Hindawi
Mobile Information Systems
Volume 2023, Article ID 1145346, 12 pages
https://doi.org/10.1155/2023/1145346

Research Article

DIR-SLAM: Dynamic Interference Removal for Real-Time VSLAM in Dynamic Environments

Xiaomin Ma,[1,2] Ye Yang,[1] Lei Zhu,[1] Yingmin Yi,[2] Jing Xin,[2] Xueping Su,[1] and Minqi Li[1]

[1]*School of Electronics and Information, Xi'an Polytechnic University, Xi'an 710000, Shaanxi, China*
[2]*The Faculty of Automation and Information Engineering, Xi'an University of Technology, Xi'an 710048, Shaanxi, China*

Correspondence should be addressed to Ye Yang; wo712268@163.com

Received 21 April 2022; Revised 14 December 2022; Accepted 10 January 2023; Published 4 February 2023

Academic Editor: Floriano Scioscia

Traditional visual simultaneous localization and mapping (VSLAM) systems mostly rely on the static-world assumption, which limits their applications in real-world scenarios with dynamic objects. When there are dynamic objects in the scene, the localization accuracy of the system decreases seriously. In this paper, in order to minimize the interference of dynamic objects in vision localization, we propose a real-time and robust dynamic interference removal (DIR) method, which is based on both prior knowledge and geometry information. Our approach employs a novel lightweight CNN network to output semantic labels and extends the semantics based on the correlations of descriptors to generate a segmented mask. We design a geometric consistency check module to remove the dynamic interference, which computes the bundle adjustment to predetermine the static keypoints, and then the semantic weighted epipolar constraint is used to identify the dynamic outliers. The proposed method is integrated into the front end of ORB-SLAM2 to filter out the dynamic keypoints which are associated with the known and unknown dynamic objects. We conduct experiments on the public TUM RGB-D dataset, the qualitative and quantitative results prove that the DIR method can improve the performance of the state-of-the-art VSLAM system in dynamic scenarios.

1. Introduction

A mobile robot is an intelligent system that integrates multiple functions such as environment perception, dynamic anticollision, and motion control. To ensure the various capabilities of a mobile robot, visual simultaneous localization and mapping (VSLAM) are considered fundamental issues. In recent decades, VSLAM systems [1–7] based on visual sensors have attracted increasing attention and been well studied, with a rather satisfactory performance that can facilitate high-level tasks [8–10]. Typically, some well-performing VSLAM systems have been developed, such as ORB-SLAM2 [3] and LSD-SLAM [2]. Given a sequence of images, these systems can jointly estimate the camera pose and generate a continuous camera trajectory. However, the vast majority of the VSLAM systems are based on the assumption of static environments, which estimate the pose through a static feature set. As a consequence, they are vulnerable to unexpected changes in surroundings, such as dynamics, especially humans. In these scenarios, dynamic content affects the whole process of VSLAM, which inevitably degrades localization accuracy and reliability.

To address these problems, many algorithms have been adopted to make the existing VSLAM systems dynamic-object-aware [8, 11–19]. Algorithms such as random sample consensus (RANSAC) [20, 21] are employed to reject outliers, which can weaken the dynamic interference by optimizing the feature set. However, these algorithms tend to fail when moving objects occupy a major part of the camera field of view. Compared with the method of purely optimizing features, distinguishing the content of a scene as static or dynamic benefits visual localization in dynamic scenarios [8, 11–14].

With the development of machine learning, VSLAM systems combined with deep learning methods are well developed [15–19]. Advanced convolutional neural network (CNN) architectures such as Mask R-CNN[22], YOLO [23], and SegNet [24] are applied to effectively obtain prior

knowledge, which is used to classify the objects of scenes. However, these methods handle the known objects and ignore the unknown dynamic objects, which are labeled as background. Therefore, it is not enough to judge the objects only by prior knowledge. In addition, most of these methods suffer from high computational costs and easily cause information loss. Therefore, robustness and low computation cost are two challenges for such approaches.

In this paper, we propose a real-time and robust dynamic interference removal (DIR) method for dynamic scenarios, which mainly includes a semantic part and a geometric consistency check module. The former is composed of a novel semantic segmentation network and dynamic correlation region, which were introduced to provide a pixel-wise classification and extend the semantics of the local areas which are correlated with dynamic objects. The latter uses bundle adjustment and semantic weighted epipolar constraint to identify and reject the dynamic outliers. The main contributions of the proposed method are summarized as follows:

(1) We propose a novel lightweight semantic segmentation network built on MobileNetV2 [25], called De-MNetV2, which is more sensitive to dynamic objects and inconspicuous details. To obtain the dynamic content completely, we define the local areas which are correlated with dynamic pixels as the dynamic correlation region and extend the corresponding semantics of this region.

(2) We design an efficient geometric consistency check module, which is based on the bundle adjustment (BA) and the epipolar geometry constraint with semantic weights. The former is computed to predetermine static keypoints for avoiding information loss, and the latter is used to robustly identify the dynamic keypoints on the known and unknown objects.

(3) We insert the proposed method into ORB-SLAM2 [3], which is called DIR-SLAM (Dynamic Interference Removal SLAM system). Experiments on the widely used TUM RGB-D benchmark dataset [26] prove convincingly that visual localization accuracy in dynamic environments can be greatly boosted.

The rest of this article is organized as follows: Section 2 summarizes various dynamic SLAM methods and presents the essence of VSLAM problems in dynamic environments. Section 3 describes the theoretical content and verification of the proposed method. Section 4 shows the experimental results and analysis. We draw some conclusions and deliver future work in Section 5.

2. Related Work

In dynamic environments, some areas of the image may be taken up by dynamic pixels. As a result, the visual localization accuracy cannot be guaranteed resulting from the fusion of the dynamic content. To address this problem, we give a comprehensive analysis of the existing dynamic VSLAM algorithms in Section 2 and explain the nature of dynamic VSLAM problems in Section 2.2.

2.1. Existing Dynamic VSLAM Algorithms. The direct methods mainly depend on the temporal or spatial coherence of dynamic points, such as the comparison of geometric structures [8, 11–14]. Jaimez et al. [11] use the K-means clustering algorithm and reprojection errors to classify geometric clusters as static or dynamic, then the dense dynamic points are removed. Scona et al. [8] employ both sensor information fusion and points of static probability to optimize the robot's pose. Sun et al. [14] designed a motion removal method to address the problem of RGB-D SLAM in dynamic environments, which can estimate the possible foreground points by dense optical flow computing. According to the 3D information provided by the RGB-D camera, the depth information can be regarded as the only classification criterion. Li and Lee [12] present a static weighting method for handling depth edge points to indicate the likelihood of one point being part of the static environment, which can improve the tracking and mapping performance.

Despite being suitable for dynamic environments, these methods use all pixels in the image for pose estimation; therefore, projection errors caused by interference such as camera noise and illumination changes cannot be properly handled, and thus reliable localization results are often not consistently achieved. In addition, geometric structures can only determine the moving objects and not the moveable objects, such as people who keep things static in their environment. Therefore, it is necessary to introduce prior knowledge to infer the moveable objects.

