electronics

MDPI

Article

MCBM-SLAM: An Improved Mask-Region-Convolutional Neural Network-Based Simultaneous Localization and Mapping System for Dynamic Environments

Xiankun Wang and Xinguang Zhang *

School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China; perseverance2022@163.com
* Correspondence: 06140002@sues.edu.cn

Abstract: Current research on SLAM can be divided into two parts according to the research scenario: SLAM research in dynamic scenarios and SLAM research in static scenarios. Research is now relatively well established for static environments. However, in dynamic environments, the impact of moving objects leads to inaccurate positioning accuracy and poor robustness of SLAM systems. To address the shortcomings of SLAM systems in dynamic environments, this paper develops a series of solutions to address these problems. First, an attention-based Mask R-CNN network is used to ensure the reliability of dynamic object extraction in dynamic environments. Dynamic feature points are then rejected based on the mask identified by the Mask R-CNN network, and a preliminary estimate of the camera pose is made. Secondly, in order to enhance the picture matching quality and efficiently reject the mismatched points, this paper proposes an image mismatching algorithm incorporating adaptive edge distance with grid motion statistics. Finally, static feature points on dynamic objects are re-added using motion constraints and chi-square tests, and the camera's pose is re-estimated. The SLAM algorithm of this paper was run on the KITTI and TUM-RGBD datasets, respectively, and the results show that the SLAM algorithm of this paper outperforms the ORB-SLAM2 algorithm for sequences containing more dynamic objects in the KITTI dataset. On the TUM-RGBD dataset, the Dyna-SLAM algorithm increased localization accuracy by an average of 71.94% when compared to the ORB-SLAM2 method, while the SLAM algorithm in this study increased localization accuracy by an average of 78.18% when compared to the ORB-SLAM2 algorithm. When compared to the Dyna-SLAM technique, the SLAM algorithm in this work increased average positioning accuracy by 6.24%, proving that it is superior to Dyna-SLAM.

Keywords: SLAM; Mask R-CNN; motion constraints; chi-square test; attention

Citation: Wang, X.; Zhang, X. MCBM-SLAM: An Improved Mask-Region-Convolutional Neural Network-Based Simultaneous Localization and Mapping System for Dynamic Environments. *Electronics* **2023**, *12*, 3596. https://doi.org/10.3390/electronics12173596

Academic Editors: Felipe Jiménez and Shiho Kim

Received: 16 June 2023
Revised: 21 July 2023
Accepted: 21 August 2023
Published: 25 August 2023

1. Introduction

Simultaneous localization and mapping (SLAM) is a fundamental requirement for the autonomous navigation of unmanned vehicles. Visual SLAM is a preferred method over laser SLAM because the cameras are inexpensive, easy to deploy, easy to use and can provide unmanned vehicles with a wealth of information about the environment in which they are operating. Visual SLAM has therefore attracted the interest of many researchers in recent years, resulting in a number of new SLAM solutions.

With the continuous research on SLAM, numerous systems have been suggested, such as ORB-SLAM2 [1], ORB-SLAM3 [2] and RESLAM [3]. However, these SLAM algorithms all assume that the surrounding environment is static, which can achieve high accuracy with static datasets but can produce large errors in dynamic environments [4]. When estimating camera poses, pixel matching is utilized. However, in a dynamic environment, moving items can cause mistakes in the positioning estimate towards the end. This is due to the fact that both the camera and the objects are in motion. Therefore, many scholars started to study how to remove the impact of moving items in dynamic surroundings.

Some scholars started to remove the impact of moving objects by building mathematical models, but the mathematical models built have many constraints and also increase the computational effort, reducing the real-time performance of SLAM algorithms. The field of image recognition has seen significant advancements with the development of deep learning, and numerous scholars have proposed incorporating deep learning networks into SLAM systems to mitigate the influence of moving objects. Deep learning-based visual SLAM algorithms such as Dyna-SLAM [5], DS-SLAM [6] and others have been created. Although the proposed deep learning networks have performed well in recognizing dynamic objects, there is further room for improvement in the completeness of dynamic object recognition. Therefore, this paper proposes a Mask R-CNN network based on an attention mechanism to make the edge segmentation of dynamic objects more complete to enhance the accuracy of the SLAM system's localization. The main contributions of this paper are as follows:

(1) This paper adds an attention mechanism module to the existing neural network to solve the problem of incomplete dynamic object segmentation;
(2) An image mismatch rejection algorithm incorporating grid motion statistics with adaptive margins is proposed;
(3) Re-addition of static feature points on potential dynamic features using chi-square test and motion constraints.

2. Related Work

Currently, there is ongoing research on visual SLAM for dynamic environments, and many excellent algorithms have been proposed to handle dynamic objects in dynamic environments during this period. These dynamic object detection methods are broadly classified into four categories: dynamic object detection using geometric mathematical models; dynamic object detection using semantic segmentation; dynamic object detection using a combination of semantic segmentation and geometric mathematical models; and dynamic object detection using multi-sensor fusion with each other.

Xing et al. [7] suggest using a dynamic detection tracking module that combines semantic and metric information to remove dynamic features from dynamic objects. Zhong et al. [8] proposed a robust SLAM system, WF-SLAM, which uses geometric information and semantic segmentation tightly coupled to obtain dynamic information in dynamic scenes and defines weights for feature points to transform the bit-pose optimization into weight optimization. Lin et al. [9] used a combination of panoramic segmentation and optical flow point features to detect potential moving targets. To ensure reliable tracking, they developed a strategy to supplement key points. Yin et al. [10] loosely fused the stereo scene flow with the inertial measurement unit to achieve dynamic feature detection and tightly coupled dynamic and static features with the Inertial Measurement Unit (IMU) for nonlinear optimization. Li et al. [11] proposed the visual SLAM algorithm DP-SLAM. It uses coefficient features and combines geometric constraint and semantic segmentation to track dynamic key points. The algorithm operates within a Bayesian probability estimation framework. Cheng et al. [12] proposed a SLAM system. It includes two parallel threads: the object detection thread, which obtains two-dimensional semantic information, and the semantic mapping thread, which fuses semantic and geometric information to reject dynamic features in the tracking thread. Ni et al. [13] proposed a SLAM algorithm for monocular cameras in dynamic environments that controls the number of feature points by labeling them with a reliability concept and then uses an improved frame difference detection method based on a partial detection strategy to detect dynamic targets in the environment. Wang et al. [14] used a multi-motion segmentation method to segment the motion models of different motion targets to obtain accurate masks of the motion targets. Chen et al. [15] provided a new neural network, the contour-optimized hybrid expanded convolutional neural network (CO-HDC) algorithm, which performs lightweight computation based on contour segmentation accuracy and enhances the contour using a contour quality evaluation algorithm as a way to distinguish dynamic feature points from static

feature points. Wu et al. [16] proposed a new YOLO-SLAM by adding the Darknet19 network to the traditional YOLOv3 network to speed up the SLAM system to generate semantic information. Zhang et al. [17] used a lightweight YOLOv5 network to improve the system's running speed. At the same time, a pyramid shaped scene parsing network segmentation head was added at the head of the YOLOv5 network to achieve semantic extraction in the environment. Gou et al. [18] proposed a 3D semantic system, PW_SLAM, to obtain semantic information by integrating a semantic segmentation network (PWnet) with a SLAM system and then filter dynamic key points using a dynamic uncertainty keypoint classifier (DUKC) to improve localization accuracy. Liu et al. [19] proposed a dynamic feature point detection algorithm based on double K-mean clustering with static weights for static feature points. Zhang et al. [20] proposed a method to detect closed loops using a combination of image patching and feature selection. This helps to counter the impact of insufficient feature points extracted, which can occur when dynamic objects are rejected. Dai et al. [21] divided the feature points and dynamic feature points on static objects into different groups and removed irrelevant points and dynamic feature points during point correlation optimization. Rosinol et al. [22] proposed a new approach: 3D dynamic scene graph (DSG). This method aims to narrow the gap between human and robot perception by capturing both the metric and semantic information of the dynamic environment seamlessly. Cheng et al. [23] designed a dynamic region detection method. It uses Bayesian analysis and takes into account both prior knowledge and information gathered during the object detection process. Han et al. [24] proposed a PSPNet-SLAM system. It combines the SLAM system with the PSPNet semantic segmentation network to identify and eliminate dynamic feature points using optical flow and semantic segmentation. Zang et al. [25] used a deep learning network based on the attention mechanism of YOLOv5s to obtain the a priori dynamic objects in the scene, selected the feature points by the percentage of the a priori dynamic information in each frame and finally determined the dynamic regions using the Lucas–Kanade optical flow and RANSAC algorithms. Yuan et al. [26] combined the geometric constraint method for line segment features and the polar constraint method for feature points as a way to realize the separation of dynamic and static objects and eliminate the dynamic noise of points and line segments using the dynamic feature tracking method based on Bayesian theory. Zhang et al. [27] proposed an improved Mask R-CNN network to address the issue of incomplete edge detection. The network includes an added Mask R-CNN network at the edge detection end, and motion consistency detection is used for dynamic feature points to improve accuracy. Gong et al. [28] screened keyframes using an optical flow method, extracted feature points in keyframes using adaptive thresholding and eliminated dynamic points using YOLOV5.

