Environments," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 1 Oct 2018* (IEEE, 2018) pp. 1168–1174.

[14] Z. Pan, J. Hou and L. Yu, "Optimization RGB-D 3-D reconstruction algorithm based on dynamic SLAM," *IEEE Trans. Instrum. Meas.* **23**(72), 1–3 (2023).

[15] S. Cheng, 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 Trans. Instrum. Meas.* **72**, 1–2 (2022).

[16] T. Müller, A. Evans, C. Schied and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graphics (TOG)* **41**(4), 1–15 (2022).

[17] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Commun. Acm* **65**(1), 99–106 (2021).

[18] Y. Zhao, W. Lv, S. Xu, J. Wei, G. Wang, Q. Dang, Y. Liu and J. Chen, "Detrs Beat Yolos on Real-Time Object Detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024) pp. 16965–16974.

[19] X. Zhao, W. Ding, Y. An, Y. Du, T. Yu, M. Li, M. Tang and J. Wang, "Fast segment anything. arxiv preprint arxiv:2306.12156 (2023).

[20] I. Suárez, J. M. Buenaposada and L. Baumela, "Revisiting binary local image description for resource limited devices," *IEEE Rob. Autom. Lett.* **6**(4), 8317–8324 (2021).

[21] D. H. Kim and J. H. Kim, "Effective background model-based RGB-D dense visual odometry in a dynamic environment," *IEEE Trans. Rob.* **32**(6), 1565–1573 (2016).

[22] R. Wang, W. Wan, Y. Wang and K. Di, "A new RGB-D SLAM method with moving object detection for dynamic indoor scenes," *Remote Sens-BASEL* **11**(10), 1143 (2019).

[23] B. Song, X. Yuan, Z. Ying, B. Yang, Y. Song and F. Zhou, "DGM-VINS: Visual–inertial SLAM for complex dynamic environments with joint geometry feature extraction and multiple object tracking," *IEEE Trans. Instrum. Meas.* **72**, 1–1 (2023).

[24] W. Dai, Y. Zhang, P. Li, Z. Fang and S. Scherer, "Rgb-d slam in dynamic environments using point correlations," *IEEE Trans. Pattern Anal.* **44**(1), 373–389 (2020).

[25] J. C. Soares, M. Gattass and M. A. Meggiolaro, "Crowd-SLAM: Visual SLAM towards crowded environments using object detection," *J. Intell. Rob. Syst.* **102**(2), 50 (2021).

[26] T. Ji, C. Wang and L. Xie, "Towards Real-Time Semantic RGB-D SLAM in Dynamic Environments," *2021 IEEE International Conference on Robotics and Automation (ICRA), 30 May 2021* (IEEE) pp. 11175–11181.

[27] K. Zhang, C. Dong, H. Guo, Q. Ye, L. Gao, S. Xiang, X. Chen and Y. Wu, "A semantic visual SLAM based on improved mask R-CNN in dynamic environment," *Robotica* **42**(10), 3570–3591 (2024).

[28] X. Yu, W. Zheng and L. Ou, "CPR-SLAM: RGB-D SLAM in dynamic environment using sub-point cloud correlations," *Robotica* **42**(7), 2367–2387 (2024).

[29] L. Kenye and R. Kala, "Improving RGB-D SLAM in dynamic environments using semantic aided segmentation," *Robotica* **40**(6), 2065–2090 (2022).

[30] S. Li, J. Gu, Z. Li, S. Li, B. Guo, S. Gao, F. Zhao, Y. Yang, G. Li and L. Dong, "A visual SLAM-based lightweight multi-modal semantic framework for an intelligent substation robot," *Robotica* **42**(7), 2169–2183 (2024).

[31] Y. Fan, Q. Zhang, Y. Tang, S. Liu and H. Han, "Blitz-SLAM: A semantic SLAM in dynamic environments," *Pattern Recognit.* **121**, 108225 (2022).

[32] J. He, M. Li, Y. Wang and H. Wang, "OVD-SLAM: An online visual SLAM for dynamic environments," *IEEE Sens. J.* **23**(12), 13210–13219 (2023).

[33] Z. J. Du, S. S. Huang, T. J. Mu, Q. Zhao, R. R. Martin and K. Xu, "Accurate dynamic SLAM using CRF-based long-term consistency," *IEEE Trans. Visual. Comput. Graphics* **28**(4), 1745–1757 (2020).

[34] Q. Zhang and C. Li, "Semantic SLAM for mobile robots in dynamic environments based on visual camera sensors," *Meas. Sci. Technol.* **34**(8), 085202 (2023).

[35] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Rob. Autom. Lett.* **2**(4), 2263–2270 (2017).

[36] Y. Zhong, S. Hu, G. Huang, L. Bai and Q. Li, "WF-SLAM: A robust VSLAM for dynamic scenarios via weighted features," *IEEE Sens. J.* **22**(11), 10818–10827 (2022).

[37] J. Zhang, M. Henein, R. Mahony and V. Ila, "VDO-SLAM: A visual dynamic object-aware SLAM system. " arxiv preprint arxiv:2005.11052 (2020).

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press

[38] Y. Yao, Z. Luo, S. Li, T. Fang and L. Quan, "Mvsnet: Depth Inference for Unstructured Multi-view Stereo," *Proceedings of the European Conference on Computer Vision (ECCV)* (2018) pp. 767–783.
[39] Y. Yao, Z. Luo, S. Li, T. Shen, T. Fang and L. Quan, "Recurrent MVSNet for High-Resolution Multi-view Stereo Depth Inference," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019) pp. 5525–5534.
[40] B. Kerbl, G. Kopanas, T. Leimkühler and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph* **42**(4), 139–131 (2023).
[41] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu and Y. Yuan, "Efficientvit: Memory Efficient Vision Transformer with Cascaded Group Attention," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023) pp. 14420–14430.
[42] J. Sturm, N. Engelhard, F. Endres, W. Burgard and D. Cremers, "A Benchmark for the Evaluation of RGB-D SLAM Systems," *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2012) pp. 573–580.
[43] E. Palazzolo, J. Behley, P. Lottes, P. Giguere and C. Stachniss, "ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2019) pp. 7855–7862.

Cite this article: S. Xiang, C. Dong, K. Zhang, G. Tai, T. Yuan, H. Yan and X. Chen (2025). "A robust visual simultaneous localization and mapping system for dynamic environments without predefined dynamic labels and weighted features", Robotica **43**, 3366–3394. https://doi.org/10.1017/S0263574725102506

https://doi.org/10.1017/S0263574725102506 Published online by Cambridge University Press