electronics

Article

The Method of Static Semantic Map Construction Based on Instance Segmentation and Dynamic Point Elimination

Jingyu Li [1,†], Rongfen Zhang [1,*,†], Yuhong Liu [1,†], Zaiteng Zhang [1,†], Runze Fan [1,†] and Wenjiang Liu [2,†]

[1] College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China; joyfourier@163.com (J.L.); liuyuhongyx@sina.com (Y.L.); zzteng0466@163.com (Z.Z.); davidfrz1996@126.com (R.F.)
[2] School of Information, Guizhou University of Finance and Economics, Guiyang 550025, China; liuwj@mail.gzufe.edu.cn
* Correspondence: rfzhang@gzu.edu.cn
† These authors contributed equally to this work.

Abstract: Semantic information usually contains a description of the environment content, which enables mobile robot to understand the environment and improves its ability to interact with the environment. In high-level human–computer interaction application, the Simultaneous Localization and Mapping (SLAM) system not only needs higher accuracy and robustness, but also has the ability to construct a static semantic map of the environment. However, traditional visual SLAM lacks semantic information. Furthermore, in an actual scene, dynamic objects will reduce the system performance and also generate redundancy when constructing map. These all directly affect the robot's ability to perceive and understand the surrounding environment. Based on ORB-SLAM3, this article proposes a new algorithm that uses semantic information and the global dense optical flow as constraints to generate dynamic-static mask and eliminate dynamic objects. Then, to further construct a static 3D semantic map under indoor dynamic environments, a fusion of 2D semantic information and 3D point cloud is carried out. The experimental results on different types of dataset sequences show that, compared with original ORB-SLAM3, both Absolute Pose Error (APE) and Relative Pose Error (RPE) have been ameliorated to varying degrees, especially on freiburg3-walking-xyz, the APE reduced by 97.78% from the original average value of 0.523, and RPE reduced by 52.33% from the original average value of 0.0193. Compared with DS-SLAM and DynaSLAM, our system improves real-time performance while ensuring accuracy and robustness. Meanwhile, the expected map with environmental semantic information is built, and the map redundancy caused by dynamic objects is successfully reduced. The test results in real scenes further demonstrate the effect of constructing static semantic maps and prove the effectiveness of our algorithm.

Keywords: simultaneous localization and mapping; instance segmentation network; dynamic point elimination; static semantic map

Citation: Li, J.; Zhang, R.; Liu, Y.; Zhang, Z.; Fan, R.; Liu, W. The Method of Static Semantic Map Construction Based on Instance Segmentation and Dynamic Point Elimination. *Electronics* **2021**, *10*, 1883. https://doi.org/10.3390/electronics10161883

Academic Editors: Donghoon Kim, Henzeh Leeghim and Jae Hyun Jin

Received: 25 June 2021
Accepted: 4 August 2021
Published: 5 August 2021

1. Introduction

SLAM (Simultaneous Localization and Mapping) is a method for intelligent mobile devices to locate the pose and build map of the surrounding environment in unknown scenes. It is widely used in many fields, such as unmanned driving, robot, and AR (Augmented Reality). A typical SLAM framework is mainly composed of front-end odometry, back-end pose optimization, and loop detection. According to the different types of sensors used by SLAM system to obtain environmental data, it can be roughly divided into: Laser SLAM using lidar, whose front end is laser odometry and Visual SLAM using camera, and its front end is visual odometry [1]. The sensor used by Visual SLAM is not only low in price, but also more intuitive to obtain environmental content. In recent years, vision-based SLAM solutions have been fully developed, such as MonoSLAM [2], ORB-SLAM2 [3], and ORB-SLAM3 [4].

However, these schemes assume that the unknown environment is static, while real scenes often contain both dynamic objects and static objects. The visual odometry is a functional module that processes sensor data, performs feature selection, extraction, and matching, and obtains pose optimization and partial mapping results in a short time [5]. Unluckily, the movement of dynamic objects in the environment will directly affect the results of visual odometry feature selection, extraction, matching, and data association [6], ultimately affect the performance of the entire system. At the same time, for the needs of human–computer interaction, the problem of Visual SLAM is not only about pose positioning and construction of environmental consistency maps. Many practical applications of Visual SLAM such as robotic home care often require higher system accuracy, robustness, and a semantic map with perceptual information that can provide the robot with more higher-level environmental information and help complete complex interactive tasks [7].