All the above algorithms are authoritative studies in the field of SLAM, but these algorithms still have some drawbacks. Among them, when using mathematical models for dynamic feature point rejection, the model is too computationally complex. When using semantic segmentation for dynamic feature point rejection, although the computation is less complicated compared with the method of building mathematical models, there is also incomplete segmentation of dynamic object edges, which leads to less accurate localization accuracy in SLAM systems. Therefore, in this paper, we will add an attention mechanism to the semantic segmentation of the Mask R-CNN network and enhance the integrity of edge segmentation using this method.

3. System Framework

3.1. SLAM System Framework

ORB-SLAM2 is an outstanding work based on the feature point method, and the system is highly applicable to monocular cameras, binocular cameras and RGB-D depth cameras. ORB-SLAM2 is composed of three threads, which are the tracking thread, local map building thread and loopback detection thread, through which the robustness of tracking keyframes, trajectory construction and map building performance is improved. Furthermore, the SLAM system in this paper is improved on the basis of these three threads.

The SLAM system structure of this article is shown in Figure 1. First, the SLAM system in this paper proposes an image mismatching algorithm that incorporates grid motion statistics with adaptive margins. Second, a semantic segmentation module is added to the tracking thread, in which the semantic segmentation module adopts the Mask R-CNN network based on the attention mechanism, through which the dynamic objects in the scene are segmented out and the preliminary estimation of the bit position is performed at the same time. Finally, in order to enhance the robustness of the system, motion constraints and chi-square tests are used to re-add the static feature points on the dynamic objects, and the estimation of the camera position is corrected.

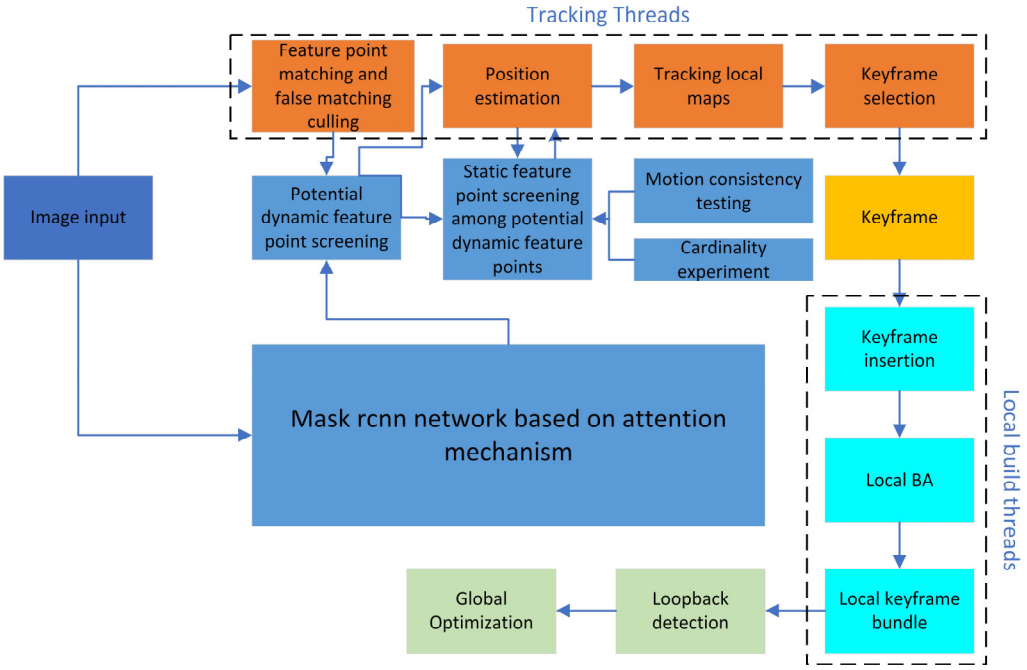


Figure 1. Framework of SLAM system.

3.2. Mask R-CNN Network Based on Improved Attention Mechanism

For the SLAM system in this paper, there may be a large number of mismatching points after the violent matching of feature points, so an improved GMS algorithm is proposed in this paper to eliminate the mismatching points. The main idea of the GMS algorithm is to count the number of supported points in a pair of regions (i.e., the score), by which the correct matches can be distinguished from the incorrect ones.

The formula for the score, S_{ij}, of each feature point, X_i, in the GMS algorithm:

$$S_{ij} = \sum_{k=1}^{9} N_{i^k j^k}, k = 1, 2, 3, \ldots, 9 \tag{1}$$

The N_{ij} denotes the number of support points located in the adjacent grid, and k are the nine neighboring grids around the feature point X_i (which also includes the grid where the feature point is located); the score of the feature point X_i, is the number of all feature matching pairs in the nine grids except the feature point X_i. To distinguish between correctly matched pairs and incorrectly matched pairs, the binomial distribution is used in this paper to approximate the distribution of scores, and the binomial distribution is shown below:

$$S_i \sim \begin{cases} B(n, p_t) & X_i \text{ is the correct match} \\ B\left(n, p_f\right) & X_i \text{ is an incorrect match} \end{cases} \tag{2}$$

The variable n denotes the average number of feature points found in each grid, the variable p_t represents the likelihood of correctly matching feature points to the correspond-

ing region grid, and the variable p_f represents the likelihood of mistakenly matching feature points to their respective regions.

The formula for the GMS algorithm to determine the correct and incorrect matches is shown in Equation (3).

$$cell - pair\{i,j\} \in \begin{cases} True & S_{ij} > \tau \\ False & other \end{cases} \tag{3}$$

where S_{ij} represents the score of each feature point, X_i, τ represents the threshold value, and a score greater than the threshold value is judged to be a correct match. A match that falls in two grids at the same time is referred to as a similar neighborhood, i.e., $cell - pair$. The $\{i,j\}$ is a pair of two matching grid regions.

Since the GMS algorithm handles the edge feature points of the grid by simply shifting the feature points by half the network in the x, y and $x-y$ tilt directions, respectively, this edge handling method results in judging many correct matches as incorrect matches. This paper proposes the following improvement to address such a shortcoming: let the support points on the edges of the grid be assigned to the surrounding grid, which will lead to an increase in the score of correct matches thus making the distance between the scores of correct and incorrect matches pull apart and increasing the distinguishability of correct and incorrect matches.

As shown in the figure below, point A is at the edge of grid 6, and the grid containing the edge is the statistical area of the support points when actually calculating the score support, which means that point A at this time not only belongs to grid 6 but also to grid 5 and grid 3. At this point, when calculating feature point $X1$ and feature point $X2$, if the match is correct, the number of this support points will increase by 1 (i.e., the score is increased by 1), and if the match is incorrect, the score will not increase (the value is small). By using this method, the score of correct matches can be greatly improved.

In this study, an adaptive algorithm is utilized to determine the distance between the edges of the grid, which is also the length of the arrow in the Figure 2. For the edge distance of the grid, if the edge distance is too large, the number of false support points will increase, and if the edge distance is too small, the number of support points will be insufficient. Therefore, in this paper, the grid edge distance is calculated by the following equation:

$$d = \alpha \frac{(\omega, h)}{\sqrt{G}} \tag{4}$$

where d is the optimal grid edge distance, ω and h are the length and width of the image, G is the number of grids and $\alpha \in [0,1]$ is the weighting factor of the edge distance. The (ω, h) represents the length of the hypotenuse of a right triangle formed by using the length and width of the image as the two right sides of the triangle.

