MOR-SLAM: A New Visual SLAM System for Indoor Dynamic Environments Based on Mask Restoration

Chengzhi Yao , Lei Ding * , Yonghong Lan

Posted Date: 21 August 2023

doi: 10.20944/preprints202308.1419.v1

Keywords: Visual SLAM, repaired mask, dynamic environments, ORB-SLAM2

Article

MOR-SLAM: A New Visual SLAM System for Indoor Dynamic Environments Based on Mask Restoration

Chengzhi Yao [1], Lei Ding [1,*] and Yonghong Lan [2]

[1] College of Computer Science and Engineering, Jishou University, Jishou 416000, China
[2] College of Automation and Electronic Information, Xiangtan University
* Correspondence: dinglei@jsu.edu.cn

Abstract: The traditional Simultaneous Localization and Mapping (SLAM) systems are based on the strong static assumption, and their performance will degrade significantly due to the presence of dynamic objects located in dynamic environments. To decrease the effects brought by the dynamic objects, based on ORB-SLAM2 system framework, a novel dynamic semantic SLAM system called MOR-SLAM is presented using mask repair method, which can accurately detect dynamic objects and realize high-precision positioning and tracking of the system in dynamic indoor environments. First, an instance segmentation module is added to the front end of ORB-SLAM2 to distinguish dynamic and static objects in the environment and obtains a preliminary mask. Next, to overcome the under-segmentation problem in instance segmentation, a new mask inpainting model is proposed to ensure that the integrity of object masks, which repairs large objects and small objects in the image with depth value fusion method and morphological method respectively. Then, a reliable basic matrix can be obtained based on the above repaired mask. Finally, the potential dynamic feature points in the environment are detected and removed through the reliable basic matrix, and the remaining static feature points are input into the tracking module of the system to realize the high-precision positioning and tracking in dynamic environments. The experiments on the public TUM dataset show that, compared with ORB-SLAM2, the MOR-SLAM improves the absolute trajectory accuracy by 95.55%. In addition, compared with DynaSLAM and DS-SLAM on the high-dynamic sequences (fr3/w/rpy and fr3/w/static), the MOR-SLAM improves the absolute trajectory accuracy by 15.20% and 59.71%, respectively.

Keywords: visual SLAM; repaired mask; dynamic environments; ORB-SLAM2

1. Introduction

Simultaneous Localization and Mapping (SLAM)[1] is a system that integrates computer vision and robotics, and its purpose is to realize autonomous navigation and map construction of robots in unknown environments. Today SLAM systems have been widely used in autonomous driving[2], indoor navigation[3], AR[4], industrial automation[5] and other fields. The main purpose of visual SLAM is to perceive the environment around the robot through machine vision algorithms, and determine the position of the robot and the structure of the environment based on the perception results. By continuously modeling the environment, the robot can obtain more precise location and map information to better complete autonomous navigation and task execution. Since 2007, vSLAM have achieved satisfactory results, such as ORB-SLAM2[6] based on feature point method, DSO[7] and LSD-SLAM[8] based on direct method, etc. However, these visual SLAM algorithms are all built on the premise of a static environment, and dynamic objects, such as walking people, moving chairs, and other objects, inevitably appear in the real environment. Running these SLAM algorithms in a dynamic environment can seriously affect the accuracy and the system's robustness. Therefore, how to improve the stability and reliability of SLAM systems in dynamic environments has become the focus of current research.

As a classical vSLAM system, ORB-SLAM2 mainly uses the RANSAC algorithm to filter outliers. PTAM[9] mainly uses the robust kernel function to deal with the extremely small moving objects in

 doi:10.20944/preprints202308.1419.v1

the environment. However, when there are large areas or high numbers of moving objects in the environment, there will be too few detectable feature points in the image, which result in inaccurate camera pose estimation. With the recent development of artificial intelligence[10] in diagnosis[11], duplicate checking[12], optimization[13,14], manipulator control[15–17], and robots[18,19] etc. As an important branch of artificial intelligence, deep learning has made major breakthroughs in image recognition and natural language processing. Many researchers have tried to combine deep learning with SLAM and add a segmentation network to the front end to reduce the effect of predefined dynamic object pairs. Then many semantic dynamic SLAM systems have been proposed to deal with dynamic objects in dynamic environments. However, many SLAM systems remove the feature points on the object mask obtained by the segmentation network and ignore the problem that the segmented object mask cannot wholly cover the moving object. So, some dynamic feature points will still leak into the environment through the dynamic objects are removed, and the estimated camera pose error will be affected by the leaked dynamic feature points.

Based on the most classic ORB-SLAM2 system, this paper proposes a dynamic semantic SLAM system called MOR-SLAM which can work in an indoor dynamic environment. Considering that Mask R-CNN[20] can form the bounding box and dynamic objects mask simultaneously, we use the Mask R-CNN as the front end of MOR-SLAM for instance segmentation to obtain the semantic information of dynamic objects in the environment. At the same time, in order to solve the incomplete problem for the segmented object mask, MOR-SLAM adopts a new mask repair model, which uses the morphological method to repair the small segmented object mask, and uses a fusion method that combined mask map with depth map to improve the big segmented object mask, respectively. Then, after removing the feature points in the mask area, use the feature points outside the mask area are used to calculate the initial camera pose and use the obtained camera pose to construct epipolar constraints. So the potential dynamic feature points will be removed through the above method. Finally, the remaining static feature points will be used to compute the system positioning and mapping.

The main contributions of this paper are as follows:

1 A new repairing scheme for small dynamic object masks is proposed. The morphological method is used for repairing when the mask area is relatively small.
2 A new repairing scheme for large dynamic object masks is proposed. The depth value fusion method is used for repairing when the mask area is relatively large.
3 A novel and efficient dynamic semantic SLAM system (MOR-SLAM) is proposed based on ORB-SLAM2, which reduces the interference of dynamic objects in dynamic environments by combining instance-level segmentation networks and multi-view geometry techniques. The experiment on the TUM RGB-D dataset[21] shows that the system can operate stably in a highly dynamic environment and significantly improve the accuracy of the camera trajectory.

The rest of the paper is organized as follows: Section 2 presents the related work. Section 3 presents the overall pipeline of our method and the implementation details of the MOR-SLAM system. Section 4 presents and analyzes the experimental results of our system on the TUM public dataset. Section 5 gives the summary and outlook for this work.

2. Related work

The existing visual SLAM schemes are mainly divided into three categories: geometry-based methods, optical flow-based methods, and deep learning-based methods.

2.1. Geometry-based methods

The three-dimensional feature points in the space will satisfy the relationship of geometric constraints under the assumption of a static environment[22]. Then the multi-view geometric constraints can be used to segment the environment's dynamic and static feature points. However, not only dynamic feature points but also reprojection errors, intensity differences, and grayscale differences will violate geometric constraints in static environments.

 doi:10.20944/preprints202308.1419.v1

This method was first proposed by Kundu et al.[23] by defining two geometric constraints to construct a fundamental matrix to extract the dynamic feature points. The first one is the epipolar line constraint, which requires the matching point of a feature point in the subsequent image frame to locate on its corresponding epipolar line. If the location of the matching point of the feature point is too far away from the epipolar line, it is likely to be a dynamic feature point. The second constraint is the vector constraint(FVB), which concludes that the feature points located outside the set boundary are likely to belong to dynamic objects when the feature points move along the epipolar line between two consecutive frames. Finally, the Bayesian recursive filter determines whether the point is a dynamic feature point. Zou and Tan et al.[24] calculated the size of the reprojection error of the feature points from the previous frame to the current frame and compared it with the threshold to distinguish static and dynamic feature points. Sun et al.[25] calculated the intensity difference between two segmentation, and completed the classification of the pixels, the dynamic feature from the environment.