With the prosperity of deep learning technologies, the feature-based VSLAM combined with deep-learning methods which can provide prior knowledge have developed rapidly to infer the dynamic objects with impressive performance [15–18]. Yu et al. [15] employ SegNet to obtain the semantics and check the moving consistency, then optimize the localization by filtering keypoints on humans. Bescos et al. [16] combine multiview geometry models and Mask R-CNN for detecting dynamic objects and use the region growth algorithm to remove all the dynamic points in the mapping process to estimate static maps. Cheng et al. [17] jointly employ YOLO3, Faster R-CNN, and SSD detection models as the prior knowledge generation module, and then a Bayesian framework is applied to determine and discard dynamic regions.

2.2. VSLAM Problems in Dynamic Environments. In feature-based VSLAM systems, the interference caused by dynamic objects is multifaceted and mainly reflected in keypoints, descriptors, and geometric structures. Dynamic keypoints on the moving objects, lead to inaccurate landmarks of tracking. Meanwhile, because the patch-based descriptors are constructed by sampling neighboring points of an area [27], thus the descriptors will contain dynamic content when dynamic keypoints exist in the area, which is not conducive to feature matching and pose estimation. Finally, the

dynamic keypoints destroy the consistency and cause conflicts in geometric structures, which directly reduce the accuracy of visual localization.

The essence of VSLAM problems in dynamic environments is observations; hence, we filter out unreliable observations to remove dynamic interference. As we discussed in Section 2.1, since the objects with high-dynamic probability easily cause pose estimation errors and trajectory tracking failures, we contend it is essential to use the deep learning networks to prior infer the moveable objects. However, the prediction results of the network are often inaccurate, hence the geometric structures cannot be ignored, which express the consistency of points. Figure 1 shows that dynamic keypoints can destroy the epipolar geometry constraints.

Compared with the methods mentioned in Section 2.1, our proposed method falls into the feature-based VSLAM combined with deep-learning methods and we describe the detailed characteristics in Section 3.

3. DIR-SLAM

3.1. Method Overview. ORB-SLAM2 [3] is the most used solution for visual localization and has shown excellent performance in most practical situations. However, in dynamic environments, it suffers a lot. Therefore, we propose a dynamic interference removal method (DIR), which is named DIR-SLAM. The flowchart of DIR-SLAM is shown in Figure 2.

Figure 3 illustrates the details of the DIR method. It contains the following four sections: (1) semantic part; (2) dynamic correlation region; (2) feature matching; (3) geometric consistency check module. In the semantic part, we design a lightweight semantic segmentation network to output the semantic labels, which are arranged according to the likelihood of movement from 0 (background) to 20 (person). Then, the semantic content is extended according to the dynamic correlation region and generates a segmented mask. The dynamic correlation region is defined in Section 3.3. We use the segmented mask to provide semantic weights. For feature matching, the keypoints are tracked between the current frame and the previous frame by optical flow [28] to generate initial feature matches. In the geometric consistency check module, the BA is computed first to reserve static keypoints which are consistent with the previous camera pose and then the epipolar geometry constraint with semantic weights is calculated to identify and reject the dynamic outliers.

3.2. Semantic Segmentation. Consecutive frames captured by a moving camera are inevitably blurred or appeared ghosting, which requires higher scene parsing ability. In addition, frequently changing details of dynamic objects lead to strong interference and inconspicuous pixels in frames. Therefore, it is worthwhile to capture details. To address these problems, we propose a developed MobileNetV2 [25] network, i.e., De-MNetV2. The network structure of De-MNetV2 is shown in Figure 4.

Figure 4 shows the pyramid pooling module (PPM) of PSPNet [29] is connected with the backbone, namely, the PSP header, which gathers global context information and provides a complete understanding of the scene. Considering that the low-level layers of the network are rich in spatial details [31], we insert two skip connection branches to fuse the low-level features for increasing details, which benefits high-level features. The branches first extract the low-level features through dilation convolution, and then we use the fully connected layer for the dimensions consistently. Finally, the details from branches and the global context information provided by the PSP header are superposed and sent to the decoder to predict the semantic labels.

The PSP Header can both develop adaptability and scene parsing in dynamic environments to reduce mismatches and confusion categories. Meanwhile, the details can benefit the network by enhancing the classification performance. We hold the opinion that these improvements are useful for classifications in dynamic environments.

3.3. Dynamic Correlation Region. In feature-based VSLAM methods, the camera pose is estimated by matching the descriptors of keypoints, such as ORB [32], SIFT [33], and FREAK [34]. Here, we take the ORB algorithm as an example to illustrate the correlation between dynamic points and neighbors. Next, we define the areas affected by dynamic objects as the dynamic correlation region.

The ORB algorithm uses Rotation-Aware BRIEF (rBRIEF) descriptor. To generate the rBRIEF descriptor, a pixel-wise circular sampling patch centered at the keypoint is first rotated according to the orientation of the keypoint to guarantee rotation invariance. The circular patch centered at the keypoint is shown in Figure 5.

In Figure 5, assuming$O$ is static, the orientation angle θ of O is calculated by [35]

$$\theta = \operatorname{atan} 2\left(m_{01}, m_{10}\right), \tag{1}$$

where m_{01}, m_{10} are defined as follows [35]:

$$m_{pq} = \sum_{x,y} x^p y^q I\left(x, y\right). \tag{2}$$

It can be seen that θ is closely relevant to the intensity $I\left(x, y\right)$ of the patch. Hence, when O is moving, centroid C becomes to $C^{'}$ and θ drifts to $\theta^{'}$. The orientation deviation angle β can be expressed as follows:

$$\beta = \operatorname{abs}\left(\theta^{'} - \theta\right). \tag{3}$$

In these cases, the rBRIEF descriptor will contain dynamic content, which can affect feature matching. To validate the influence, we simulate the orientation calculation process of rBRIEF descriptor. Pairs of frames are captured from different conditions to represent the various dynamic environments. We show the comparative results in Figure 6. First, we gray the image and filter it with the Gaussian kernel (7×7). The orientation is computed by (2), and the angular deviation β between image pairs of each pixel is calculated by equation (3). For more intuitively, β value between $[0^{\circ}, 360^{\circ}]$

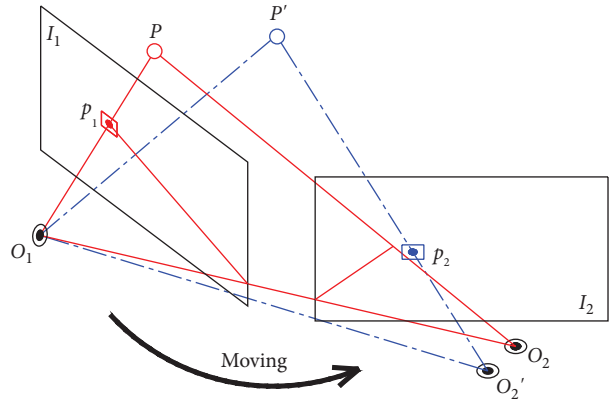


Figure 1: The dynamic interference to geometric structures. Assume that P is a moving point, which moves to P' in image I_2. p_2 is the dynamic keypoint which is matched with p_1. O_2 is the true camera pose. O_2' is estimated by the matched keypoints.