In this paper, we quantify the distance between the correct match distribution and the incorrect match distribution by defining a distance metric D:

$$D = (m_s - m_D) - (s_S - s_D) \tag{5}$$

where m_S, m_D, s_S and s_D represent the mean and standard deviation of two binomial distributions. The formula for the mean and standard deviation of binomial distributions are:

$$D = \left(n \cdot p_t - n \cdot p_f\right) - \left(\sqrt{n \cdot p_t(1-p_t)} + \sqrt{n \cdot p_f\left(1-p_f\right)}\right) \tag{6}$$

Since p_f tends to 0, we have:

$$D = n \cdot p_t - \sqrt{n \cdot p_t(1-p_t)} \tag{7}$$

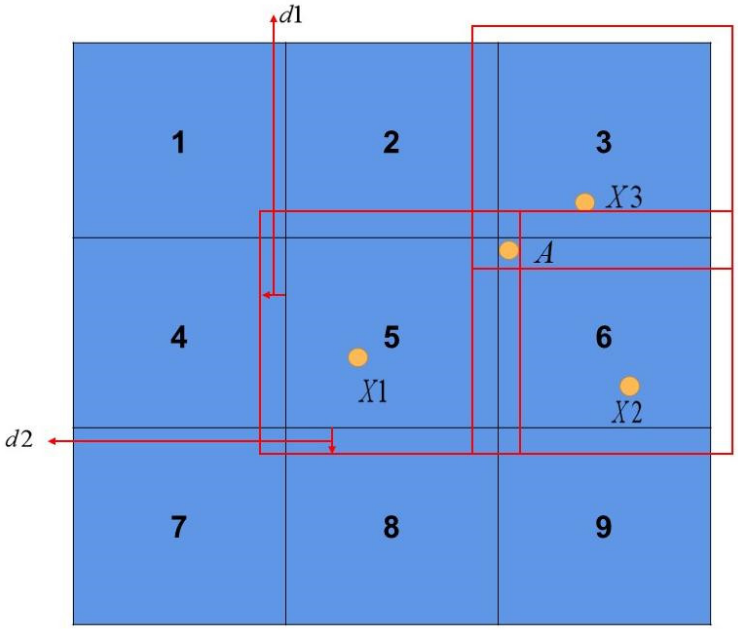


Figure 2. Processing of edge feature points.

From the distance metric, D, it can be seen that the larger the n (number of feature matching pairs in the grid), the larger the value of D. A larger D also indicates a greater distance between two binomial distributions of possible events, i.e., a greater distinction between correct and incorrect matches. This method is equivalent to increasing the score, S, while reducing the number of cycles in the main body of the algorithm, i.e., increasing the value of n. The value of D increases with the increase in n, which also allows better differentiation between correct and incorrect matches. The Figures 3 and 4 show the comparison of the operation results of the original GMS algorithm and the improved GMS in this paper. This paper's algorithm has a higher number of accurate matching pairs compared to the GMS algorithm.

3.3. Mask R-CNN Network Based on Attention Mechanism

The Mask R-CNN [29] network is the best of the 2017 ICCV and improves on the Faster R-CNN network with the addition of a new branch of prediction masks. In addition, the Fast R-CNN [30] network also proposes ROIAlign, which solves the problem of localization error in ROIPooling during operation. The Mask R-CNN network is capable of performing both target detection and instance segmentation functions. The specific structure of the Mask r-CNN network is shown in Figure 5. For this paper, the main operation process of the Mask R-CNN network is as follows: firstly, for the environment, dynamic objects are detected, and the candidate frames of dynamic objects are selected; then we classify the dynamic objects and choose their class in the target frame; finally, the dynamic objects in the environment are segmented at the pixel level and the image mask is generated.

In order to solve the problem of erroneous segmentation at dynamic object edges, the Convolutional Block Attention Module (CBAM) spatial attention module was included. This paper presents the CBAM-Mask network structure, which can be viewed in Figure 6. In Figure 6, it can be seen that this paper adds the spatial attention mechanism module-CBAM at each stage after ResNet-50. This module is mainly used to distinguish the importance of different feature regions, enhancing the important ones and suppressing the unimportant ones. Furthermore, for ResNet-50, the last layer of the Conv2_x stage output is linked to the FPN for feature fusion at the higher-level features to fuse more underlying feature information.

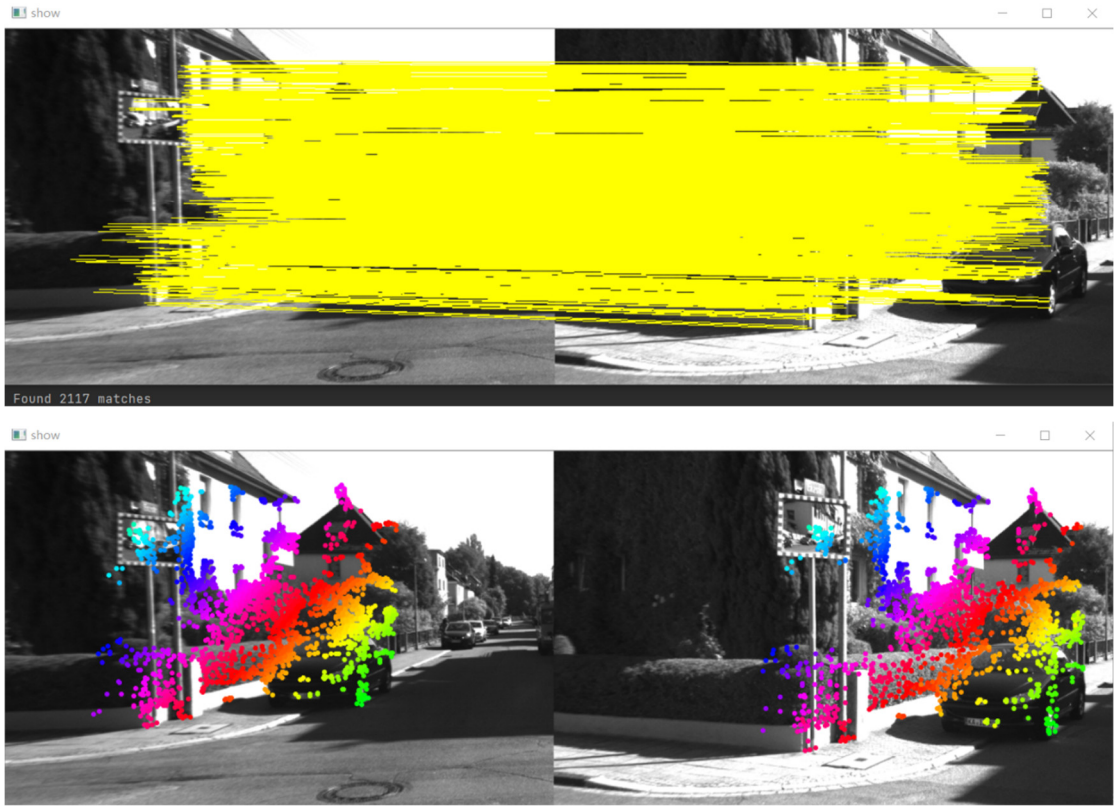


Figure 3. Original GMS algorithm matching results.

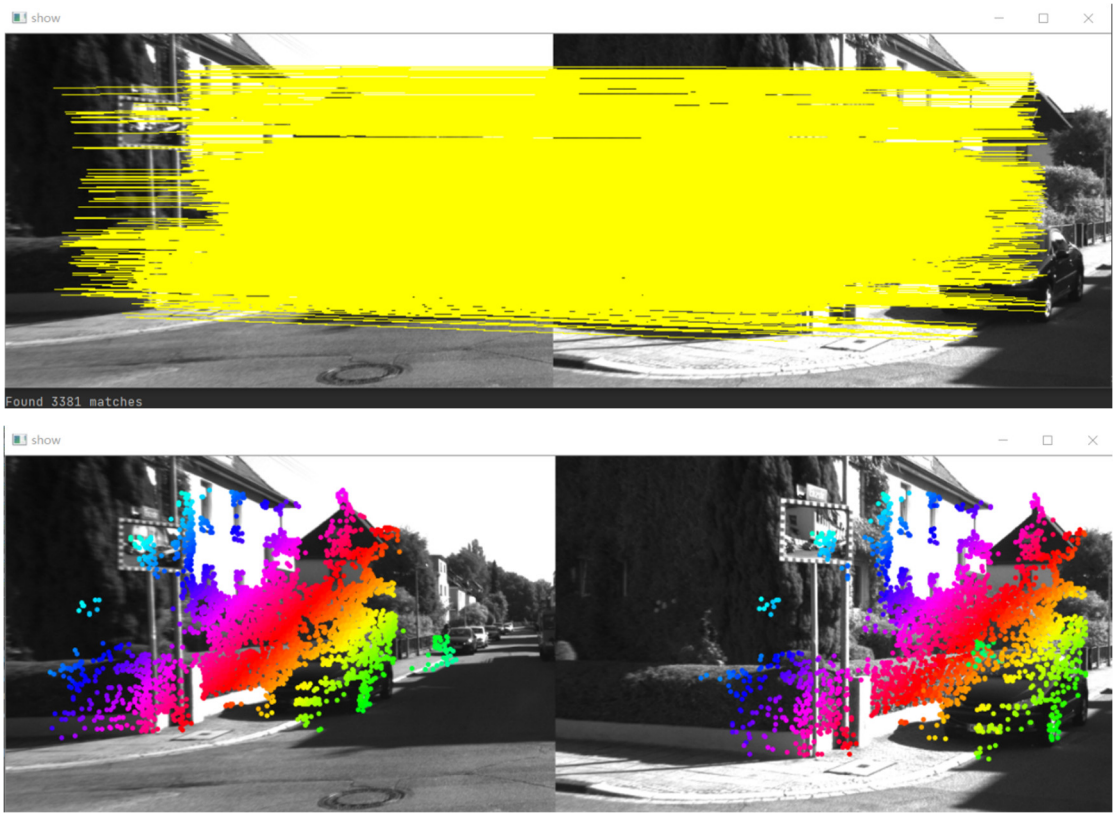


Figure 4. Matching results of the improved GMS algorithm.