Furthermore, some SLAM systems perform simultaneous reconstruction of static background maps. Newcombe et al. [26] proposed the first SLAM system using RGBD cameras for dense mapping. The system uses the depth map to obtain point cloud and normal vector coordinates, then uses the ICP method to solve the camera pose. The TSDF value is updated according to the camera position and attitude. The surface is estimated according to the TSDF value.

Since all residual calculations are part of the standard vSLAM scheme, and there is no additional computational burden during execution, the geometry-based scheme has better real-time performance. However, the determination of dynamic-static properties is only determined by the geometric errors, and cannot handle the case of temporary stops of moving objects and distinguish between residuals caused by moving objects and residuals caused by mismatching. Both of these situations will lead to high geometric errors, so it is difficult to design a more complex and robust algorithm to maintain the stability of vSLAM in the environment only using the geometry algorithm.

2.2. Optical flow-based methods

The optical flow-based method uses the pixel transformation between two adjacent frames of images to perform motion estimation[27]. It can represent the motion field in the image, so it can be used to segment moving objects.

Derome et al. [28,29] use the residual between the predicted image and the binocular camera observation image to calculate the optical flow, and use the estimated camera pose to predict the previous frame image from the current frame through time backward processing. Finally, the residual of the points is used to detect dynamic objects. Fang et al. [30] improved the optical flow method by introducing the Kalman filter, feature point matching technology, and uniform sampling strategy, which significantly improved the ability to detect and track dynamic objects. Although this method is computationally faster than other methods, it is less accurate. Wang et al. [31] completed the segmentation of moving objects through optical flow calculation, sparse point trajectory clustering, and densification operations in sequence, and used model selection to deal with over-segmentation or under-segmentation.

There are also methods that utilize scene flow to segment moving objects. Long et al. [32] first divides the scene flow with a grid, and then uses an adaptive threshold algorithm to detect dynamic objects in the environment. On this basis, a deep average clustering segmentation method is used to find potential dynamic targets. Finally, combining the results of grid segmentation with depth average clustering segmentation, dynamic objects are accurately found, and feature points in the dynamic object area are removed. Alcantarilla et al.[33] calculate the modulus of the 3D motion vector in the scene flow, and use the residual motion likelihood to judge the dynamic-static properties of the object. If the residual is low, the feature point is likely to belong to a static object. However, this method takes a long time to calculate, and its real-time performance is difficult to meet people's needs. Zhang et al.[34] proposed a dense fusion RGB-DSLAM scheme based on optical flow semantics. The system uses

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

Pwc-Net[35] to calculate the optical flow of two consecutive frames, and then combines the intensity and depth information between the two frames to estimate camera pose. Next, optical flow and pose are used to calculate 2D scene flow. Finally scene flow is used to segment dynamic objects in the environment.

Optical flow-based techniques have similar properties to geometry-based methods, which can accurately detect and recognize the position and motion state of moving objects without knowing scene information, are sensitive to slight motion, and work in real-time. However, optical flow-based methods work under the assumption of constant brightness, and they are susceptible to lighting effects. Like geometry-based schemes, the optical-flow methods also need to struggle with degenerative motion. For example, the moving object with a small motion vector is easily seen as a part of the static background when the object moves in a direction and speed close to the camera along the epipolar plane.

2.3. Deep Learning-based methods

With the great development of deep learning in the field of image processing in the past ten years, many dynamic SLAM systems are accustomed to using semantic segmentation for preprocessing at the front end, and using prior semantic information to assist the system in obtaining a more accurate camera pose estimation.

In the prior semantic information acquisition module, most visual SLAM solutions in dynamic environments use the existing mature target detection and semantic segmentation network frameworks for initial dynamic area division, and the most used network architectures include Mask R-CNN, SSD citeliu2016ssd, SegNet[36] and other segmentation networks. Yu et al. [37] added semantic segmentation and dense octree mapping modules based on the ORB-SLAM2 system. The system first filtered out the main moving target pedestrians in the environment through the semantic segmentation module, and then used epipolar constraints to perform a mobile consistency check. If some dynamic feature points are detected on an object, all the feature points on the object will be removed, and the remaining static feature points will be used for camera pose estimation and dense 3D semantic octree map construction. Dyna-SLAM proposed by Bescos et al.[38] uses the instance segmentation network MaskRCNN to segment the moving objects with prior knowledge, uses the proposed low-cost tracking to calculate the initial camera pose, and uses multi-view geometry to detect potential moving objects in the environment (such as a chair being moved by someone, etc.).

The Dynamic-SLAM proposed by Xiao et al. [39] is based on the SSD target detection algorithm, and proposes a missing detection compensation algorithm based on the basic motion model, which can calculate the speed of moving objects and greatly improve the Recall Rate of detection. Liu et al. [40] proposed a semantic-based real-time dynamic vSLAM algorithm called RDS-SLAM on the basis of ORB-SLAM3[41], which greatly improved the real-time performance of the algorithm. Based on the RDS-SLAM, RDMO-SLAM is proposed combing optical flow estimation and MaskRCNN to obtain semantic information[42]. In order to improve the segmentation accuracy on the object boundary, Xu et al. [43] used the region-growing algorithm combined with the prior information of semantic segmentation to obtain the actual edge of the dynamic object depth image, which improved the incompleteness of the mask edge to a certain extent. In order to overcome the problem of incomplete segmentation in instance segmentation, Xie et al. [44] used the K-means clustering method to segment the depth image into K clusters, and a similarity test method is designed to find the clustering of human beings. Then the clustering block with the largest similarity value is considered as the human clustering block, and the clustering block is binarized to obtain an improved mask.

In summary, the method combined with deep learning has great potential in dynamic SLAM. Most methods combined with deep learning use target or semantic segmentation networks to detect and segment objects and determine their motion state according to the attributes of the object category. However, these methods are crude. However, these methods are simple. First of all, compared with the original object outline, the segmented mask by the semantic segmentation network is incomplete,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

which will cause some dynamic object information to leak into the environment and have a certain impact on the pose estimation and the map construction. In addition, some scholars have noticed the incomplete mask problem and proposed some solutions. However, these methods simply use the same way to solve the problem of incomplete masks and ignore the impact of different object sizes, resulting in unsatisfactory repair results. Therefore, the size of the object area should be considered when solving the incomplete mask problem.

3. System Design

3.1. Overview of the MOR-SLAM framework

A MOR-SLAM framework is proposed based on ORB-SLAM, and it contains the following modules: tracking, local mapping, closed-loop threads, instance segmentation, mask repair, dynamic feature point detection, and multi-view geometry threads.

The flow of the entire system is shown in Figure 1. The system first inputs the RGB images captured by the camera to the feature extraction and instance segmentation threads, respectively. In the input images, we divide objects into three categories: static objects, dynamic objects, and potential dynamic objects. Among the three categories, dynamic objects mainly refer to human beings, and potential dynamic objects refer to those objects that are usually stationary and move due to other factors at a certain time (such as chairs moved by people, etc.). The system extracts all the image's static and dynamic feature points in the feature extraction thread.