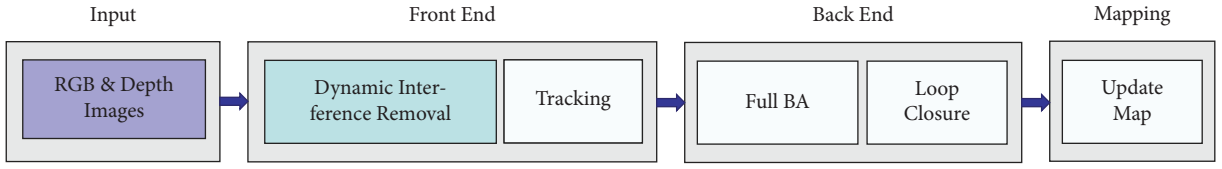


Figure 2: The framework of DIR-SLAM. We insert the DIR method into the front end of ORB-SLAM2 and serve as a preprocessing stage to remove dynamic interference. Tracking, back end, and mapping threads are the same as ORB-SLAM2.

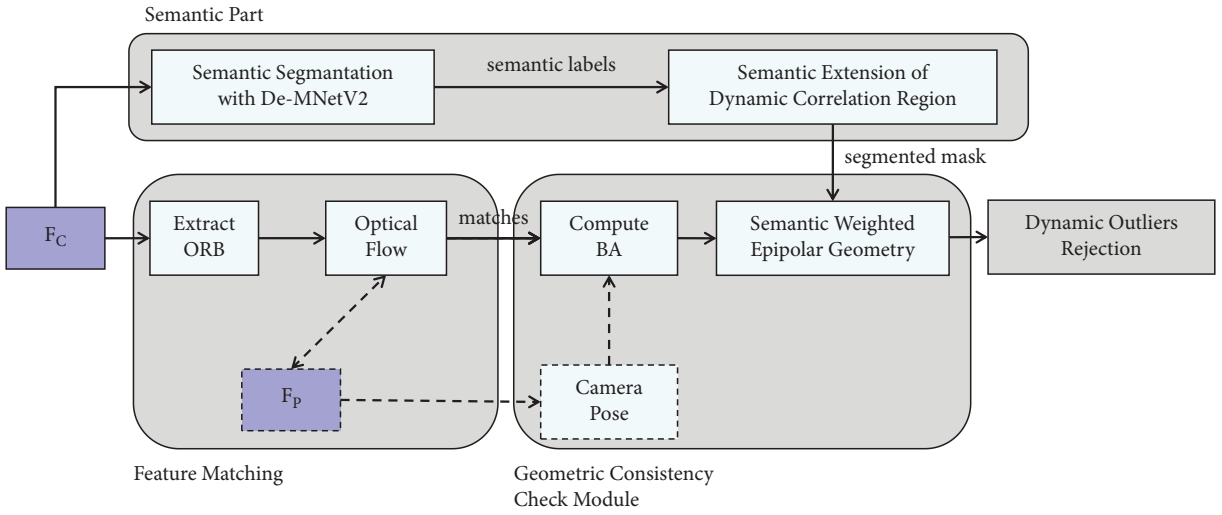


Figure 3: The flow of the DIR method in 2 consecutive RGB color frames. F_c indicates the current frame, F_p indicates the previous frame. The method requires (1) feature matches between F_c and F_p; (2) the segmented mask obtained by prior semantic information from both De-MNetV2 and dynamic correlation region; (3) the pose of the previous frame F_p.

is scaled to the red color channel of $[0, 255]$ and represented by a mask. The redder the color, the more severe the angular deviation, which shows the stronger the impact of dynamic objects.

Simulation results indicate that dynamic objects can inevitably affect static points. Especially in the circular neighboring areas with the radius r. We notice that dynamic semantics should occupythese areas, which will better play the role of prior knowledge. Hence, we define these areas as the dynamic correlation region.

According to (1) and (2), we employ the morphological dilation algorithm to extend the semantics of dynamic objects for covering the dynamic correlation region. The dilation kernel k can be explained by

$$k = \text{patch\_size} = 2 \cdot r - 1. \tag{4}$$

Then, the segmented mask of the frame is updated. Without loss of generality, if another feature extractor is adopted, parameter k just needs to be adjusted based on the patch size.

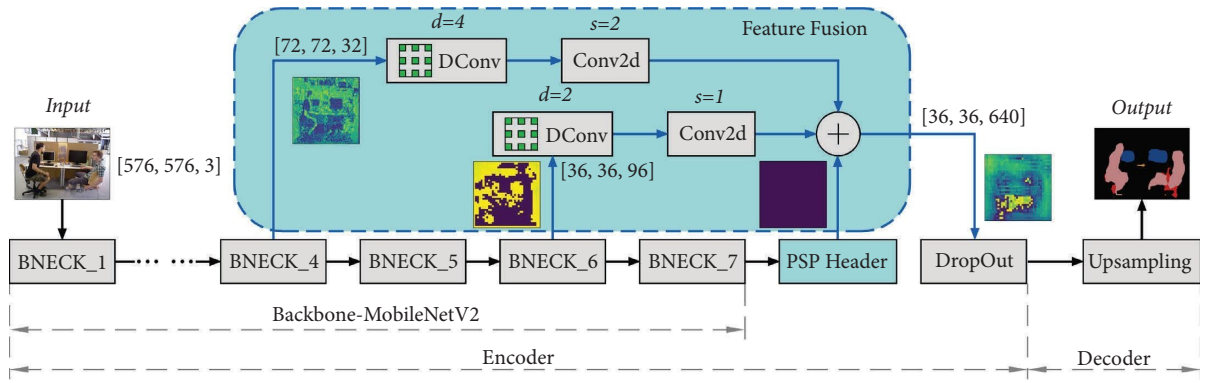


Figure 4: The overview of De-MNetV2. The network is built on MobileNetV2. The PSP header is the same as the pyramid pooling module (PPM) of PSPNet [29]. *D*Conv is the dilated convolution, and *d* is the dilation rate. Conv2*d* is the fully connected layer with 1×1 convolution kernel, and s represents the step size. After *D*Conv layer, the batch normalization [30] layer is connected, extra batch normalization and ReLU activation are added after each Conv2*d* layer.

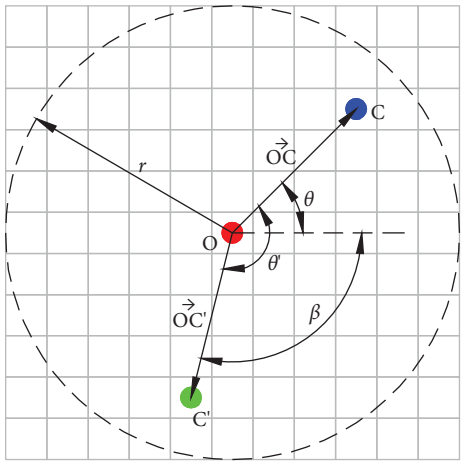


Figure 5: Orientation calculation within a circular area of radius r. O is the center of the patch and the centroid is C. When a dynamic object appears, the centroid C will drift to C' resulting from the changes in intensity.

3.4. Geometric Consistency Check Module. We design an adaptive geometric consistency check module between two consecutive frames, which can be applied in scenarios with known and unknown dynamic objects to robustly remove the dynamic interference. First, we compute BA [36] to estimate the static keypoints, which are consistent with the previous camera pose. Then, the epipolar geometry constraint with semantic weights is calculated to identify and reject dynamic outliers.