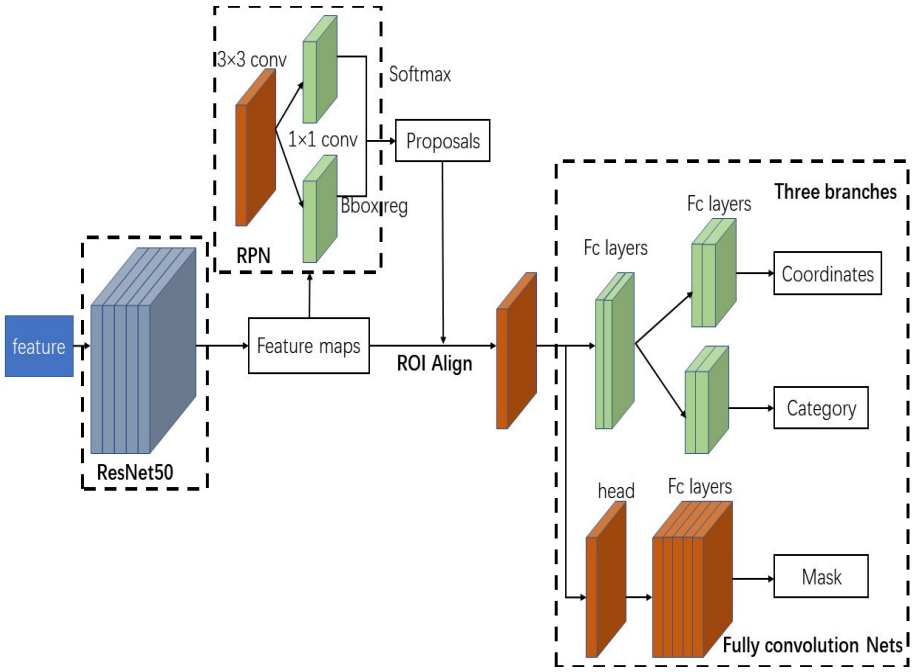


Figure 5. Basic structure of Mask R-CNN network.

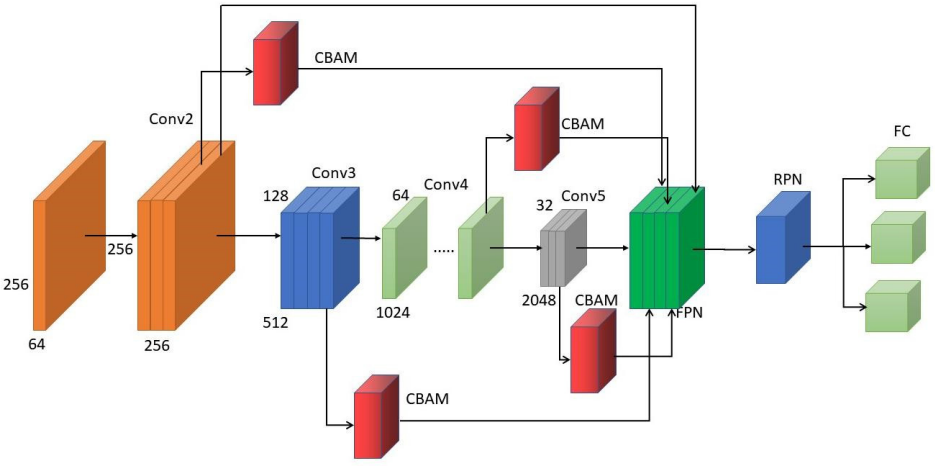


Figure 6. CBAM-Mask network structure.

The spatial attention module of CBAM focuses on modeling the correlation between channels and space by performing a convolution operation on the feature map and then using the global information of the convolved feature map dynamically. One of the specific operations is shown below, where the input image is first subjected to maximum pooling and average pooling. Finally, the convolution operation is performed and Sigmoid activation is processed to generate a matrix of size $1 \times \mathrm{H} \times L$, the H and the L are represented as the height and width of the feature map. Figure 7 illustrates the specific structure of the spatial attention module of CBAM.

$$\mathrm{M}(F) = [AvgPool(F); MaxPool(F)] \tag{8}$$

$$\mathrm{M_s}(F) = \sigma(f^{7\times7}(M(F))) \tag{9}$$

where F is the feature map; $AvgPool$ is the average pooling; $MaxPool$ is the maximum pooling; f is the convolution operation; the activation function is σ; and the spatial attention parameter matrix is $\mathrm{M_s}(F)$.

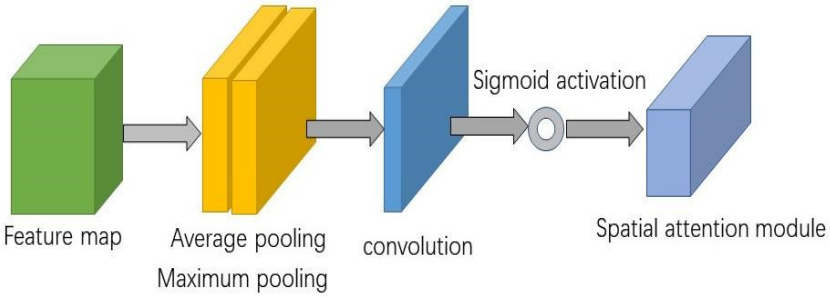


Figure 7. Spatial attention module.

A comparison of the test results with and without the addition of the attention mechanism is shown in Figure 8. It can be seen from the figure that when the attention mechanism is not added, many segmentation regions are incomplete, i.e., the segmentation regions are not accurate, while the accuracy of the segmentation is improved compared to the former for the network with the attention mechanism module added.

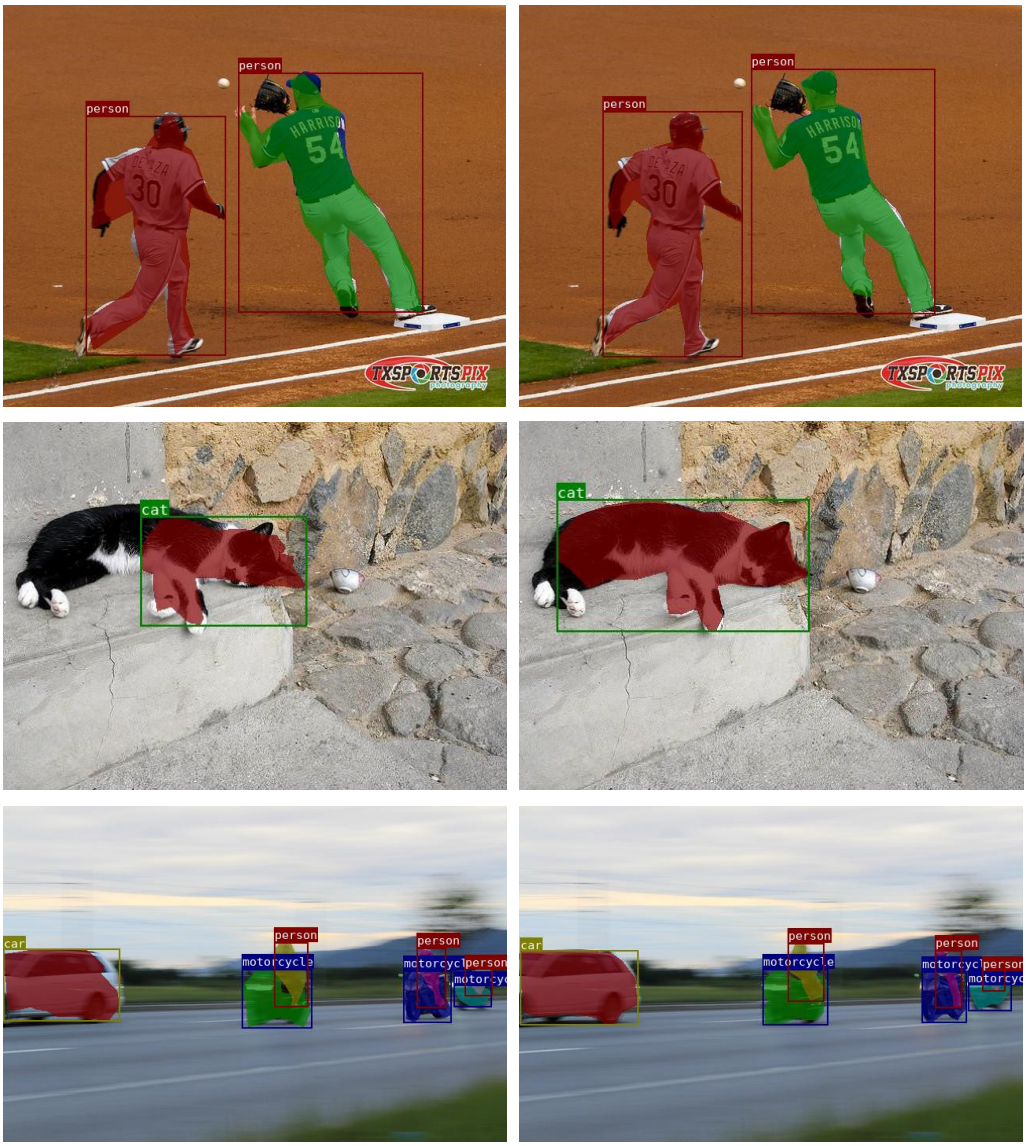


Figure 8. Comparison of the results of adding attention and not adding attention (the **left column** has no added attention mechanism; the **right column** has added attention mechanism).