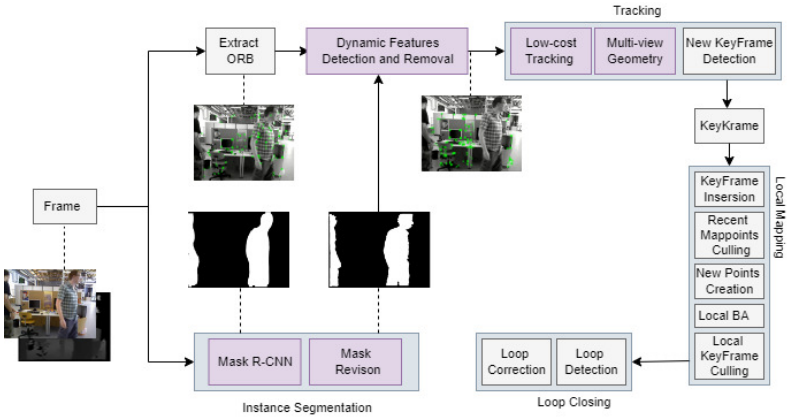


Figure 1. Overview of MOR-SLAM

At the same time, in the instance segmentation thread, MaskRCNN is used to obtain the original mask image corresponding to the object of the image, and the objects are divided into large and small objects. Then, different repair methods for objects of different sizes are used to repair their corresponding mask in the Mask Revision module. Next, the repaired mask image is input into the dynamic feature point detection module to remove all the feature points in the mask area, and the remaining feature points are input into the tracking thread. The low-cost tracking algorithm proposed in[38] calculates the initial pose, and the multi-view geometry method detects and removes potential dynamic feature points in the image. The remaining static feature points are input into the tracking thread to position and map the SLAM system.

3.2. Segmentation of moving objects

Mask R-CNN is an object detection model based on Faster R-CNN, which can simultaneously predict the category of the object and the precise mask. Compared with the traditional target detection model, Mask R-CNN can not only detect the object's position but also obtain the precise contour information of the object. So it has a wide range of applications in image processing,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

instance segmentation, face recognition, and other fields. To accurately detect dynamic objects in the environment, Mask R-CNN based on TensorFlow [46] is used to perform pixel-level semantic segmentation on images, and the corresponding network framework is shown in Figure 2.

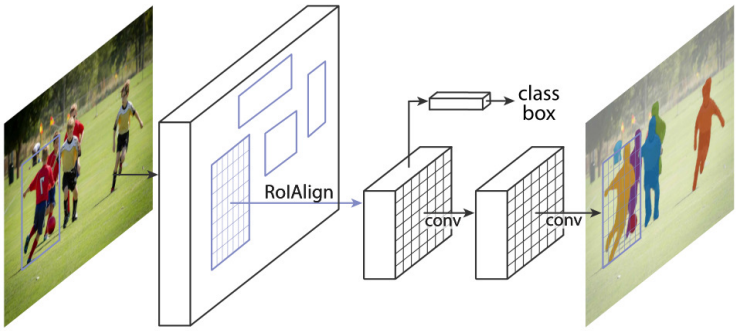


Figure 2. The framework of Mask R-CNN

The input of Mask Rcnn is a three-channel RGB image, and the output of the network is a matrix of size m × n × l, where l represents the appropriate amount of objects in the image. For each output channel l, potential dynamic objects (such as people, cars, bicycles, boats, sheep, cows, horses, birds, cats, dogs, etc.) in the environment are detected by Mask R-CNN and a binary mask is obtained, and a segmentation of all dynamic objects appearing in a scene image can be obtained by merging all channels into one.

3.3. Mask repair

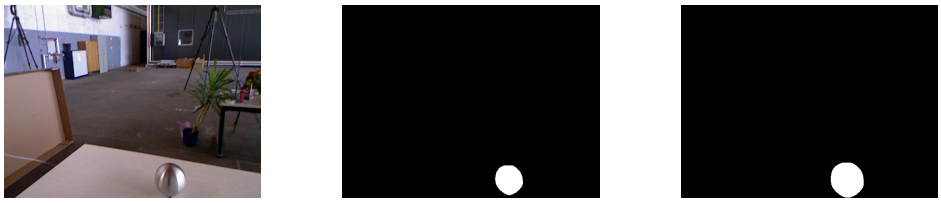
The object mask segmented by Mask R-CNN usually has the problem of incompleteness, which may make part of the information of the dynamic object leak into the environment and affect the accuracy of pose estimation and map construct. To this end, we propose a new mask repair model to solve the incompleteness of the mask. The morphological method and the depth value fusion method are used to repair the mask for the small objects and the large objects, respectively.

3.3.1. Mask repair for small objects

When the object is relatively small, the corresponding area in the depth map often has an incomplete or hollow parts, so the depth map is not appropriate to improve the mask of small objects is not appropriate. Therefore, the morphological method is more suitable for improving the incompleteness of small object masks.

As shown in Figure 3(a), a small ball is slowly rolling on the table, and Figure 3(b) is the initial mask image of the RGB image after Mask R-CNN segmentation. It can be seen from the figure that although the segmented mask image is close to the outline of the small ball, its edge part is still quite different from the complete object outline. To improve the mask's incompleteness, the median filter is first used to process the mask, which aims to eliminate the potential speckle noise in the small object mask and protect the edge part of the mask. Then, the erosion and dilation operation is used to make the edge part of the mask closer to the original object as much as possible. Finally, the improved small object mask image is shown in Figure 3(c). It can be seen from the figure that the inpainted mask is closer to the original object shape than the original mask.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1



(a) rgb (b) original mask (c) Improved mask

Figure 3. Comparison of mask repair for small objects

3.3.2. Mask repair for large objects

The Morphological method cannot effectively solve the incompleteness of relatively large mask, and even complicate the incompleteness of the mask. Therefore the depth map is used to repair the mask.

A threshold τ is given, and the objects are treated as large objects when the ratio of the segmented object mask area to the entire RGB image area is greater than τ. On the contrary, the objects are treated as small objects. The semantic information of the large object obtained by the segmentation network is used to obtain the object label, such as Obj_i, where i ∈1,2,3...,n. Then each depth value of the depth map area $depth_{mask}^{Obj_i}$ corresponding to its mask area is added up to obtain the depth value set $depth^{Obj_i}$ of the object. Maximum, minimum, and average depth values in the depth set can be respectively represented as follows:

$$d_{min} = \text{GetMin}\left(depth^{Obj_i}\right) \tag{1}$$

$$d_{max} = \text{GetMax}\left(depth^{Obj_i}\right) \tag{2}$$

$$d_{avg} = \text{GetAvg}\left(\frac{\sum_{i=1}^{n} depth_i^{Obj_i}}{n}\right) \tag{3}$$

here n represents the number of pixels, and $depth_i^{Obj_i}$ represents the depth value of ith pixel. To obtain the range of depth value fusion, e_{max} and e_{min} are first obtained by subtracting the average value d_{avg} from the maximum d_{max} and the minimum d_{min}, respectively, where e_{min} takes the absolute value. Then, the range of depth value fusion can be obtained from formulas 6 and 7.

$$e_{max} = d_{max} - d_{avg} \tag{4}$$

$$e_{min} = \left|d_{min} - d_{avg}\right| \tag{5}$$

$$\tau_{max} = \frac{e_{max}}{d_{avg}} d_{max} \tag{6}$$

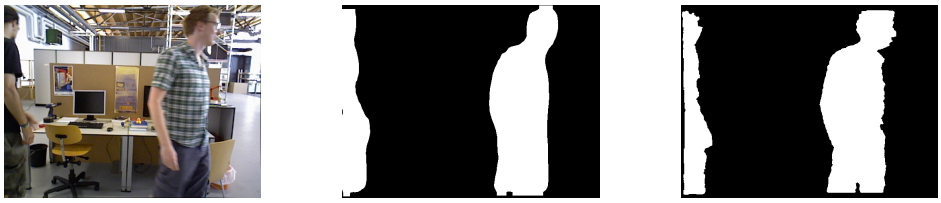
$$\tau_{min} = \frac{e_{min}}{d_{avg}} d_{min} \tag{7}$$