As shown in Figure 7, we assume that the keypoint p_c in the current frame F_c is matched with p_p in the previous frame F_p, which means the camera observes P_w. The coordinates of the matched pair can be expressed as $p_c = [u_c v_c 1]$ and $p_p = [u_p v_p 1]$. Since the pose of the previous frame is known, according to the reprojection model, we can respect the observed point to the world coordinate frame and compute the corresponding 3D coordinates $P_w = [X_w Y_w Z_w]$. Thus, the reprojection error between p_c and p_p is calculated as follows:

$$\left\| p_c - \pi_c\left(P_w, \mathbf{T}_p\right) \right\| \leq \varepsilon, \tag{5}$$

where π_c is the projection function of the current frame and $\mathbf{T}_p$ is the pose of the previous frame. We set a small value to ε (1.0), thus the keypoints that satisfy equation (5) are considered static, which are directly reserved as inliers. For other matched pairs, the epipolar line l_c of the keypoint p_c is calculated as follows:

$$\begin{aligned} l_c &= \mathbf{F} \cdot p_p^T \\ &= \mathbf{F} \cdot \left[u_p v_p 1\right]^T \\ &= [ABC]^T, \end{aligned} \tag{6}$$

where $\mathbf{F} \in \mathbb{R}^{3\times3}$ denotes the fundamental matrix and A, B, and C denote real vectors. Then, the distance from the p_c to the epipolar line l_c is denoted by the following equation:

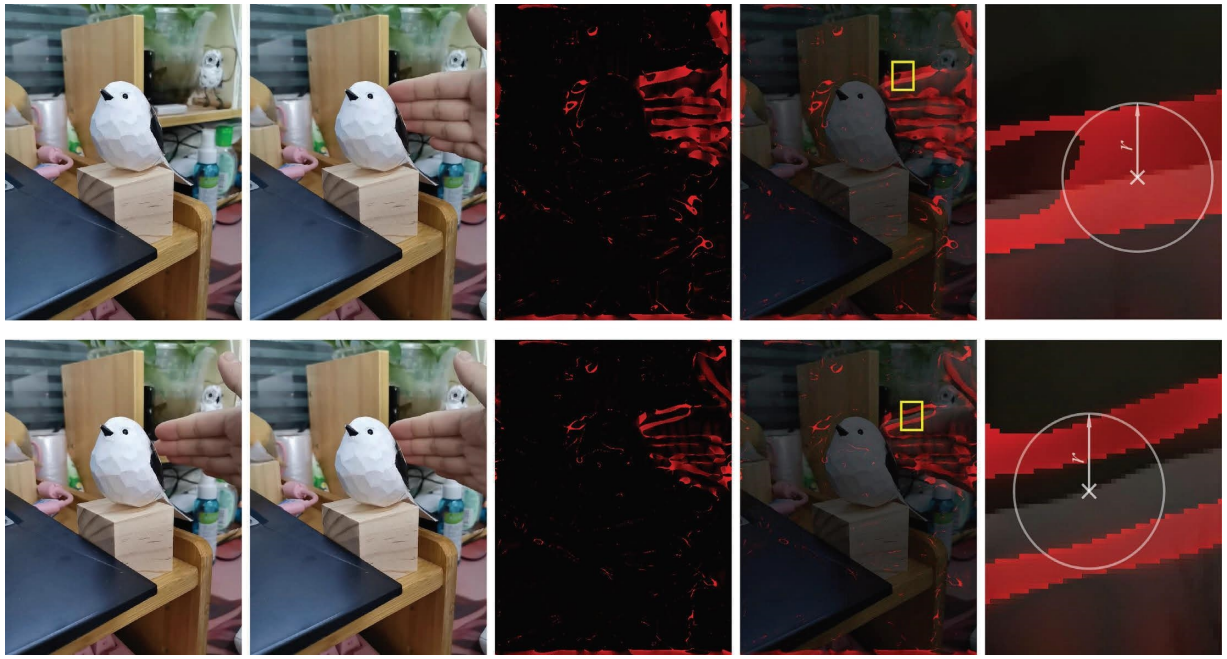


Figure 6: The deviation angles in different conditions. From left to right: the original image, the comparison image with dynamic objects, β mask, β mask superimposed with the original image, and the partially enlarged diagram of the yellow rectangle. Finally, the white circle is the patch that the radius is r.

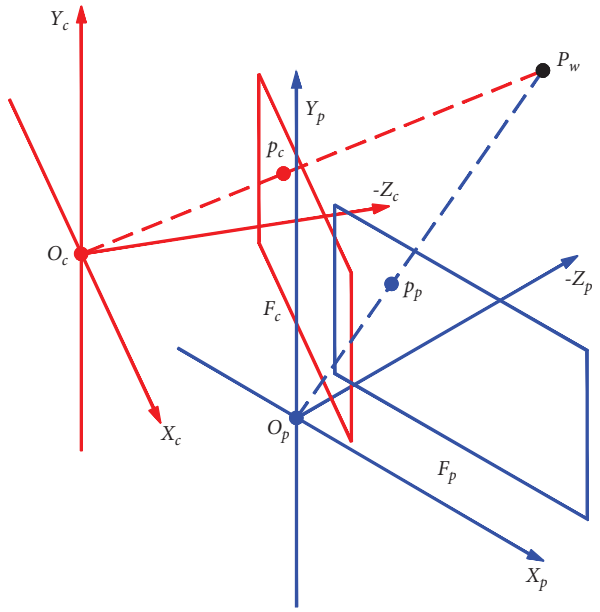


Figure 7: The camera coordinate frames of the current frame and previous frame.

$$d = \frac{\left\| p_c \mathbf{F} p_p^T \right\|}{\sqrt{\|A\|^2 + \|B\|^2}}. \tag{7}$$

Because of the sensor errors and dynamic interference, keypoints deviate from the epipolar lines. The larger the distance d, the more likely the keypoint is to be dynamic. Here, we employ the prior semantic information to weigh the distance d, where the label values are arranged according to the likelihood of movement from low (0) to high (20). Therefore, the label of higher motion probability covers lower ones after the semantic extension in Section 3.3. We assign the label values and corresponding semantic weights as shown in Figure 8.

As shown in Figure 8, the semantic weight increases with the likelihood of the object moving. We employ the segmented mask to provide the semantic weight $W(p_c)$ ofp_c. The final distance function is calculated as follows:

$$d_c = W(p_c) \cdot d. \tag{8}$$

d_c is used to identify and reject dynamic outliers. When d_c is larger than a certain threshold (1.0), the keypoint is considered a dynamic point and is rejected.

We extract the results of the single-step of geometric consistency check module as shown in Figure 9. In Figure 9(a) the green points are predetermined inliers by BA. These keypoints are recognized as static and have priority reserved for avoiding information loss. In Figure 9(b), the distance of semantic weighted epipoplar geometry is computed, and the filtered dynamic keypoints are shown as red points. Blue points represent the remaining static keypoints, which are employed to track the pose.