With the continuous development of deep learning, some classic deep learning networks have been proposed, such as CNN [8], R-CNN [9], SegNet [10], Faster R-CNN [11], Mask R-CNN [12], and so on. Combining deep learning network with vSLAM can help robots perceive scenes from both geometric and semantic levels, abstractly understand and cognize the environmental content, and obtain high-level perception of the environment. Compared with the simple target detection and recognition networks commonly used, such as YOLOv3 [13] and SSD [14], the semantic segmentation network has great advantages in two aspects when worked with Visual SLAM. On the one hand, semantic segmentation can obtain a more accurate outline of the target rather than the rectangular frame where the target location is. It means that, in the process of Visual SLAM dynamic point elimination, a relatively accurate prior dynamic object range can be provided to avoid the loss of tracking accuracy or tracking failure caused by excessive feature points being eliminated. On the other hand, semantic segmentation can be used to directly obtain the 2D semantic information in the scene, which is convenient for the construction of an environment map integrated with semantic information. Some related solutions are based on ORB-SLAM [15] or ORB-SLAM2, due to the novel ORB-SLAM3 algorithm being formally reported this year. ORB-SLAM3 [4] mainly proposes a new visual inertia navigation and multi-map fusion algorithm, as well as pinhole and fisheye camera models, and it proposed a maximal probability map based on a close combination of features. Therefore, its performance has been greatly improved compared to the previous version. In general, the ORB-SLAM3 algorithm is more mature than the previous version, which will promote the engineering landing development of Visual SLAM to a certain extent.

The present work of this article mainly revolves around ORB-SLAM3. In order to improve the performance of ORB-SLAM3 in a dynamic scene, we propose a new method to improve the accuracy and robustness of visual odometry in a dynamic scene, and, for the goal of providing environmental semantic information, we further construct an indoor static semantic map. The main contributions are as described in the following three points:

1. On the basis of ORB-SLAM3, we use multiple concurrency technology to add an instance segment thread. This thread uses FPN(Feature Pyramid Network) [16]+ Mask R-CNN network and is written in C++ language to extract the semantic information of image frames. Since the main language style of ORB-SLAM3 is C++, this makes the modules of the system become orderly and harmonious.
2. We propose a new method of combining with a deep learning FPN+Mask R-CNN network with global dense optical flow to obtain semantic information and eliminate the dynamic points in objects under the dynamic scene, which solves the redundant tracking problem of visual odometry and improves the accuracy and robustness of ORB-SLAM3 in dynamic scene effectively.
3. Our system integrates 2D semantic information and 3D point cloud to construct a semantic map with perceptual information, further improving the robot's ability to perceive and understand the surrounding environment.

In the rest of this article, the structure is as follows: Section 2 provides some related work in improving the performance of visual odometry, reducing the impact of dynamic

objects, and constructing perceptible semantic map. Section 3 describes the design and implementation of our SLAM system. Section 4 provides the performance of our system in a dataset and real scene to illustrate the effectiveness of our system. Finally, the work of this article is summarized and discussed in Section 5.

2. Related Work

In practical applications, the accuracy and robustness of the visual odometry is very important to the Visual SLAM system, and the construction of a perceptible environment map is also an indispensable condition for high-level interaction.

2.1. Improvement of Visual Odometry Performance

In order to improve the performance of the visual odometry, some algorithms are proposed. For example, Cui [17] used image intensity for data association in common frames and use photometric calibration for accuracy and robustness. Zhang [18] used a method that matched lines and computed a collinear relationship of points to assist bundle adjustment, and then modify perspective-n-point to improve the tracking accuracy under a poorly textured situation. Konstantinos-Nektarios [19] proposed a novel visual semantic odometry framework to enable medium-term continuous tracking of points using semantics. Zhu [20] fused a purely event-based tracking algorithm with an inertial measurement unit, to provide accurate metric tracking of a camera's full 6 DOF pose. As for ORB-SLAM3 [4], it proposes a feature-based tightly coupled visual inertial navigation system, which completely relies on the maximum posterior estimation, so that the system can run robustly in real time indoors or outdoors, and is more accurate than ORB-SLAM2 by two to five times.