For more efficient fusion, a square slider with a size of 2*2 is used to traverse the depth image. When the slider traverses to the edge area, the mask depth map area is expanded by adding points whose depth value variation stays within the range of depth value of target objects to the mask depth area.

$$\begin{cases} d_{min} - \tau_{min} \le d(u,v) \le d_{max} + \tau_{max} & (u,v) \in depth_{mask} \\ else & (u,v) \notin depth_{mask} \end{cases} \tag{8}$$

where $d_{u,v}$ represents the depth value on the (u,v) pixel. Finally, the original mask is replaced by the fused depth region represented as a binary image. Algorithm 1 gives the specific steps of mask restoration. As shown in Figure 4, the optimized mask is closer to the contour of the object region than the original mask.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1



(a) rgb (b) original mask (c) Improved mask

Figure 4. Comparison of mask repair for large objects

Algorithm 1 Mask improvement algorithm

Input: Current depth image dep_c,original Mask $Mask_o$,mask size threshold τ,rgb image size $Area_{img}$
Output: revised mask

Label the objects $Obj_1, ..., Obj_n$ in $Mask_o$;
for each mask $Mask_o^{Obj_i}$ of Obj_i **do**
  Obtain a depth map $depth_{mask}^{Obj_i}$ of the corresponding area of the mask $Mask_o^{Obj_i}$
  Calculate the area $Are_{mask}^{Obj_i}$ of $Mask_o^{Obj_i}$
  if $Are_{mask}^{Obj_i} \geq \tau$ **then**
    Record depth values $depth^{Obj_i}$ in $Mask^{Obj_i}$
    $d_{min} = \text{GetMin}\left(depth_n^{Obj_i}\right)$;
    $d_{max} = \text{GetMax}\left(depth_n^{Obj_i}\right)$;
    $d_{avg} = \text{GetAvg}\left(\frac{\sum_{i=1}^{n} depth_i^{Obj_i}}{n}\right)$;
    $e_{max} = d_{max} - d_{avg}$,$e_{min} = \left|d_{min} - d_{avg}\right|$;
    $\tau_{max} = \frac{e_{max}}{d_{avg}} d_{max}$,$\tau_{min} = \frac{e_{min}}{d_{avg}} d_{min}$;
    for each $point(u, v)$ in $depth_area^i$ **do**
      if $d_{min} - \tau_{min} \leq d(u, v) \leq d_{max} + \tau_{max}$ **then**
        $(u, v) \in depth_{mask}^i$
      else
        $(u, v) \notin depth_{mask}^i$
      end if
    end for
    Represent the fused depth region with a binary image
  else
    Repair $Mask_o^{Obj_i}$ with morphological methods
  end if
end for
return result

3.4. Epipolar Geometric Constraints Based on Mask Repair

Mask R-CNN can only remove the feature points in the predefined dynamic object mask area, and there are some potential moving targets in the environment (such as chairs that may be moved by people, etc.). So it is necessary to further detect these objects to determine whether they are dynamic objects.

Considering the real-time nature of the system, we adopt lightweight and low-cost pose estimation in DynaSLAM. Based on the previous mask repair, for the feature points outside the mask area of the dynamic object in one frame, the sparse pyramid Lucas-Kanade optical flow [45] algorithm is used to track these feature points and obtain the corresponding feature points in the next frame. Based on epipolar geometry constraint, only static points can satisfy epipolar constraints. Figure 5 shows the relationship between corresponding image points in two consecutive frames of images.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

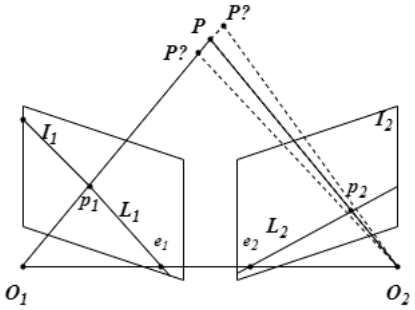


Figure 5. Epipolar geometry schematic

Let points p_1 and p_2 be the imaged points of point P in two continuous frames respectively, and L_1 and L_2 be the epipolar lines of the two corresponding planes. The normalized coordinates of p_1 and p_2 are expressed as:

$$p_1 = [u_1 \quad v_1 \quad 1] \quad , \quad p_2 = [u_2 \quad v_2 \quad 1] \tag{9}$$

where u_1, v_1, u_2, and v_2 are respectively the pixel coordinates of the matched key points. The correspondence between p_1 and p_2 is as follows:

$$p_2^T F p_1 = 0 \tag{10}$$

The epipolar line L_1 can be calculated by the following equation:

$$L_1 = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = F p_1 = F \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} \tag{11}$$

Whether a point tracked by the optical flow is a dynamic feature point can be determined by calculating the distance from the feature point to the epipolar line. The distance D from the point p_1 to the epipolar line L_2 is expressed as:

$$D = \frac{|p_2^T F p_1|}{\sqrt{\|X\|^2 + \|Y\|^2}} \tag{12}$$

If D is greater than or equal to the set threshold, the point will be determined as a dynamic feature point and be removed from the image. Conversely, if D is smaller than the threshold, the point will be determined as a static feature point and retained. The process of using epipolar geometry to detect dynamic points is shown in Algorithm 2, where κ is the set distance threshold.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

Algorithm 2 Dynamic Points Detection Algorithm

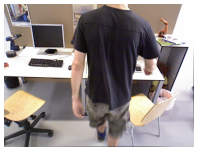
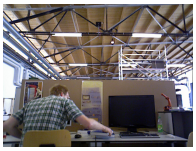
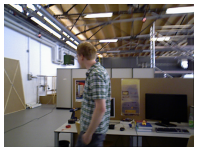
Input: Previous frame F_1,Current frame F_2,Previous frame′ feature point p_1;
Output: The set of dynamic points

Current frame′ feature point p_2 = CalcOpticalFlowPysLk(F_1, F_2, p_1);
F=FindFundamentalMatrix(p_1, p_2);
for each matched pairs p_1,p_2 **do**
    L_1=FindEpipolara(p_1, F)
    D=CalcDistanceFromEpipolarLine(p_2, L_1)
    if D>κ **then**
        Append p_1 to S
    end if
end for
return result

3.5. Background Repair

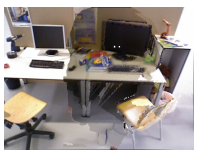
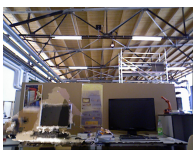
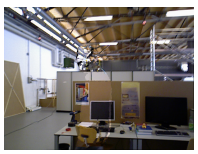
After removing the dynamic objects in the image, the static information in the previous image is used to patch the background occluded by the dynamic objects. Finally, we can synthesize a real image after removing dynamic objects. This framework includes static environment synthesis and plays a vital role in applications, such as SLAM back-end loop closure detection and mapping or virtual reality construction.