4. Experiments

To prove the effectiveness of the proposed method in this paper, we conduct comparative experiments on the methods and evaluate the results quantitatively and qualitatively. In this section, we experimentally evaluate the effectiveness of the proposed method from the following two parts: (1) DeMNetV2 network; (2) DIR-SLAM.

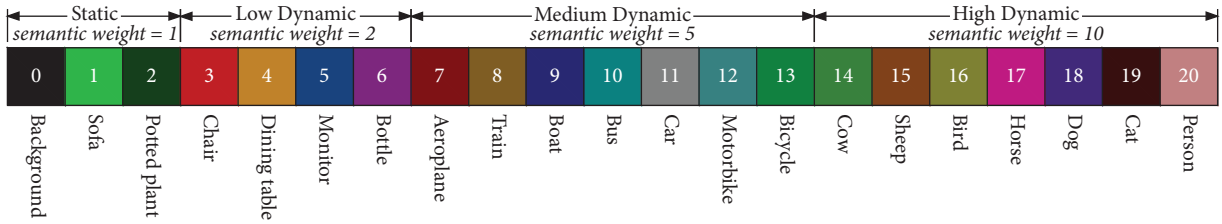
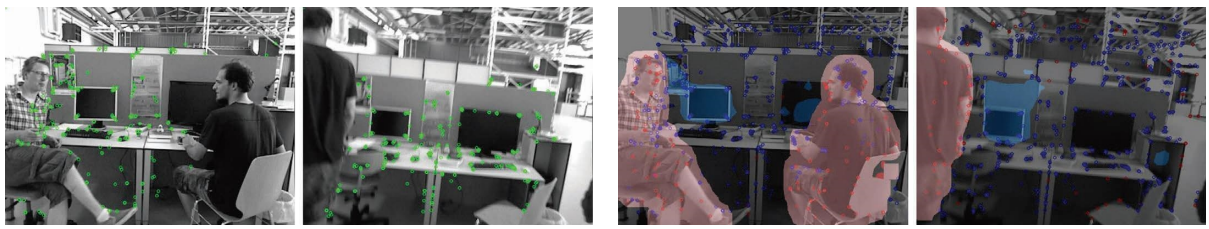


Figure 8: Illustration of semantic weights for each category of objects. We assign the semantic weights for objects according to the likelihood of movements. Here, the weights for static, low dynamic, medium dynamic, and high dynamic are, respectively, assigned to 1, 2, 5, and 10.



(a) (b)

Figure 9: The single-step results of the geometric consistency check module: (a) the predetermined static keypoints by BA; (b) the results of dynamic outliers rejection superimposed with the segmented masks.

4.1. De-MNetV2 Network. The De-MNetV2 network is trained on PASCAL VOC 2012 dataset [37]. The model can detect 20 classes that contain common dynamic objects, e.g., people, cats, and dogs, which suffices to meet the testing requirements of the TUM dataset [26]. If the environment is complex, the model should be trained on the COCO dataset [38] to classify more categories. The implementation is training on the public platform Keras with a GTX 2080Ti, Intel E5-2678 v3 CPU, and 64GB RAM. For data augmentation, we randomly scale (from 0.5 to 1.5) and left-right flip the input images. The images are cropped to 576×576 and grouped with batch size 6. We set the initial learning rate to 0.0001, which gradually decreases to 0 by following the "poly" strategy $(1 - \text{iter}/\max_{\text{iter}})^{\text{power}}$ and (power = 0.9) [39]. The network is trained with Adam, of which the weight decay is set to 0.00001. The pixel-wise dice loss [40] is used as the loss function.

We conduct the experiments with the evaluation metric mean Intersection Over Union (mIOU) and mean Pixel Accuracy (mPA). Table 1 gives the comparison of semantic segmentation on the PASCAL VOC 2012 validation set.

Our network achieves mIOU of 75.75% and mPA of 84.06%. Compared with the original MobileNetV2, the mIOU of MobileNet + PSPNet is reduced by 2.31%. Because the multiscale pyramid pooling module abstracts the high-level features, which enhance scene parsing but reduce classification capabilities in lightweight networks. After the insertion of skip connection branches, the mIOU of De-MNetV2 is 2.14% higher than that of MobileNet + PSPNet, which indicates that the fusion of details from low-level layers can refine the high-level features and improve overall performance. Compared to MobileNet + DeepLabV3, which performs the best semantic segmentation performance in the original paper [25], our network is competitive.

Table 1: Comparison on PASCAL VOC 2012 validation set.

Network	Header	Params (M)	mIOU (%)	mPA (%)
MNet V2*	ASPP	2.15	74.92	83.49
	PSP	2.41	73.61	82.60
De-MNetV2	PSP	2.59	**75.75**	**84.06**

[1]MNet V2*: second last feature map is used for DeepLabv3 heads [39] or PSPNet heads, named, separately MobileNet + DeepLabV3 [25] and MobileNet + PSPNet. [2]The output stride is fixed on 16. All the results come from our own experiments. [3]All the networks are trained on the PASCAL VOC2012 train set without pretraining weights.

Figure 10 lists the comparison of scene parsing ability and details between De-MNetV2 and MobileNet ∗ DeepLabV3. These results suggest that De-MNetV2 is more sensitive to dynamic objects, and the misclassifications and discontinuous labels are fewer. We suggest that the De-MNetV2 is more suitable for our requirements.

From Tables 2 and 3, we can see that DIR-SLAM gets competitive ATE and RPE values in most of the sequences, which shows that our method achieves excellent performance. The results illustrate that the proposed DIR method has an exponential growth rate compared to the original ORB-SLAM2 in high-dynamic scenarios, which is effective and excellent.

Similar to the DS-SLAM, we segment the scene and use the epipolar constraints to determine and reject the dynamic outliers. The reasons our method outperforms DS-SLAM are because we extend the semantics in the dynamic correlation region to obtain the prior information more completely, and we use the semantic weights to make the movement of dynamic targets more obvious. Besides, we do not cluster the keypoints by pixels or geometric structures. All the keypoints are identified robustly and independently, the experimental results present the feasibility. Compared with the DynaSLAM, the experimental results show that our performance is very close.

Figure 10: Comparison of segmentation results. (a): the images are segmented by MobileNet + DeepLabV3 [25], which performs best in the original paper. (b): the images are segmented by our network.

In the semantic part, DynaSLAM extends semantics similarly, but without going into further theoretical analysis. DynaSLAM relies on Mask R-CNN and multi-view geometry to improve semantics, which brings in expensive computational costs. Our method is implemented by a lightweight De-MNetV2 network and semantic extension of the dynamic correlation region, which are fast and efficient.

We evaluate the accuracy of our system with different configurations, and the RMSE of ATE is shown in Table 4. We test four different configurations of DIR-SLAM.

(1) MNet V2*: MobileNet ∗ DeepLabV3 is used for semantic segmentation

(2) Correlation*: the semantics do not extend to cover the dynamic correlation region

(3) Semantic*: the semantic weights are not used

(4) BA*: BA of the geometric consistency check module is not computed

Table 4 shows that the results of DIR-SLAM are better than others. In view of Table 2, we notice that the experimental results of our method are not ideal in low-dynamic sequences. Due to robustness, ORB-SLAM2 is enough to overcome the dynamic interference caused by slight movements. For f3/s/xyz sequence, the ATE of ORB-SLAM2 is 0.0097 meter and better than our method. However, Table 4 shows that the DIR-SLAM configured with Semantic* outperforms others in this sequence, in which the semantic weights are removed. Hence, we consider that prior knowledge can lead to information loss. We use the BA to verify the consistency between keypoints and the pose of the previous frame, which predetermines the static keypoints and reduces information loss. In terms of experimental results of DIR-SLAM, the gaps between DIR-SLAM and ORB-SLAM2 in f3/s sequences are small.