3.4. Cardinality Experiment and Motion Consistency Detection

As shown in Figure 9, C_1 and C_2 are the optical center positions of the camera at two adjacent moments, T_1 and T_2. Point p_1 is a feature point on the potential dynamic object, when p_1 is a static feature point, the pixel coordinates of this static feature point on the two imaging planes are x_1 and x_1, respectively, and these two coordinates fall on the epipolar lines, l_1 and l_2, of the two imaging planes, respectively. If p_1 is a dynamic feature point, at the T_2 moment, p_1 has moved to the p_2 position, the pixel coordinate in the C_2 imaging plane is x_3. The expression of pole line l_2 is shown in Equation (10), where pole line l_2 satisfies the expression of Equation (11).

$$ax + by + c = 0 \tag{10}$$

$$[a, b, c]^T = Fx_1 \tag{11}$$

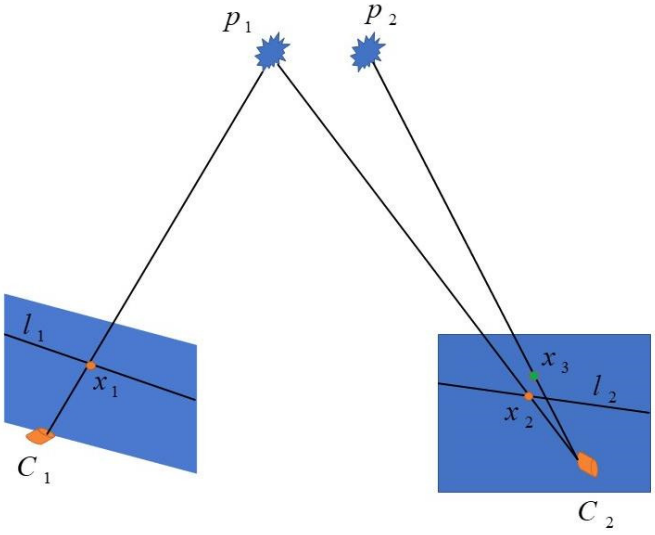


Figure 9. Motion consistency detection with cardinality experiment.

The F in Equation (11) is the transformation basis matrix between two image frames. The distance from x_2 to the poles is expressed as shown in Equation (12):

$$d = \sqrt{\frac{[a, b, c] \cdot x_3}{a^2 + b^2}} \tag{12}$$

where the square of distance d in Equation (12) obeys a cardinal distribution with a confidence level of 95%, a rejection domain of 3.84 and a degree of freedom of 1. In ORB-SLAM2, an image pyramid is used to achieve scale invariance, and the scaling factor of the image pyramid is 1.2. The variance of the pixel coordinates of the feature points of the image pyramid in layer n can be obtained as 1.2^{2n}. The square of the distance from dynamic feature point x_2 to pole l_2 should satisfy Equation (13), and the dynamic and static characteristics of all potential dynamic features can be further determined using Equation (13).

$$d^2 > 3.84 \times 1.2^n \tag{13}$$

Let P_1 and P_2 be the coordinates of point p_1 under two camera coordinate systems that satisfy Equation (14).

$$P_2 = T_{c2} T_{c1}^{-1} P_1 \tag{14}$$

In Equation (14), T_{c1} and T_{c2} represent the transformation matrix between the world coordinate system and the two camera coordinate systems.

Q_2 is the coordinate of point p_2 under the camera coordinate system C_2. Where P_2 and Q_2 satisfy a cardinal distribution with a confidence level of 95%, a rejection domain of

7.81 and a degree of freedom of 3. Then, at this point, the dynamic characteristic point P_2 satisfies Equation (15):

$$(P_2 - Q_2)(P_2 - Q_2)^T > 7.81 \times 1.2^{2n} \tag{15}$$

4. Experimental Analysis

The computer hardware used in this paper is an Asus laptop (Intel i5-10300H CPU, GTX1660Ti graphics card, 16 GB RAM), and the SLAM algorithm of this paper is experimentally analyzed on TUM-RGBD and KITTI datasets, by which the robustness of this paper's algorithm under a dynamic environment is evaluated. First, this study compares our SLAM algorithm with ORB-SLAM2 and other excellent SLAM algorithms in dynamic environments on the TUM-RGBD indoor dataset to reflect the robustness of this paper's SLAM algorithm in indoor environments. This paper evaluates the effectiveness of the SLAM algorithm proposed in this study in dynamic outdoor environments by comparing it with ORB-SLAM2 and other top-performing SLAM algorithms using the KITTI outdoor dataset. The aim of this comparison is to demonstrate the robustness of the SLAM algorithm proposed in this paper.

4.1. Experiments on SLAM Algorithm in Dynamic Environment

4.1.1. Experimental Analysis on TUM RGBD Dataset

The TUM-RGBD dataset, published by TUM's Computer Vision Lab, contains texture-rich office scenes, some of which contain many objects that are constantly moving. The SLAM algorithm in this study involves a great deal of dynamic de-objects, so the sequences containing dynamic elements are chosen for the experiments in this paper: s_static, w_halfsphere, w_static and w_xyz. In the TUM-RGBD dataset, the main dynamic objects are the office personnel in the scene. The w_static in the sequence is a low dynamic sequence, and the other three are high dynamic sequences. The letter "s" stands for sitting and "w" stands for standing, and the word after the "_" denotes the camera movement mode in the sequence. It can be seen from Figure 10 that the SLAM in this paper can eliminate the effect of dynamic objects well under the office dynamic sequence.

Figure 10. Results of feature point extraction after removing dynamic objects and without removing dynamic objects.

In the experimental comparison, the root mean square error (RMSE) of the absolute trajectory error (ATE) is used as the quantitative evaluation criterion of the experiments in this paper, and the results of the trajectory comparison running under high dynamic sequences are used as the qualitative evaluation criterion of this paper. The equations for ATE and RMSE are shown in Equations (16) and (17). The standard deviation (SD) is used as a quantitative criterion for the degree of dispersion of the trajectory estimation of the SLAM system in this paper. From Table 1, it can be seen that in the low dynamic sequence s_static, the three SLAM algorithms' trajectory estimation dispersion are close to each other, but in the other three high dynamic sequences, the SLAM algorithms of this paper's trajectory estimation dispersion are better than the other two algorithms. Table 2 displays one of

the qualitative outcomes, which shows quantitatively the maximum, minimum, mean and median values of RMSE. Table 3 shows that the RMSEs of the three SLAM algorithms are similar in the low-dynamic sequence s_static. In the three high dynamic sequences, the SLAM and Dyna-SLAM algorithms in this paper have smaller errors compared to the ORB-SLAM2 algorithm, and the accuracy of the SLAM algorithm in this paper is improved compared to the Dyna-SLAM algorithm. By analyzing the information presented in Table 3, one can observe that the median, mean and minimum RMSE values of the SLAM algorithm discussed in this paper are reduced by 78.79%, 75.15% and 80.59% on average. The median, mean and minimum values of the RMSE of the Dyna-SLAM algorithm were reduced by 73.22%, 68.07% and 74.53% on average. The accuracy of the SLAM algorithm in this paper is better than Dyna-SLAM in terms of the average reduction rate, and the SLAM algorithm in this paper has an average accuracy improvement of 78.18% relative to ORB-SLAM2 and an average accuracy improvement of 71.94% relative to Dyna-SLAM.