Due to the known positions of the current frame and the previous frame, the RGB and the depth image channels of all keyframes before the current frame (the last 20 in the experiment) are projected onto the corresponding channels of the current frame. Some seams are black in the synthesized images containing only static information because the scene parts in these blank areas did not appear in the previous keyframes. Alternatively, even if they did, no valid depth information corresponds to the scene. These gaps cannot be geometrically reconstructed and require more elaborate repair techniques. Figure 6 shows three background inpainting maps for different dynamic sequences in the TUM RGB-D public dataset. The dynamic objects have been segmented and removed in Figure 6, and most of the original dynamic object regions have been embedded with static background information.


(a) Original RGB images

(b) Original depth image Yee


(c) Inpainted RGB images

(d) Inpainted depth image

Figure 6. Background inpainting results. (**a**) we show three RGB input frames, and in (**c**) we show the output of our system, in which all dynamic objects have been detected and the background has been reconstructed. (**a**) and (**d**) show respectively the depth input and output, which has also been processed.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

4. Experimental results

This article selected four sets of highly dynamic sequences (fr3/walk/) from the public TUM RGBD datasets. To verify the stability and robustness of MOR-SLAM in the dynamic environment, the original ORB-SLAM2, DynaSLAM, and DS-SLAM were employed for comparison with MOR-SLAM, respectively. In the highly dynamic sequence, two people shuttled back and forth in the laboratory, occasionally sitting on a chair; in the low dynamic sequence, two people just sat at a table to talk, gesturing and sitting on the chair with slight movement. The experiment is conducted on a computer with Intel core i9-12900H, GeForce RTX 3070Ti GPU.

4.1. Quantitative Evaluation of Trajectory Error

The absolute trajectory error (ATE) and relative trajectory error (RPE) are used to quantitatively evaluate our system. The absolute trajectory error is calculated as the difference between the real value of the camera pose and the estimated value of the SLAM system, which is suitable for evaluating the performance of the SLAM system. The relative trajectory error is used to calculate the difference between the camera pose's real value and the SLAM system's estimated value at two same time stamps. It can be understood as a real-time comparison between the real value of the pose and the estimated value, which is suitable for estimating the drift of the system. Root mean square error (*RMSE*) and standard deviation (S.D.) are used to quantify ATE and RPE, and *RMSE* and *S.D.* are calculated as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(P_i - T_i)^2} \tag{13}$$

$$S.D. = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{14}$$

where P_i, T_i, and μ represent the estimated pose, the real pose, and the mean of $x_1, x_2, ...x_N$, respectively. We use the following values to measure the improvement effect of MOR-SLAM compared with ORB-SLAM2:

$$\sigma = (1 - \frac{\epsilon}{\eta}) \times 100\% \tag{15}$$

where σ, η, ϵ represent the improvement result, the size of the error value of ORB-SLAM2, and the experimental result of MOR-SLAM, respectively. The value of σ can well reflect the improvement effect of MOR-SLAM compared with ORB-SLAM2.

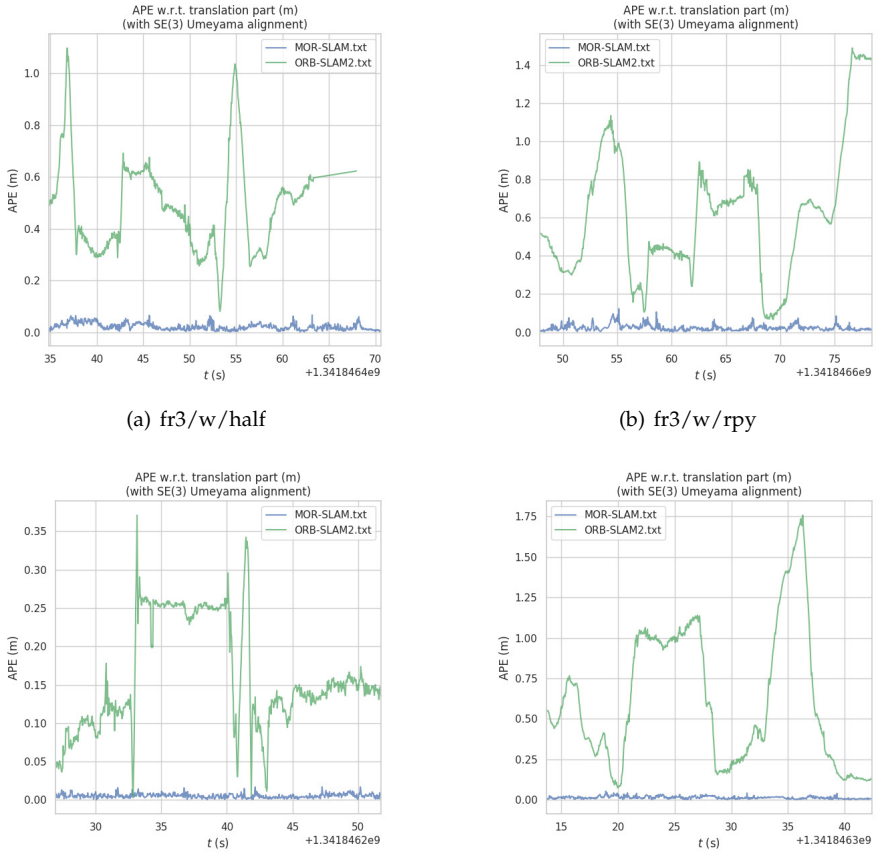
Table 1 presents the ATE comparison results among MOR-SLAM, MOR-SLAM, Dyna-SLAM, and DS-SLAM. MOR-SLAM achieves the best results on fr3/w/half, fr3/w/rpyfr3, and fr3/w/static sequences. On the fr3/w/xyz sequence, Dyna-SLAM achieves the best value on *RMSE*, while MOR-SLAM has a better effect on *S.D.*. On the low dynamic sequence fr3/s/xyz, ORB-SLAM2 has the best result.

Table 1. Absolute Track Error Results(ATE)[m]

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		MOR-SLAM		Improvements	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	0.4543	0.2524	0.0296	0.0157	0.0303	0.0159	**0.0257**	**0.0128**	94.34%	94.93%
fr3/w/rpy	0.5391	0.2283	0.0354	0.0190	0.4442	0.2350	**0.0288**	**0.0161**	94.65%	92.95%
fr3/w/static	0.3194	0.1819	0.0068	0.0032	0.0081	0.0033	**0.0060**	**0.0027**	98.12%	98.18%
fr3/w/xyz	0.7521	0.4712	**0.0164**	0.0086	0.0247	0.0161	0.0166	**0.0082**	95.09%	97.02%
fr3/s/xyz	**0.0092**	**0.0044**	0.0127	0.0060	0.0115	0.0056	0.2288	0.0989	-23.86%	-21.47%

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

In addition, we used the evo tool (https://github.com/MichaelGrupp/evo) to draw a targeted ATE comparison between ORB-SLAM2 and MOR-SLAM on four highly dynamic sequence data sets. The ATE comparison results between MOR-SLAM and ORB-SLAM2 are shown in Figure 7. We use SE3(3) to align the motion trajectory with the ground truth trajectory. As can be seen from Figure 7, the pose error of MOR-SLAM is significantly lower than that of ORB-SLAM2.



(a) fr3/w/half (b) fr3/w/rpy

(c) fr3/w/static (d) fr3/w/xyz

Figure 7. Comparison of ATE between ORB-SLAM2 and MOR-SLAM on four highly dynamic data sets

Table 2-3 show the results of translation and rotation RPE. From the results of the translation RPE, Dyna-SLAM works best on fr3/w/half sequences, while MOR-SLAM has better values than other SLAM systems on fr3/w/rpy and fr3/w/static sequences. The results of Table 2-3 show that Dyna-SLAM and MOR-SLAM has the best results on *RMSE* and *S.D.* on the fr3/w/xyz sequence, respectively, and ORB-SLAM2 still works best on the low dynamic fr3/s/xyz sequence. Additionally, MOR-SLAM performs better on rotation RPE on highly dynamic sequence datasets than the other two dynamic SLAMs.