Figure 11 shows the comparative results of trajectories in sequences. We use three types of lines to express the trajectory. It can be seen that the trajectory of DIR-SLAM is closer to the ground truth, which indicates that the localization results of DIR-SLAM are more precise. Although the experimental results have a certain degree of uncertainty, they still basically follow regularity. Based on the results of qualitative analysis, the DIR-SLAM is more suitable for high-dynamic scenarios with larger camera movements.

4.2. DIR-SLAM. We have evaluated DIR-SLAM in the public TUM RGB-D dataset [26] and compared it with other state-of-the-art VSLAM systems [15, 16]. The runtime analysis is presented to show the efficiency of our method. Furthermore, we demonstrate the performance of our method with a Kinect V1 in a real environment.

The descriptions of sequences for evaluation are as follows: The f3 sequences are dynamic object sequences, which contain four types of camera motions. (1) half (half sphere): the camera has been moved on a small half sphere of approximately one-meter diameter. (2) rpy: the camera has been rotated along the principal axes (roll-pitch-yaw) at the same position. (3) static: the camera has been kept in place manually. (4) xyz: the camera has manually been moved along three directions (xyz) while keeping the same orientation. Specifically, f3/s means f3_sitting sequences, which depict low-dynamic scenarios, and f3/w means f3_walking sequences, which depict high-dynamic scenarios. The suffix *v* represents validation sequences with undisclosed ground truth. Each sequence contains both RGB and depth images recorded at the full frame rate (30Hz) of 640 × 480 size. We performed all the experiments in a notebook with 2.6 GHz Intel i7-9750H, 16 GB RAM, NVIDIA GTX1660Ti, and Ubuntu16.04.

The quantitative evaluation indicators of the comparison, respectively, employ Absolute Trajectory Error (ATE) and Relative Pose Error (RPE). Our method is compared with other methods in terms of Root-Mean-Square Error (RMSE) and Standard Deviation (SD).

We chose the most advanced dynamic VSLAM methods DS-SLAM [15], DynaSLAM [16], and the original ORB-SLAM2 [3], for performance comparisons. All the methods are based on ORB-SLAM2, and the comparison results are shown in Tables 2 and 3.

Table 2: Evaluation of ATE for the results of ORB-SLAM2, DS-SLAM, DynaSLAM, and DIR-SLAM (m).

Sequences	ORB-SLAM2		DS-SLAM [15]		DynaSLAM [16]		DIR-SLAM (ours)		Improvement	
	RMSE	SD	RMSE	SD	RMSE	SD	RMSE	SD	RMSE (%)	SD (%)
f3/w/half	0.4525	0.2675	0.0303	0.0159	**0.025**	—	0.0275	0.0133	93.92	95.03
f3/w/rpy	0.7438	0.3644	0.4442	0.2350	0.035	—	**0.0298**	**0.0184**	95.99	94.95
f3/w/static	0.4074	0.1246	0.0081	0.0036	**0.006**	—	0.0062	0.0027	98.48	97.83
f3/w/xyz	0.7978	0.4553	0.0247	0.0161	0.015	—	**0.0147**	**0.0068**	98.16	98.51
f3/w/half/v	0.5899	0.3418	—	—	—	—	**0.0218**	**0.0120**	96.30	96.49
f3/w/rpy/v	0.5935	0.3239	—	—	—	—	**0.0291**	**0.0184**	95.10	94.32
f3/w/static/v	0.7189	0.1952	—	—	—	—	**0.0087**	**0.0045**	98.79	97.69
f3/w/xyz/v	1.6046	0.6560	—	—	—	—	**0.0116**	**0.0051**	99.28	99.22
f3/s/half	0.0244	0.0136	—	—	0.017	—	**0.0127**	**0.0060**	47.95	55.88
f3/s/rpy	0.0197	0.0126	—	—	**0.015**	—	0.0193	0.0116	2.03	7.94
f3/s/static	0.0080	0.0037	0.0065	0.0033	—	—	**0.0059**	**0.0028**	26.25	24.32
f3/s/xyz	**0.0097**	**0.0049**	—	—	—	—	0.0099	0.0049	−2.06	0.00

Table 3: Evaluation of RPE for the results of ORB-SLAM2, DS-SLAM, and DIR-SLAM (m).

Sequences	ORB-SLAM2		DS-SLAM [15]		DIR-SLAM (ours)		Improvement	
	RMSE	SD	RMSE	SD	RMSE	SD	RMSE (%)	SD (%)
f3/w/half	0.3593	0.2873	0.0297	**0.0152**	**0.0296**	0.0154	91.76	94.64
f3/w/rpy	0.4436	0.3331	0.1503	0.1168	**0.0414**	**0.0249**	90.67	92.52
f3/w/static	0.2143	0.1937	0.0102	0.0048	**0.0085**	**0.0040**	96.03	97.93
f3/w/xyz	0.4284	0.2877	0.0333	0.0229	**0.0195**	**0.0090**	95.45	96.87
f3/w/half/v	0.2853	0.2291	—	—	**0.0263**	**0.0159**	90.78	93.06
f3/w/rpy/v	0.3482	0.2855	—	—	**0.0354**	**0.0223**	89.83	92.19
f3/w/static/v	0.2477	0.2188	—	—	**0.0102**	**0.0055**	95.88	97.49
f3/w/xyz/v	0.4295	0.3816	—	—	**0.0153**	**0.0072**	96.44	98.11
f3/s/half	0.0250	0.0182	—	—	**0.0143**	**0.0066**	42.80	63.74
f3/s/rpy	0.0250	0.0148	—	—	**0.0248**	**0.0141**	0.80	4.73
f3/s/static	0.0094	0.0042	0.0078	0.0038	**0.0073**	**0.0034**	22.34	19.05
f3/s/xyz	**0.0121**	**0.0062**	—	—	0.0127	0.0067	−4.96	−8.06

Table 4: Evaluation of ATE on the TUM RGB-D dataset using the proposed method with different configurations (m).

Sequences	MNet V2*	Correlation*	Semantic*	BA*	DIR-SLAM (ours)
f3/w/half	0.0283	0.0304	0.0392	0.0306	**0.0275**
f3/w/rpy	**0.0297**	0.1186	0.1816	0.0574	0.0298
f3/w/static	0.0069	0.0084	0.0093	0.0085	**0.0062**
f3/w/xyz	0.0152	0.0159	0.0652	0.2729	**0.0147**
f3/w/half/v	0.0231	0.0316	0.0324	0.0291	**0.0218**
f3/w/rpy/v	0.0303	**0.0280**	0.0294	0.0306	0.0291
f3/w/static/v	0.0088	0.0084	0.0093	0.0088	**0.0083**
f3/w/xyz/v	0.0121	0.0115	0.3411	0.0125	**0.0113**
f3/s/half	0.0137	0.0144	0.0152	0.0146	**0.0127**
f3/s/rpy	0.0204	0.0213	0.0215	0.0341	**0.0193**
f3/s/static	0.0062	0.0063	0.0074	0.0064	**0.0059**
f3/s/xyz	0.0101	0.0107	**0.0089**	0.0114	0.0099

Our system is a real-time semantic SLAM system. In order to show the efficiency, we compared the average computation time of major processing modules between DIR-SLAM and ORB-SLAM2. To find a relation of the computational cost and the amount of dynamic points, we choose the f3/w/static sequence and f3/w/xyz sequence for comparison. As mentioned in Section 4.2, the motion in f3/w/xyz is more complicated because the camera is always moving. The results are shown in Table 5.