2.2. Visual SLAM in a Dynamic Scene

The emergence of a dynamic object will not only affect the accuracy of pose tracking and increase the extra computational burden, but also bring inconvenience to advanced applications such as robot navigation and interaction in practical applications. Therefore, it is necessary to eliminate the influence brought by dynamic factors. For example, Chao [21] used the semantic segmentation network SegNet and LK sparse optical flow combined with motion consistency to eliminate dynamic objects. Berta [22] used instance segment network Mask R-CNN and Multi-view geometry to eliminate dynamic objects. However, the speed of the multi-view geometry and the accuracy of the sparse optical flow algorithm are often unsatisfactory. Deyvid [23] used scene flow to propagate dynamic objects within the map. Palazzolo [24] used the residuals obtained after an initial registration, together with the explicit modeling of free space in the model. Rünz [25] used a multiple model fitting approach where each object can move independently from the background and still be effectively tracked. Their methods have certain requirements for computing power and hardware, and our ideal compromise is to use the global dense optical flow combined with semantic information.

2.3. Semantic Information of Maps

Perceivable semantic maps are essential for robots to complete interactive behaviors, and there are different ways to give semantic information to maps. For example, Weinmann [26] use the methods of neighborhood selection, feature extraction, feature selection, and classification to directly segment the point cloud to obtain semantics. Qi [27] used YOLOv3 to obtain object types and contour to construct environmental label maps, while Guan [28] used semantic information to process point clouds and objects to construct real-time semantic maps. Yue [29] used multi-robot collaboration, and the local semantic maps are shared among robots for global semantic map fusion. Qin [30] used robust semantic features, inertial measurement unit, and wheel encoders to generate a global visual semantic map. Wei [31] used instance networks and built instance-oriented 3D semantic maps directly from images acquired by the RGB-D camera. As previously men-

tioned, semantic segmentation has great advantages when obtaining semantic information compared to other ways. In this work, we will fuse 2D semantic information obtained from Mask R-CNN instance segment networks into 3D point clouds to endow the map semantic information.

3. System Description

In this section, our Visual SLAM algorithm will be introduced in detail. Section 2 includes four aspects: first is the main framework of the system; second is the instance segmentation network; third is the dynamic point eliminate algorithm; and last is the method of constructing static semantic map.

3.1. System Components

As mentioned in Section 2, the performance of ORB-SLAM3 algorithm is better than ORB-SLAM2. Therefore, our algorithm solution chooses ORB-SLAM3 as the basic framework. However, ORB-SLAM3 does not have good robustness in dynamic scenes. In order to reduce the impact of dynamic object on the accuracy and robustness, we designed the system to eliminate dynamic objects firstly, and then, we further built static semantic maps in indoor dynamic scenes. The main framework of system is shown in Figure 1. The purple part is our improvement point, the yellow part is our work on the choice and deployment of the instance segmentation network, the green part is the final output of our system, and, due to ORB-SLAM3 building a sparse point cloud map, the orange part is the added dense point cloud construction algorithm [32].

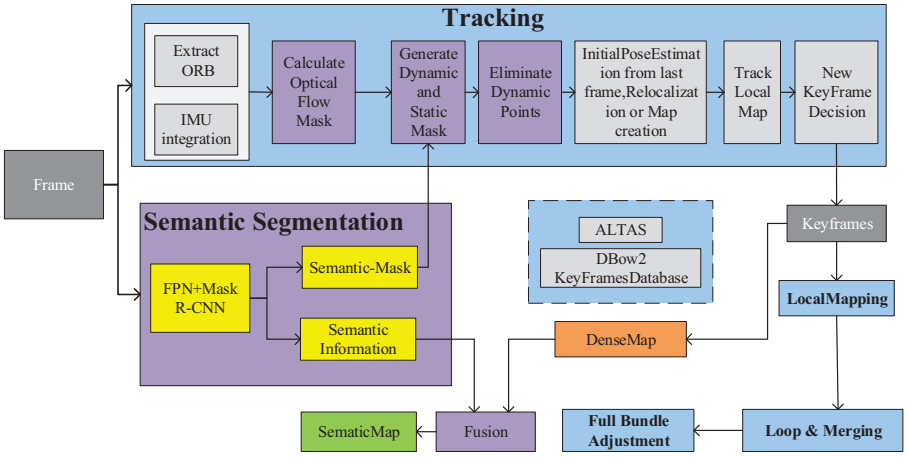


Figure 1. The main framework of the system.