Table 2. Translation relative pose error results(RPE)[m]

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		MOR-SLAM		Improvements	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	0.3216	0.2629	**0.0284**	**0.0149**	0.0297	0.0152	0.0362	0.0166	88.74%	93.68%
fr3/w/rpy	0.3880	0.2823	0.0448	0.0262	0.1503	0.1168	**0.0410**	**0.0211**	89.43%	92.52%
fr3/w/static	0.1928	0.1773	0.0089	0.0044	0.0102	0.0038	**0.0087**	**0.0037**	95.48%	97.91%
fr3/w/xyz	0.4834	0.3663	**0.0217**	0.0119	0.0333	0.0229	0.0237	**0.0109**	97.79%	97.02%
fr3/s/xyz	**0.0117**	**0.0057**	0.0142	0.0073	0.0133	0.0069	0.3220	0.1384	-26.52%	-23.28%

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

Table 3. Rotation Relative Pose Error Results(RPE)[m]

Sequences	ORB-SLAM2		Dyna-SLAM		DS-SLAM		MOR-SLAM		Improvements	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
fr3/w/half	6.6515	5.3990	**0.7842**	0.4012	0.8142	0.4101	0.8479	**0.3589**	87.25%	93.35%
fr3/w/rpy	7.5906	5.4768	0.9894	0.5701	3.0042	2.3065	**0.8702**	**0.4385**	88.53%	91.99%
fr3/w/static	3.5991	3.2457	0.2612	0.1259	0.2690	0.1215	**0.2594**	**0.1106**	92.79%	96.59%
fr3/w/xyz	8.8419	6.6762	**0.6284**	0.3848	0.8266	**0.2826**	0.6363	0.3830	92.80%	94.26%
fr3/s/xyz	**0.4874**	**0.2532**	0.5042	0.2651	0.5044	0.2623	0.6655	0.3128	-36.54%	-23.53%

From the results obtained in Table 1-3, we can conclude that adding semantic information to the front end of SLAM in a highly dynamic environment can improve the system's positioning accuracy and pose estimation. DynaSLAM, DS-SLAM, and MOR-SLAM have all achieved good results on four highly dynamic environment sequences. Compared with ORB-SLAM2, average *RMSE* improvement values of ATE, translation RPE and rotation RPE of MOR-SLAM on highly dynamic sequences are 95.55%, 92.86% and 90.34%, respectively. In addition, compared with Dyna-SLAM and DS-SLAM, *RMSE* improvement values of ATE, translation RPE and rotation RPE of MOR-SLAM on some highly dynamic sequences (fr3/w/rpy and fr3/w/static) are 15.20% and 59.71%, respectively. It can be seen that MOR-SLAM can obviously improve the positioning accuracy of SLAM in a dynamic environment.

As shown in Table 4, to quantitatively compare the trajectory points tracked by four systems, we present the results of their successful tracking of trajectory points. Compared with ORB-SLAM2, Dyna-SLAM, and DS-SLAM on high dynamic sequences, MOR-SLAM has the highest number of tracking trajectory points. In practice, especially in long-term navigation, better coverage of track points is more important than a slight increase in accuracy. On highly dynamic sequences, the average successful rate of accurately tracking trajectory points using MOR-SLAM is 99.74%.

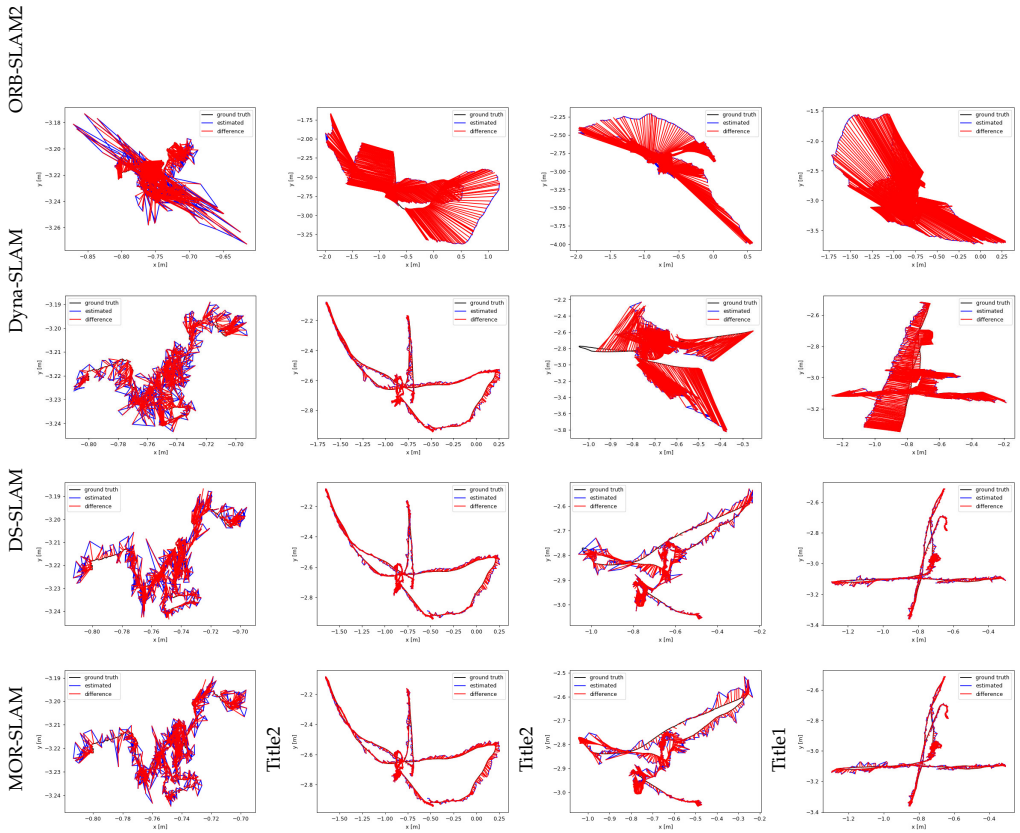
Table 4. Results of successfully tracking track points

Sequences	Total	ORB-SLAM2		Dyna-SLAM		DS-SLAM		MOR-SLAM	
		Tracked	Ratio	Tracked	Ratio	Tracked	Ratio	Tracked	Ratio
fr3/w/half	1021	942	92.26%	1011	99.02%	**1018**	**99.71%**	**1018**	**99.71%**
fr3/w/rpy	866	825	95.27%	711	82.10%	**864**	**99.77%**	**864**	**99.77%**
fr3/w/static	717	**714**	**99.58%**	696	97.07%	**714**	**99.58%**	**714**	**99.58%**
fr3/w/xyz	827	809	97.82%	757	91.54%	**826**	**99.88%**	**826**	**99.88%**

4.2. Qualitative Evaluation of Trajectory Error

Figure 8 compares the trajectory errors of the four SLAM systems on highly dynamic sequences. The black line shows the real trajectory of the camera, the blue line is the trajectory estimated by the SLAM system, and the red line shows the difference between the previous two SLAM systems. It can be seen from the figure that the camera trajectory estimated by the ORB-SLAM2 system in a highly dynamic environment has a large error compared with the camera trajectory itself, and the Dyna-SLAM and the MOR-SLAM systems can reduce this error very well. Among them, the MOR-SLAM system is better than the Dyna-SLAM system in the fr3/w/half, fr3/w/rpy and fr3/w/static sequences.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1



(a) fr3/w/static (b) fr3/w/half (c) fr3/w/rpy (d) fr3/w/xyz

Figure 8. This is the camera trajectory image. The black lines are the true trajectory, the blue lines are our estimated camera trajectory and and the red line represents the difference between the true and estimated trajectories. On the four datasets, our estimated trajectories are substantially around the ground truth trajectories.