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

$$RMSE(ATE_{1:n}, \Delta) = \left(\frac{1}{m}\sum_{i=1}^{m} \|trans(ATE_i)\|^2\right)^{\frac{1}{2}} \tag{17}$$

where ATE_i represents the ATE of frame i, P_i represents the bit pose estimated by the algorithm, Q_i represents the real bit pose, Δ represents the time interval and S represents the similar transformation matrix from the estimated bit pose to the real bit pose.

Table 1. Comparison of the SD results of ORB-SLAM, Dyna-SLAM and MCBM-SLAM in the TUM-RGBD dataset.

Sequence	SD(m)		
	ORB-SLAM2	**Dyna-SLAM**	**MCBM-SLAM**
s_static	0.0042	0.0039	0.0040
w_halfsphere	0.3085	0.0192	0.0187
w_static	0.1925	0.0048	0.0042
w_xyz	0.3267	0.0085	0.0065

Table 2. Comparison of the RMSE results of ORB-SLAM, Dyna-SLAM and MCBM-SLAM in the TUM-RGBD dataset.

Sequence	ORB-SLAM2				Dyna-SLAM				MCBM-SLAM			
	Median	**Mean**	**Min**	**Max**	**Median**	**Mean**	**Min**	**Max**	**Median**	**Mean**	**Min**	**Max**
s_static	0.012	0.011	0.010	0.012	0.011	0.010	0.009	0.012	0.009	0.010	0.007	0.012
w_halfsphere	0.916	0.976	0.828	1.210	0.058	0.115	0.041	0.299	0.038	0.036	0.027	0.040
w_static	0.437	0.429	0.394	0.445	0.015	0.015	0.014	0.016	0.008	0.007	0.006	0.009
w_xyz	0.771	0.726	0.590	0.800	0.044	0.094	0.020	0.215	0.022	0.023	0.017	0.025

Table 3. RMSE reduction ratio of Dyna-SLAM and MCBM-SLAM relative to ORB-SLAM2.

Sequence	Dyna-SLAM				MCBM-SLAM			
	Median	**Mean**	**Min**	**Max**	**Median**	**Mean**	**Min**	**Max**
s_static	8.33%	9.09%	10%	-	25.00%	9.09%%	30%	-
w_halfsphere	93.67%	79.65%	95.05%	75.29%	95.85%	96.31%	96.74%	96.69%
w_static	96.57%	96.50%	96.45%	96.40%	98.17%	98.37%	98.48%	97.98%
w_xyz	94.29%	87.05%	96.61%	73.13%	96.15%	96.83%	97.12%	96.88%

The values depicted in Figures 11 and 12 qualitatively show the comparisons between the two open-source SLAM algorithms and the SLAM method used in this paper on two distinct sequences. The ORB-SLAM2 trajectory undergoes a wide deviation relative to the true trajectory of the sequence. As shown in Figures 13 and 14, an analysis of the absolute trajectory errors of the ORB-SLAM2, Dyna-SLAM and MCBM-SLAM algorithms on the w_xyz and w_halfsphere sequences is qualitatively demonstrated. The darker color in the figure indicates the larger absolute trajectory error, from which we can see that the comparison results of absolute trajectory error between Dyna-SLAM and MCBM-SLAM algorithms in w_xyz and w_halfsphere sequences are similar, but our SLAM algorithm is superior to the Dyna-SLAM algorithm.

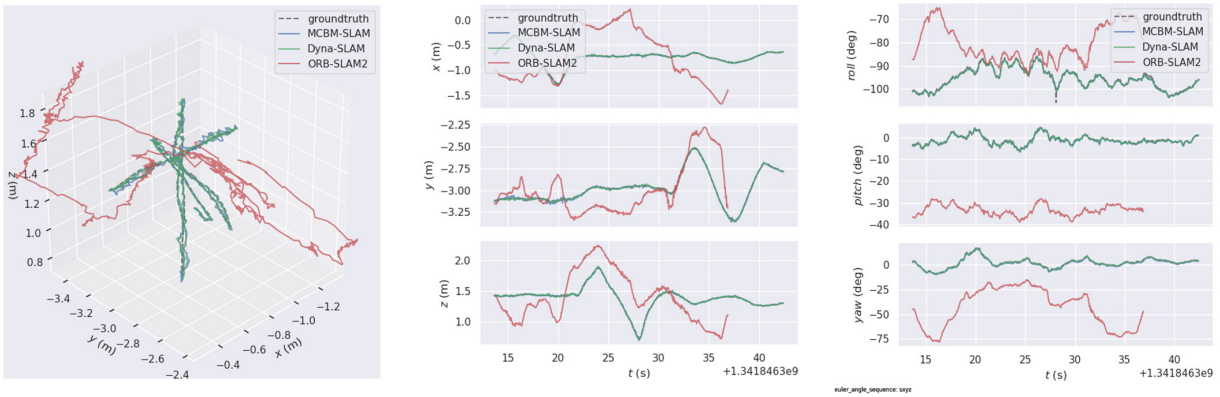


Figure 11. Comparison of the trajectories under the w_xyz sequence.

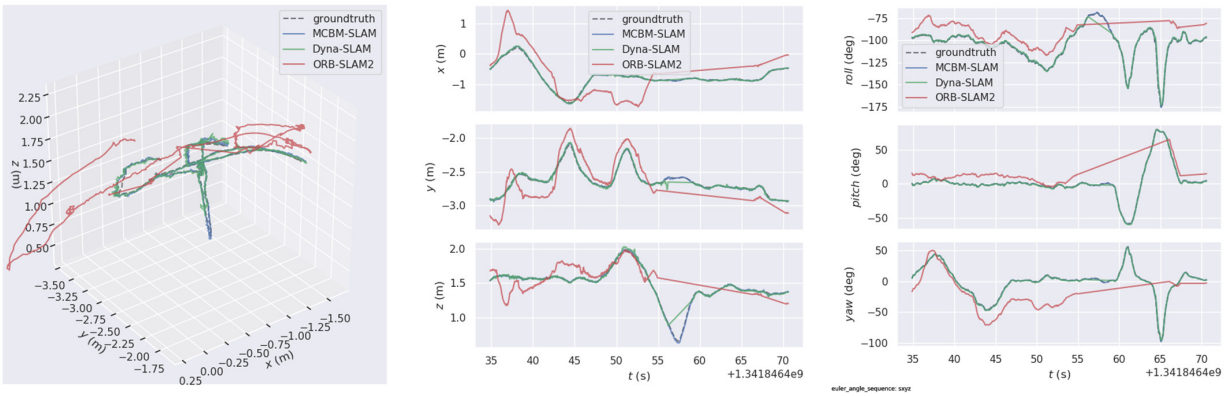


Figure 12. Comparison of the trajectories under the w_halfsphere sequence.

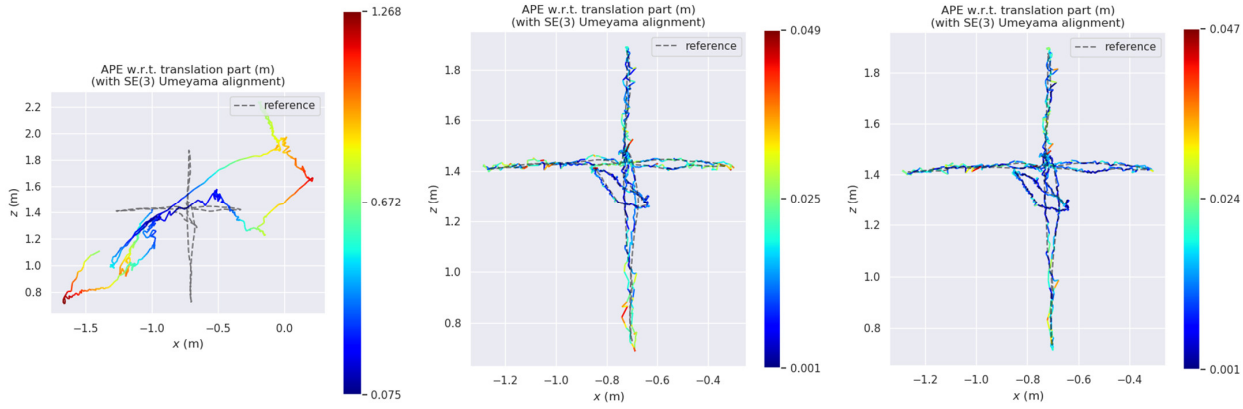


Figure 13. Comparison of the absolute trajectory errors of the three algorithms on the w_xyz sequence.