The modules in Table 5 correspond to Figure 3. Specifically, the semantic part runs in parallel with the system as a separate GPU thread. In the geometric consistency check module, f3/w/xyz takes more time than f3/w/static. Because the dynamic points in the f3/w/xyz sequence are more diffused, the fundamental matrix **F** takes more iteration time. In pose estimation, the DIR-SLAM is faster than the original ORB-SLAM2, due to dynamic outliers are rejected, the convergence rate of pose optimization is accelerating.

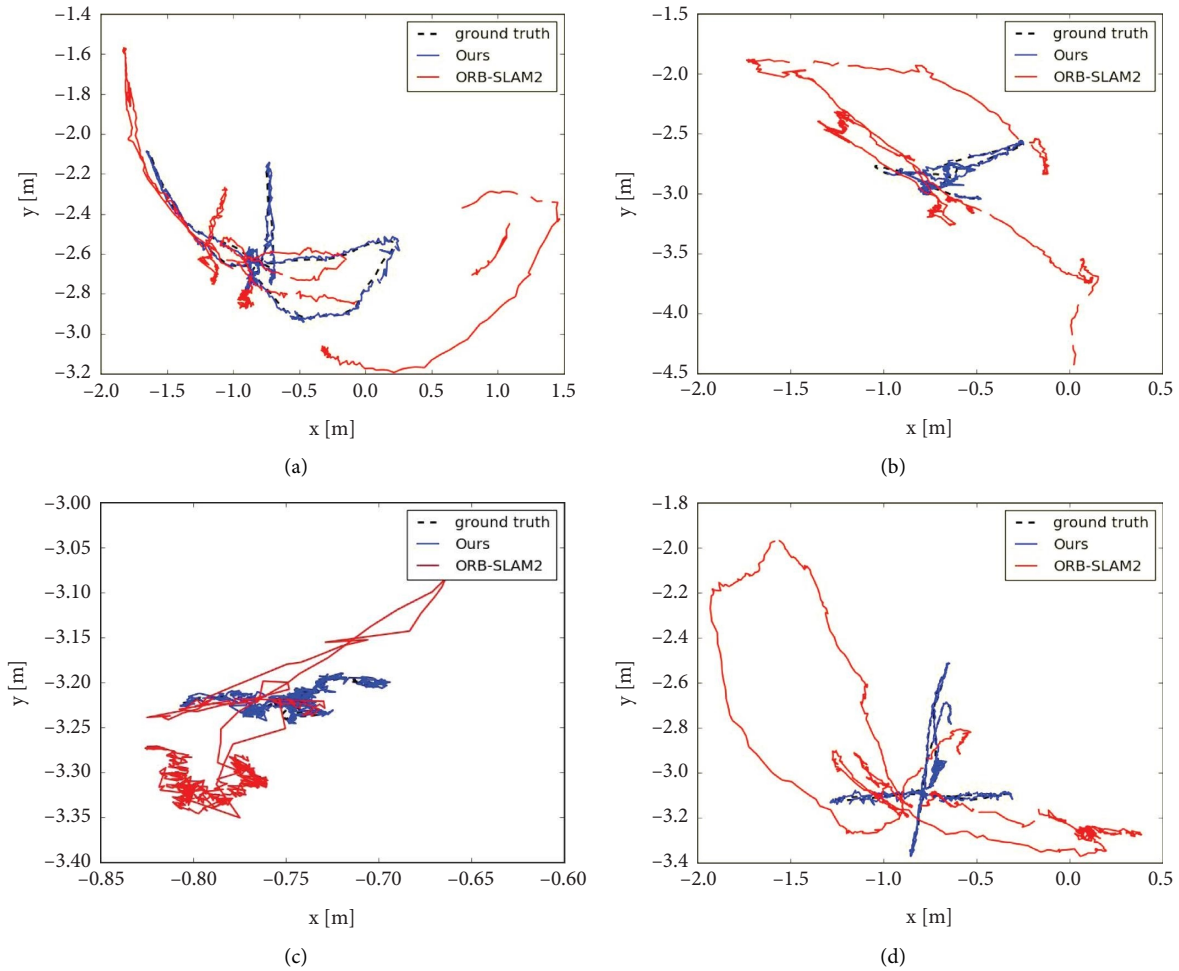
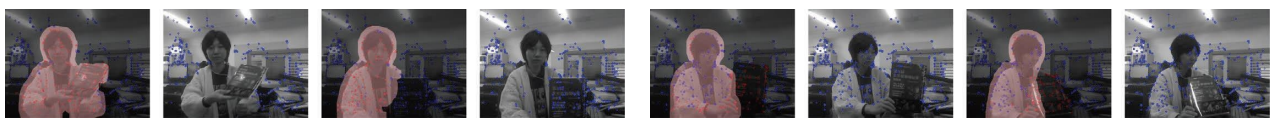


Figure 11: Trajectories of high-dynamic sequences. Each diagram from top to down displays the trajectory generated by ground truth, DIR-SLAM, and ORB-SLAM2. (a) f3/w/half. (b) f3/w/rpy. (c) f3/w/static. (d) f3/w/xyz.

Table 5: Comparison of average computation time (ms).

Modules	ORB-SLAM2		DIR-SLAM (ours)	
	f3/w/static	f3/w/xyz	f3/w/static	f3/w/xyz
Semantic part	—	—	23.1439	23.0893
Geometric consistency check	—	—	33.1647	42.8656
Pose estimation	13.6618	15.6819	6.6141	7.1349
Tracking	26.5359	27.6001	52.5604	61.2871



(a) (b)

Figure 12: Experimental results in a real environment: (a) known dynamic object: person; (b) unknown dynamic object: book.

Finally, tracking is the main thread to process every single frame, our method costs less than 100 ms and as fast as a human brain [41].

4.3. Robustness Test in a Real Environment. We integrate DIR-SLAM with ROS and conduct experiments in a real environment to demonstrate the robustness and real-time

performance. Frames are captured by a Kinect V1 camera with 640 × 480. The duration is about 2 minutes. Experimental results of DIR-SLAM during the real environment test are shown in Figure 12. The red points are dynamic keypoints, which are identified by the proposed method, and the blue points are static keypoints.

In a real environment, a person holding a book is sitting in front of the camera, and the camera is holding static. Note that the person is labeled, but the book is not. In Figure 12(a), when the person is moving meanwhile the book stays static, the dynamic keypoints are basically distributed in the person. In Figure 12(b), the book is moved but the person is not, so our method can identify the dynamic keypoints distributed in the book. We record the complete experimental test video: https://wo712268.lofter.com/post/1d4e8522_2b40b2d53.