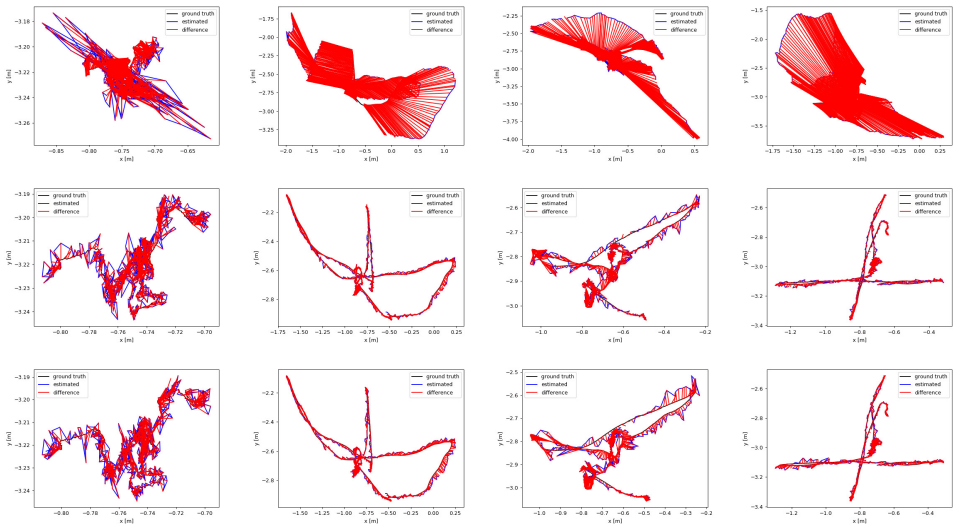
4.3. Ablation experiment

To verify the effectiveness of Mask R-CNN and mask repair on the SLAM system, this paper uses the original ORB-SLAM2 system without Mark R-CNN and mask repair as a baseline to complete the ablation experiments on the high dynamic sequence of the TUM RGBD dataset. The ATE is used for quantitative evaluation, and the quantitative and the qualitative results are given. The quantitative results are shown in the Table 5, and the qualitative results are shown in the Figure 9. From the results shown in Table 5 and Figure 9, we can find that the performance improvement in dynamic environments is obvious.

Table 5. Quantitative results of ablation experiments on the TUM RGBD dataset(ATE)[m]

	fr3/w/half		**fr3/w/rpy**		**fr3/w/xyz**		**fr3/w/static**	
	RMSE	**S.D.**	**RMSE**	**S.D.**	**RMSE**	**S.D.**	**RMSE**	**S.D.**
Baseline	0.0453	0.2524	0.5391	0.2283	0.7521	0.4712	0.3194	0.1819
Baseline+Mask R-CNN	0.0260	0.0129	0.0295	0.0170	0.0175	0.0091	0.0062	0.0029
Baseline+Mask R-CNN+Mask repair	**0.0257**	**0.0128**	**0.0288**	**0.0161**	**0.0166**	**0.0082**	**0.0060**	**0.0027**

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 21 August 2023 doi:10.20944/preprints202308.1419.v1

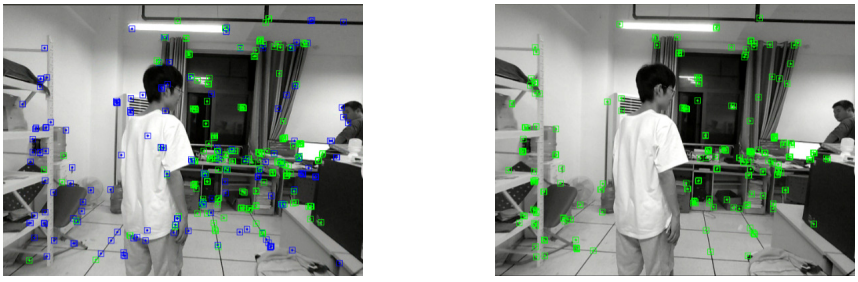


(a) fr3/w/static (b) fr3/w/half (c) fr3/w/rpy (d) fr3/w/xyz

Figure 9. Qualitative results of ablation experiments on the TUM RGBD dataset. The first row is the result of the original ORB-SLAM2 experiment. The second row is the experimental result after adding the Mask-RCNN module to the original ORB-SLAM2. The experimental results of adding Mask-RCNN and mask repair module simultaneously on the original ORB-SLAM2 are shown in the third row.

4.4. Run in real environment

To prove the MOR-SLAM's effectiveness, we conducted experiments on MOR-SLAM in a practical environment. In this experiment, a person walked around the scene. We used a handheld RealSense D455 camera to collect a large number of RGB images and their corresponding depth images of the surrounding environment. The size of RGB images and depth images are all 640 × 480. Figure 10 shows the experimental results of ORB-SLAM2 and MOR-SLAM in the practical environment. In Figure 10(a), ORB-SLAM2 extracts many features on walking people, while in Figure 10(b), MOR-SLAM removes the feature points on moving people, and almost all the extracted feature points fall on the static background.



(a) Results of ORB-SLAM2 (b) Results of our SLAM

Figure 10. Comparison of feature points extracted by ORB-SLAM2 and MOR-SLAM in the real environment

5. Conclusion

This paper proposes a deep learning-based dynamic semantic SLAM (MOR-SLAM) algorithm that works in indoor dynamic environments. The core idea of the algorithm is to use the Mask R-CNN instance segmentation network to extract semantic information in the environment. In addition, to solve the incompleteness of the segmented mask, the depth map is used to improve the segmented mask, and the improved mask can effectively cover the dynamic object area in the environment. Then

combined with multi-view geometry, the dynamic feature points in the environment are eliminated, and only the remained static feature points are used to estimate the camera's pose. This proposed method the positioning accuracy and stability of the SLAM system in a dynamic environment. To verify the effectiveness of the MOR-SLAM algorithm, we conducted experiments on the TUM RGB-D public dataset and real environments. The experimental results prove that our system can significantly improve the system's positioning accuracy compared with the ORB-SLAM2 system. Our system improves localization accuracy in highly dynamic sequences by 95.55% over the original ORB-SLAM2 system. In addition, using the original SLAM as the reference object, MOR-SLAM achieves a higher relative RMSE reduction than Dyna-SLAM and DS-SLAM systems.