As shown in Figure 1, there are mainly five threads, namely Tracking thread, Loop Mapping thread, Loop Closing thread, Full Bundle Adjustment thread, and Semantic Segmentation thread. When the image frame of the current scene (include RGB image and Depth image) is obtained, the RGB image is simultaneously passed to the Tracking thread and Semantic Segmentation thread to extract feature points and semantic information. Then, the global dense optical flow mask of the RGB image is calculated to obtain the actual dynamic object information and then further combined the dynamic information with the prior semantic results obtained from semantic segmentation; a mutually constrained dynamic-static mask is then formed. Subsequently, it follows the process of using the mask to eliminate dynamic objects among feature points to obtain keyframes and calculate dense point cloud maps. In addition, the final work is fusing the semantic information and the dense point cloud to generate the static semantic map.

3.2. Semantic Segmentation

In order to obtain the semantic information in the scene, we use the instance segmentation network Mask R-CNN to segment the semantic information in the Semantic

Segmentation thread, and rewrite it into C + + style when deploying it to our system. The network structure is shown in Figure 2.

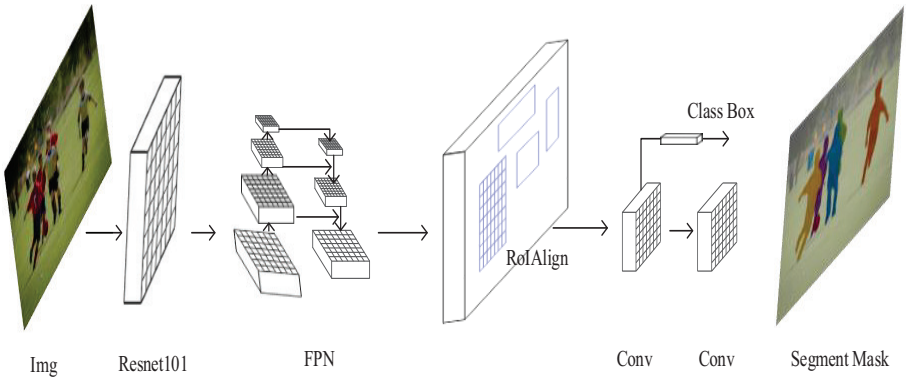


Figure 2. FPN + Mask R-CNN network framework.

Deep features of images usually have rich semantic information. To obtain accurate target recognition results in instance segmentation, in the first stage of Mask R-CNN network, Resnet101 [33] is selected as the feature extraction layer to extract the basic features of images. In addition, considering different scales of large, medium, and small targets that may appear in the actual environment when constructing semantic map, after feature extraction, the FPN network was further selected to perform jump-connection fusion between the bottom layer and the top layer of extracted features, and the basic structure of the network was adjusted according to the actual demand, so that the Mask R-CNN has better semantic segmentation accuracy and can recognize up to 80 categories on the COCO dataset [34].

3.3. Dynamic Points Elimination

In the dynamic points' elimination process, we use the method of optical flow estimation to detect dynamic objects in the scene to obtain actual dynamic information. The optical flow method is divided into sparse optical flow method and dense optical flow method. It is a two-dimensional pixel detection processing method, which uses the changes of pixels in the time domain combined with the correlation between neighboring frames to calculate the corresponding relationship between the previous frame and the current frame, so as to obtain the motion information of the object. Based on the general principle of optical flow method, as shown in Figure 3, the dense optical flow algorithm proposed by Gunner Farneback used the pixel points in the two image frames before and after to perform motion estimation, and its effect is better than that of the sparse optical flow algorithm [35].

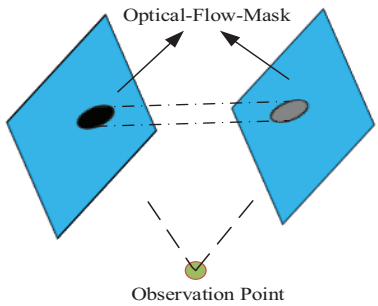


Figure 3. Optical flow estimation.