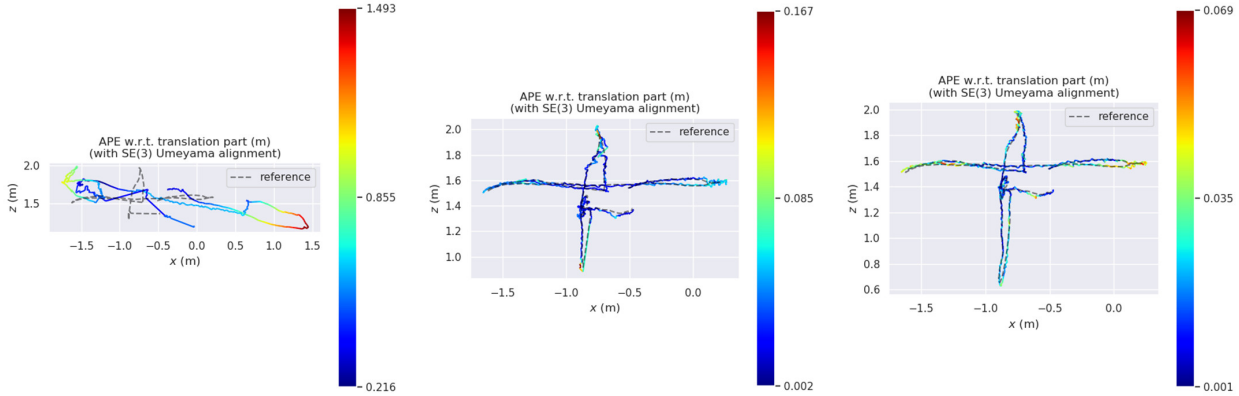
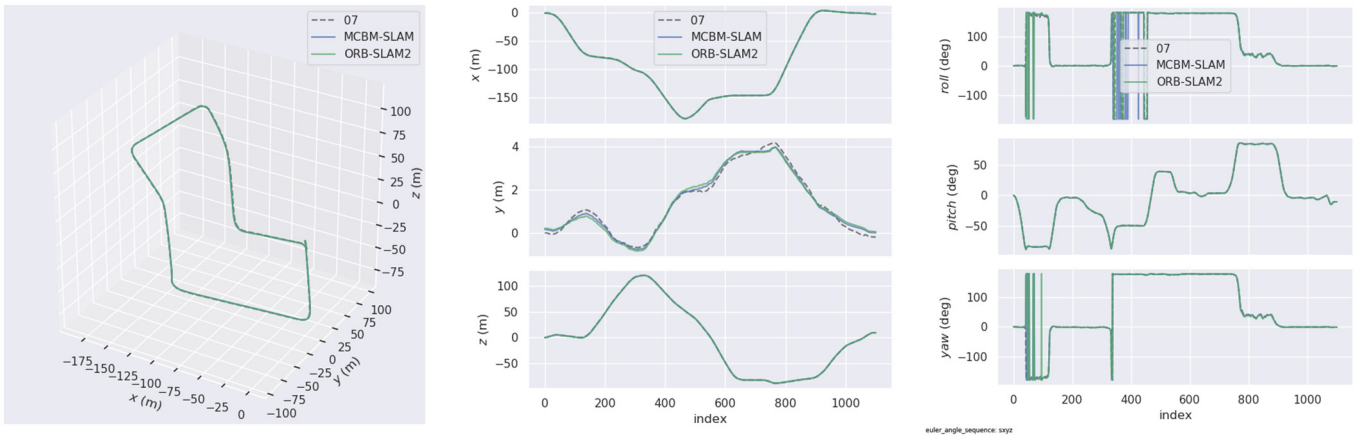


Figure 14. Comparison of the absolute trajectory errors of the three algorithms in the w_halfsphere sequence.

4.1.2. Experimental Analysis on KITTI Dataset

The KITTI dataset is a research resource for autonomous driving jointly sponsored by the Karlsruhe Institute of Technology in Germany and the Toyota Institute of Technology in Chicago. In this paper, the KITTI07 sequence is selected, where KITTI07 is mainly used for driving on roads in villages and towns, in which there are moving vehicles during the driving period and more vehicles driving in the turns, which can meet the experimental requirements of the SLAM algorithm in this paper. Figure 15 compares the SLAM algorithm's running trajectory as discussed in this paper. From Figure 15a, it can be seen that the algorithm in this paper and the ORB-SLAM2 algorithm have high similarity with the true value trajectory, so the comparison results in the xyz and rpy directions are also shown in this paper, as shown in Figure 15b,c. From Figure 15b,c, it is evident that the SLAM algorithm presented in this paper surpasses the ORB-SLAM2 algorithm, especially when the sequence is run to the stage where more dynamic objects appear in the sequence. The trajectory of ORB-SLAM deviates significantly. In Figure 16, a comparison is made between the SLAM algorithm in this paper and the ORB-SLAM2 algorithm regarding their absolute trajectory error. The results indicate that the SLAM algorithm in this paper performs better than ORB-SLAM2 in terms of absolute trajectory error.



(**a**) (**b**) (**c**)

Figure 15. Comparison of trajectories on the KITTI07 sequence. (**a**) Comparison chart with real trajectory. (**b**) Comparison results in xyz direction. (**c**) Comparison results in the rpy direction.

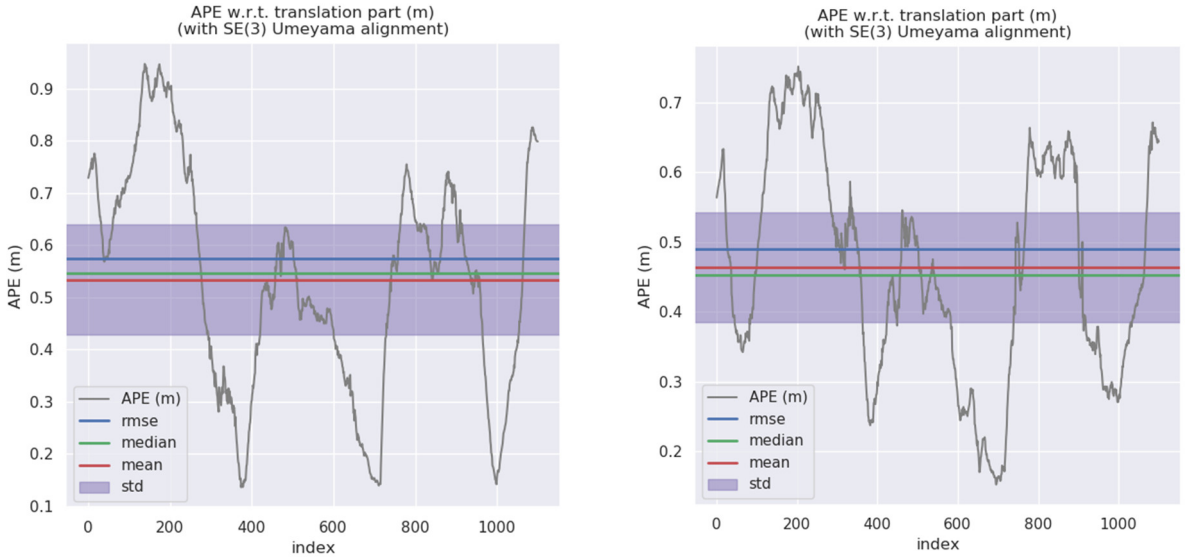


Figure 16. Comparison of absolute trajectory error between ORB-SLAM2 and MCBM-SLAM algorithm.

To further verify the reliability of the algorithms in this paper, the two algorithms are tested on the 00–10 sequences in the KITTI dataset. Among them, the root mean square error (RMSE) of the absolute trajectory error (ATE) is used as the evaluation criterion for accuracy. By referring to Table 4, it is evident that the algorithm discussed in this paper performs better as the number of vehicles in the sequence increases. However, it is worth noting that in situations with more parked vehicles on either side of the road, the ORB-SLAM2 algorithm is marginally more accurate than the SLAM technique used in this paper.

Table 4. RMSE comparison results.