5. Conclusion

In this article, we propose a real-time semantic DIR-SLAM to address the problems of visual localization in dynamic environments. As we depicted before, we reject all the dynamic keypoints on account of prior knowledge and geometry constraints. We use ORB-SLAM2 [3] as the system framework and perform experiments on the public TUM RGB-D dataset [26]. From the results, we notice that our system can outperform high-dynamic scenarios more than low-dynamic scenarios, and robustly take effect on unknown environments.

However, the errors still exist. The method cannot well deal with the trajectory deviation caused by the camera moving. To handle the pose estimation errors caused by the rapid and large changes of views, we intend to attempt an affine-invariant feature extractor that is more adaptable to the movements of the camera. Besides, we aim to join the semantic mapping and background repairing methods to DIR-SLAM to realize real-time dense mapping.

Data Availability

The author has used third-party data. More information about these data can be obtained from the reference Yu, C., Liu, Z., Liu, X.J., Xie, F., Yang, Y., Wei, Q., and Fei, Q., 2018. DS-SLAM: A semantic visual slam towards dynamic environments, in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROSs). https://doi.org/10.1109/IROS.2018.8593691. Bescos, B., Fácil, J.M., Civera, J., Neira, and J., 2018. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. IEEE Robotics and Automation Letters. https://ieeexplore.ieee.org/document/8421015.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by Shaanxi Key Laboratory of Complex System Control and Intelligent Information Processing (Grant no. 2020CP05), Shaanxi Natural Science Basic Research Project (Grant nos. 2022JQ-711 and 2022JM-348), Xi'an Science and Technology Bureau Science and Technology Innovation Leading Project (Grant nos. 21XJZZ0022 and 21XJZZ0020), and Key R & D Plan of Shaanxi Province (Grant no. 2020ZDLGY06-01), National Natural Science Foundation of China (Grant no. 61873200).

References

[1] T. Pire, T. Fischer, G. Castro, P. De Cristóforis, J. Civera, and J. Berlles, "S-ptam: stereo parallel tracking and mapping," *Robotics and Autonomous Systems*, vol. 93, pp. 27–42, 2017.

[2] J. Engel, T. Schops, and D. Cremers, "Lsd-slam: large-scale direct monocular slam," in *Proceedings of the 13th European conference on computer vision*, Zurich, Switzerland, September 2014.

[3] R. Mur-Artal and J. D. Tardós, "Orb-slam2: an open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[4] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: an accurate open-source library for visual, visual-inertial and multi-map slam," 2020, https://arxiv.org/abs/2007.11898.

[5] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-d mapping with an rgb-d camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177–187, 2014.

[6] A. Agudo, F. Moreno-Noguer, B. Calvo, and J. M. M. Montiel, "Sequential non-rigid structure from motion using physical priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 979–994, 2016.

[7] A. Agudo, "Total estimation from rgb video: on-line camera self-calibration, non-rigid shape and motion," in *Proceedings of the 25th International Conference on Pattern Recognition (ICPR)*, Milan, Italy, January 2021.

[8] R. Scona, M. Jaimez, Y. R. Petillot, M. Fallon, and D. Cremers, "Staticfusion: background reconstruction for dense rgb-d slam in dynamic environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3849–3856, Brisbane, QLD, Australia, May 2018.

[9] G. Tian, L. Liu, J. Ri, Y. Liu, and Y. Sun, "Objectfusion: an object detection and segmentation framework with rgb-d slam and convolutional neural networks," *Neurocomputing*, vol. 345, pp. 3–14, 2019.

[10] S. Yang and S. Scherer, "Cubeslam: monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[11] M. Jaimez, C. Kerl, J. Gonzalez-Jimenez, and D. Cremers, "Fast odometry and scene flow from rgb-d cameras based on geometric clustering," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, May 2017.

[12] S. Li and D. Lee, "Rgb-d slam in dynamic environments using static point weighting," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2263–2270, 2017.

[13] Y. Sun, M. Liu, and M. Q. H. Meng, "Improving rgb-d slam in dynamic environments: a motion removal approach," *Robotics and Autonomous Systems*, vol. 89, pp. 110–122, 2017.

[14] Y. Sun, M. Liu, and M. Q. H. Meng, "Motion removal for reliable rgb-d slam in dynamic environments," *Robotics and Autonomous Systems*, vol. 108, pp. 115–128, 2018.

[15] C. Yu, Z. Liu, X. J. Liu et al., "Ds-slam: a semantic visual slam towards dynamic environments," in *Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1168–1174, Madrid, Spain, October 2018.

[16] B. Bescos, J. M. Fácil, J. Civera, and J. Neira, "Dynaslam: tracking, mapping, and inpainting in dynamic scenes," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4076–4083, 2018.

[17] J. Cheng, H. Zhang, and M. Q. H. Meng, "Improving visual localization accuracy in dynamic environments based on dynamic region removal," *IEEE Transactions on Automation Science and Engineering*, vol. 17, pp. 1–12, 2020.

[18] T. Ji, C. Wang, and L. Xie, "Towards real-time semantic RGB-D SLAM in dynamic environments," 2021, https://arxiv.org/abs/2104.01316.

[19] A. Li, J. Wang, M. Xu, and Z. Chen, "Dp-slam: a visual slam with moving probability towards dynamic environments," *Information Sciences*, vol. 556, pp. 128–142, 2021.

[20] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[21] M. S. Bahraini, M. Bozorg, and A. B. Rad, "Slam in dynamic environments via ml-ransac," *Mechatronics*, vol. 49, pp. 105–118, 2018.

[22] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 2017.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, June 2016.

[24] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, June 2018.

[26] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 573–580, Vilamoura-Algarve, Portugal, October 2012.

[27] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proceedings of the European conference on computer vision*, Graz, Austria, May 2006.

[28] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–221, 2005.

[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, July 2017.

[30] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the International conference on machine learning PMLR*, Lille, France, July 2015.

[31] Z. Zhang, X. Zhang, C. Peng, X. Xue, and J. Sun, "Exfuse: enhancing feature fusion for semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.

[32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2564–2571, Barcelona, Spain, November 2011.

[33] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[34] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: fast retina keypoint," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 510–517, Providence, RI, USA, June 2012.

[35] P. L. Rosin, "Measuring corner properties," *Computer Vision and Image Understanding*, vol. 73, no. 2, pp. 291–307, 1999.

[36] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, "Bundle adjustment—a modern synthesis," in *Proceedings of the International workshop on vision algorithms*, Corfu, Greece, September 1999.

[37] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[38] T. Y. Lin, M. Maire, S. Belongie et al., "Microsoft coco: common objects in context," in *Proceedings of the European conference on computer vision*, Zurich, Switzerland, September 2014.

[39] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, September 2018.

[40] F. Milletari, N. Navab, and S. A. Ahmadi, "V-net: fully convolutional neural networks for volumetric medical image segmentation," in *Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV)*, pp. 565–571, Stanford, CA, USA, October 2016.

[41] M. C. Potter and E. I. Levy, "Recognition memory for a rapid sequence of pictures," *Journal of Experimental Psychology*, vol. 81, no. 1, pp. 10–15, 1969.