Author Contributions: C.Y.: experiment preparation, data processing, and publication writing; L.D.: experiment preparation and publication writing; Y.L..: technology support, data acquisition, and publication review. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China under grants 61966014

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine* **2006**, *13*, 99–110.
2. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access* **2020**, *8*, 58443–58469.
3. Huang, H.; Gartner, G. A survey of mobile indoor navigation systems. *Cartography in Central and Eastern Europe* **2009**, pp. 305–319.
4. Azuma, R.T. A survey of augmented reality. *Presence: teleoperators & virtual environments* **1997**, *6*, 355–385.
5. Mistry, I.; Tanwar, S.; Tyagi, S.; Kumar, N. Blockchain for 5G-enabled IoT for industrial automation: A systematic review, solutions, and challenges. *Mechanical systems and signal processing* **2020**, *135*, 106382.
6. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics* **2017**, *33*, 1255–1262.
7. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *40*, 611–625.
8. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part II 13. Springer, 2014, pp. 834–849.
9. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. 2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE, 2007, pp. 225–234.
10. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *nature* **2015**, *521*, 436–444.
11. Zhou, K.Q.; Gui, W.H.; Mo, L.P.; Zain, A.M. A bidirectional diagnosis algorithm of fuzzy Petri net using inner-reasoning-path. *Symmetry* **2018**, *10*, 192.
12. Chen, C.F.; Zain, A.M.; Zhou, K.Q. Definition, approaches, and analysis of code duplication detection (2006–2020): a critical review. *Neural Computing and Applications* **2022**, pp. 1–31.
13. Chen, Z.; Francis, A.; Li, S.; Liao, B.; Xiao, D.; Ha, T.T.; Li, J.; Ding, L.; Cao, X. Egret Swarm Optimization Algorithm: An Evolutionary Computation Approach for Model Free Optimization. *Biomimetics* **2022**, *7*, 144.
14. Zhang, X.Y.; Zhou, K.Q.; Li, P.C.; Xiang, Y.H.; Zain, A.M.; Sarkheyli-Hägele, A. An Improved Chaos Sparrow Search Optimization Algorithm Using Adaptive Weight Modification and Hybrid Strategies. *IEEE Access* **2022**, *10*, 96159–96179.
15. Xiao, L.; Liao, B.; Li, S.; Zhang, Z.; Ding, L.; Jin, L. Design and analysis of FTZNN applied to the real-time solution of a nonstationary Lyapunov equation and tracking control of a wheeled mobile manipulator. *IEEE Transactions on Industrial Informatics* **2017**, *14*, 98–105.

16. Zhang, Y.; Qiu, B.; Liao, B.; Yang, Z. Control of pendulum tracking (including swinging up) of IPC system using zeroing-gradient method. *Nonlinear Dynamics* **2017**, *89*, 1–25.
17. Xiao, L.; Zhang, Z.; Li, S. Solving time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **2018**, *49*, 2210–2220.
18. Li, J.; Peng, H.; Hu, H.; Luo, Z.; Tang, C. Multimodal information fusion for automatic aesthetics evaluation of robotic dance poses. *International Journal of Social Robotics* **2020**, *12*, 5–20.
19. Peng, H.; Hu, H.; Chao, F.; Zhou, C.; Li, J. Autonomous robotic choreography creation via semi-interactive evolutionary computation. *International Journal of Social Robotics* **2016**, *8*, 649–661.
20. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
21. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. 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.
22. Saputra, M.R.U.; Markham, A.; Trigoni, N. Visual SLAM and structure from motion in dynamic environments: A survey. *ACM Computing Surveys (CSUR)* **2018**, *51*, 1–36.
23. Kundu, A.; Krishna, K.M.; Sivaswamy, J. Moving object detection by multi-view geometric techniques from a single camera mounted robot. 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2009, pp. 4306–4312.
24. Zou, D.; Tan, P. Coslam: Collaborative visual slam in dynamic environments. *IEEE transactions on pattern analysis and machine intelligence* **2012**, *35*, 354–366.
25. Sun, Y.; Liu, M.; Meng, M.Q.H. Improving RGB-D SLAM in dynamic environments: A motion removal approach. *Robotics and Autonomous Systems* **2017**, *89*, 110–122.
26. Newcombe, R.A.; Izadi, S.; Hilliges, O.; Molyneaux, D.; Kim, D.; Davison, A.J.; Kohi, P.; Shotton, J.; Hodges, S.; Fitzgibbon, A. Kinectfusion: Real-time dense surface mapping and tracking. 2011 10th IEEE international symposium on mixed and augmented reality. Ieee, 2011, pp. 127–136.
27. Horn, B.K.; Schunck, B.G. Determining optical flow. *Artificial intelligence* **1981**, *17*, 185–203.
28. Derome, M.; Plyer, A.; Sanfourche, M.; Besnerais, G.L. Moving object detection in real-time using stereo from a mobile platform. *Unmanned Systems* **2015**, *3*, 253–266.
29. Derome, M.; Plyer, A.; Sanfourche, M.; Le Besnerais, G. Real-time mobile object detection using stereo. 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV). IEEE, 2014, pp. 1021–1026.
30. Fang, Y.; Dai, B. An improved moving target detecting and tracking based on optical flow technique and kalman filter. 2009 4th International Conference on Computer Science & Education. IEEE, 2009, pp. 1197–1202.
31. Wang, Y.; Huang, S. Towards dense moving object segmentation based robust dense RGB-D SLAM in dynamic scenarios. 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV). IEEE, 2014, pp. 1841–1846.
32. Long, F.; Ding, L.; Li, J. DGFlow-SLAM: A Novel Dynamic Environment RGB-D SLAM without Prior Semantic Knowledge Based on Grid Segmentation of Scene Flow. *Biomimetics* **2022**, *7*, 163.
33. Alcantarilla, P.F.; Yebes, J.J.; Almazán, J.; Bergasa, L.M. On combining visual SLAM and dense scene flow to increase the robustness of localization and mapping in dynamic environments. 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 1290–1297.
34. Zhang, T.; Zhang, H.; Li, Y.; Nakamura, Y.; Zhang, L. Flowfusion: Dynamic dense rgb-d slam based on optical flow. 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 7322–7328.
35. Sun, D.; Yang, X.; Liu, M.Y.; Kautz, J. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 8934–8943.
36. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence* **2017**, *39*, 2481–2495.

37. Yu, C.; Liu, Z.; Liu, X.J.; Xie, F.; Yang, Y.; Wei, Q.; Fei, Q. DS-SLAM: A semantic visual SLAM towards dynamic environments. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 1168–1174.
38. Bescos, B.; Fácil, J.M.; Civera, J.; Neira, J. DynaSLAM: Tracking, mapping, and inpainting in dynamic scenes. *IEEE Robotics and Automation Letters* **2018**, *3*, 4076–4083.
39. Xiao, L.; Wang, J.; Qiu, X.; Rong, Z.; Zou, X. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robotics and Autonomous Systems* **2019**, *117*, 1–16.
40. Liu, Y.; Miura, J. RDS-SLAM: Real-time dynamic SLAM using semantic segmentation methods. *Ieee Access* **2021**, *9*, 23772–23785.
41. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics* **2021**, *37*, 1874–1890.
42. Liu, Y.; Miura, J. RDMO-SLAM: Real-time visual SLAM for dynamic environments using semantic label prediction with optical flow. *IEEE Access* **2021**, *9*, 106981–106997.
43. Xu, Y.; Wang, Y.; Huang, J.; Qin, H. ESD-SLAM: An efficient semantic visual SLAM towards dynamic environments. *Journal of Intelligent & Fuzzy Systems* **2022**, pp. 1–10.
44. Xie, W.; Liu, P.X.; Zheng, M. Moving object segmentation and detection for robust RGBD-SLAM in dynamic environments. *IEEE Transactions on Instrumentation and Measurement* **2020**, *70*, 1–8.
45. Lucas, B.D.; Kanade, T. An iterative image registration technique with an application to stereo vision. IJCAI'81: 7th international joint conference on Artificial intelligence, 1981, Vol. 2, pp. 674–679.