Serial Number	RMSE (m)	
	ORB-SLAM2	MCBM-SLAM
00	0.9466	1.0745
01	3.4375	3.3125
02	6.0386	5.7456
03	0.3011	0.2745
04	0.1849	0.1645
05	0.6173	0.6352
06	1.3647	1.3746
07	0.4165	0.3625
08	6.6482	6.6901
09	2.6057	2.6845
10	2.050	1.9150

5. Conclusions

This paper introduces a SLAM system that utilizes deep learning to enhance positioning accuracy and reduce errors in dynamic environments. The proposed system is highly robust and effective in handling dynamic conditions. The SLAM system in this study is an improvement on ORB-SLAM2, based on motion constraints and a modified Mask R-CNN network to reject dynamic feature points. First, in order to fix the issue where dynamic objects interfere with SLAM's ability to localize accurately, a deep learning network is used for mask extraction of dynamic objects. Since the Mask R-CNN network is prone to incomplete segmentation when segmenting the mask, a spatial attention module is added to the Mask R-CNN network as a way to enhance the integrity of the mask

segmentation and to perform an initial estimation of the bit pose. Secondly, we propose an image mismatch rejection algorithm incorporating adaptive edge distance with grid motion statistics to efficiently reject mismatched points and further improve the image matching quality. Finally, static feature points on potentially dynamic objects are re-added using motion constraints and cardinality distributions, and the positional estimation is optimized. Experimental results on KITTI dataset sequences show that the SLAM algorithm in this paper has better localization accuracy than ORB-SLAM2 in highly dynamic sequences. According to the TUM-RGBD experimental findings, the SLAM algorithm in this paper has an average localization accuracy of 6.24% better than Dyna SLAM. This visually proves that the paper's SLAM algorithm is more robust. While the SLAM algorithm discussed in this paper has some limitations, it is worth noting that the Mask R-CNN network's spatial attention mechanism does enhance the rejection of dynamic objects. Although the SLAM algorithm in this paper can deal with dynamic objects, the dynamic objects dealt with are limited to dynamic objects with a priori information, so the fusion of IMU information is considered to assist in future research. Meanwhile, the deep learning network can be considered a separate thread to improve operation efficiency in future research. An optimal path-like approach can be considered in future research to reduce the dynamic object detection time and improve the real-time performance of the system.

Author Contributions: Methodology, X.W.; software, X.W.; validation, X.W. and X.Z.; formal analysis, X.W. and X.Z.; investigation, X.W. and X.Z.; resources, X.W. and X.Z.; data curation, X.W.; writing—original draft preparation, X.W.; writing—review and editing, X.W.; visualization, X.W.; supervision, X.Z.; project administration, X.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data cannot be disclosed due to privacy.

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

Abbreviations

Abbreviations in the Abstract	Full Name
SLAM	Simultaneous locali localization and mapping
Mask R-CNN	Mask Region-CNN
KITTI	Karlsruhe Institute of Technologyand Toyota Technological Institute
TUM-RGBD	The RGB-D dataset proposed by the tum Computer Vision Group
ORB-SLAM2	Simultaneous localization and mapping algorithm based on ORB features
Dyna-SLAM	Simultaneous localization and mapping algorithm in a dynamic environment

References

1. Mur-Artal, R.; Tardós, J.D. Orb-Slam2: An Open-Source Slam System for Monocular, Stereo, and Rgb-d Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
2. 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 Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
3. Schenk, F.; Fraundorfer, F. RESLAM: A Real-Time Robust Edge-Based SLAM System. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 154–160.
4. Su, P.; Luo, S.; Huang, X. Real-Time Dynamic SLAM Algorithm Based on Deep Learning. *IEEE Access* **2022**, *10*, 87754–87766. [CrossRef]
5. Bescos, B.; Fácil, J.M.; Civera, J.; Neira, J. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4076–4083. [CrossRef]
6. Yu, C.; Liu, Z.; Liu, X.-J.; Xie, F.; Yang, Y.; Wei, Q.; Fei, Q. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1168–1174.

7. Xing, Z.; Zhu, X.; Dong, D. DE-SLAM: SLAM for Highly Dynamic Environment. *J. Field Robot.* **2022**, *39*, 528–542. [CrossRef]
8. Zhong, Y.; Hu, S.; Huang, G.; Bai, L.; Li, Q. WF-SLAM: A Robust VSLAM for Dynamic Scenarios via Weighted Features. *IEEE Sens. J.* **2022**, *22*, 10818–10827. [CrossRef]
9. Lyu, L.; Ding, Y.; Yuan, Y.; Zhang, Y.; Liu, J.; Li, J. Doc-Slam: Robust Stereo Slam with Dynamic Object Culling. In Proceedings of the 2021 7th International Conference on Automation, Robotics and Applications (ICARA), Prague, Czech Republic, 4–6 February 2021; pp. 258–262.
10. Yin, H.; Li, S.; Tao, Y.; Guo, J.; Huang, B. Dynam-SLAM: An Accurate, Robust Stereo Visual-Inertial SLAM Method in Dynamic Environments. *IEEE Trans. Robot.* **2022**, *39*, 289–308. [CrossRef]
11. Li, A.; Wang, J.; Xu, M.; Chen, Z. DP-SLAM: A Visual SLAM with Moving Probability towards Dynamic Environments. *Inf. Sci.* **2021**, *556*, 128–142. [CrossRef]
12. Cheng, S.; Sun, C.; Zhang, S.; Zhang, D. SG-SLAM: A Real-Time RGB-D Visual SLAM towards Dynamic Scenes with Semantic and Geometric Information. *IEEE Trans. Instrum. Meas.* **2022**, *72*, 7501012. [CrossRef]
13. Ni, J.; Wang, X.; Gong, T.; Xie, Y. An Improved Adaptive ORB-SLAM Method for Monocular Vision Robot under Dynamic Environments. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 3821–3836. [CrossRef]
14. Wang, C.; Luo, B.; Zhang, Y.; Zhao, Q.; Yin, L.; Wang, W.; Su, X.; Wang, Y.; Li, C. DymSLAM: 4D Dynamic Scene Reconstruction Based on Geometrical Motion Segmentation. *IEEE Robot. Autom. Lett.* **2020**, *6*, 550–557. [CrossRef]
15. Chen, J.; Xie, F.; Huang, L.; Yang, J.; Liu, X.; Shi, J. A Robot Pose Estimation Optimized Visual SLAM Algorithm Based on CO-HDC Instance Segmentation Network for Dynamic Scenes. *Remote Sens.* **2022**, *14*, 2114. [CrossRef]
16. Wu, W.; Guo, L.; Gao, H.; You, Z.; Liu, Y.; Chen, Z. YOLO-SLAM: A Semantic SLAM System towards Dynamic Environment with Geometric Constraint. *Neural Comput. Appl.* **2022**, *34*, 6011–6026. [CrossRef]
17. Zhang, R.; Zhang, X. Geometric Constraint-Based and Improved YOLOv5 Semantic SLAM for Dynamic Scenes. *ISPRS Int. J. Geo-Inf.* **2023**, *12*, 211. [CrossRef]
18. Gou, R.; Chen, G.; Yan, C.; Pu, X.; Wu, Y.; Tang, Y. Three-Dimensional Dynamic Uncertainty Semantic SLAM Method for a Production Workshop. *Eng. Appl. Artif. Intell.* **2022**, *116*, 105325. [CrossRef]
19. Liu, Y.; Wu, Y.; Pan, W. Dynamic RGB-D SLAM Based on Static Probability and Observation Number. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 8503411. [CrossRef]
20. Zhang, Y.; Liu, R.; Yu, H.; Zhou, B.; Qian, K. Visual Loop Closure Detection with Instance Segmentation and Image Inpainting in Dynamic Scenes Using Wearable Camera. *IEEE Sens. J.* **2022**, *22*, 16628–16637. [CrossRef]
21. Dai, W.; Zhang, Y.; Li, P.; Fang, Z.; Scherer, S. Rgb-d Slam in Dynamic Environments Using Point Correlations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *44*, 373–389. [CrossRef]
22. Rosinol, A.; Violette, A.; Abate, M.; Hughes, N.; Chang, Y.; Shi, J.; Gupta, A.; Carlone, L. Kimera: From SLAM to Spatial Perception with 3D Dynamic Scene Graphs. *Int. J. Robot. Res.* **2021**, *40*, 1510–1546. [CrossRef]
23. Cheng, J.; Zhang, H.; Meng, M.Q.-H. Improving Visual Localization Accuracy in Dynamic Environments Based on Dynamic Region Removal. *IEEE Trans. Autom. Sci. Eng.* **2020**, *17*, 1585–1596. [CrossRef]
24. Han, S.; Xi, Z. Dynamic Scene Semantics SLAM Based on Semantic Segmentation. *IEEE Access* **2020**, *8*, 43563–43570. [CrossRef]
25. Zang, Q.; Zhang, K.; Wang, L.; Wu, L. An Adaptive ORB-SLAM3 System for Outdoor Dynamic Environments. *Sensors* **2023**, *23*, 1359. [CrossRef]
26. Yuan, C.; Xu, Y.; Zhou, Q. PLDS-SLAM: Point and Line Features SLAM in Dynamic Environment. *Remote Sens.* **2023**, *15*, 1893. [CrossRef]
27. Zhang, X.; Wang, X.; Zhang, R. Dynamic Semantics SLAM Based on Improved Mask R-CNN. *IEEE Access* **2022**, *10*, 126525–126535. [CrossRef]
28. Gong, H.; Gong, L.; Ma, T.; Sun, Z.; Li, L. AHY-SLAM: Toward Faster and More Accurate Visual SLAM in Dynamic Scenes Using Homogenized Feature Extraction and Object Detection Method. *Sensors* **2023**, *23*, 4241. [CrossRef]
29. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-Cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
30. Girshick, R. Fast R-Cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.