While eliminating dynamic information, most useful static information should be retained to ensure the accuracy and reliability of the tracking process. In the present work, in order to obtain a higher detection accuracy, we use the global dense optical flow method to detect dynamic objects in the image frame, and set a small threshold to detect small

scale moving targets. At the same time, the subsampling operation is used to improve the detection speed. Then, considering the problem of motion noise, this article takes the optical flow constraint as a soft threshold condition, which will be combined with the prior semantic information of the semantic thread to obtain further constraints. The specific algorithm flow is as follows:

(1) Carry out semantic segmentation to obtain the priori Semantic-Mask of dynamic objects, and calculate the dense optical flow to obtain the Optical-Flow-Mask generated by the actual movement of the object.

(2) Traverse the pixels of Semantic-Mask and determine whether each point has dynamic information in the corresponding 3×3 area in the Optical-Flow-Mask.

(2.1) If Semantic-Mask (i,j) is a prior dynamic point, and the optical flow information appears in the corresponding region, then the pixel of this point belongs to the dynamic object region;

(2.2) If Semantic-Mask (i,j) is a prior dynamic point, and no optical flow information appears in its corresponding region, then the pixel of this point belongs to the prior dynamic object region;

(2.3) If Semantic-Mask (i,j) is a prior static point, and the optical flow information appears in the corresponding region, then the pixel of this point belongs to the dynamic object region;

(2.4) If Semantic-Mask (i,j) is a prior static point, and no optical flow information appears in the corresponding region, then the pixel of this point belongs to the static target region;

(3) Fuse pixels in all dynamic areas to generate the final dynamic-static mask;

(4) Combine the dynamic-static information of the mask to judge the previously extracted feature points, if the feature point belongs to the dynamic area in the mask, the feature point will be eliminated.

3.4. Static Semantic Map Construction

The original ORB-SLAM3 generates sparse point clouds. In our system, after the dynamic points is eliminated, the keyframes is obtained, and then the dense point cloud through the keyframe is further calculated. In the end, the obtained 2D semantic information is fused with a 3D dense point cloud to build a static 3D semantic dense point cloud map. The algorithm flow is shown in Figure 4.

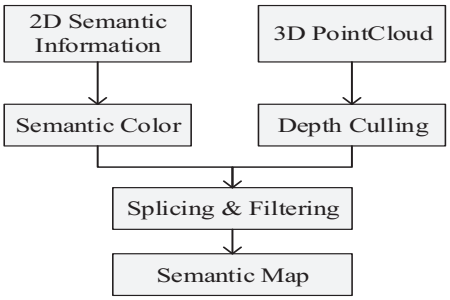


Figure 4. Semantic point cloud generation.

In the fusion of 2D semantic information and 3D point cloud, the points of 3D point cloud with inappropriate distance will be eliminated according to the depth information at firstly. Then, the semantic colors of different objects from the acquired 2D semantic information are extracted, and the semantic color information according to the coordinate index corresponding to the depth value is obtained. Finally, the semantic color information will be added to 3D space points of corresponding depth values, and gives the point cloud with semantic information. The entire semantic map construction process is carried out on the basis of dynamic object elimination. In this way, the redundancy of map information brought by the dynamic object participating in the mapping is avoided, and the static map in the actual scene is restored to a certain extent. That is, through the fusion processing

of the 3D point cloud, the point cloud is endowed with semantic information, and a perceptible static semantic map of the indoor dynamic environment is generated.

4. Experimental Results

In order to test the actual effect of our algorithm, we used two types of scenes in the TUM dataset [36] (high dynamic scenes and low dynamic scenes).

Generally, when using the TUM dataset, APE (Absolute Pose Error) and RPE (Relative Pose Error) are used to evaluate the robustness and accuracy of visual odometry. APE represents the global trajectory consistency, the smaller the value, the higher the consistency, and the better the robustness of the system. RPE is used to measure the drift degree of rotation and transformation process, and, the smaller the drift is, the more accurate the system is [36]. Our experiments also use APE and RPE to analyze and compare the estimated trajectory and the real trajectory. In addition, then, we calculate APE and RPE to get a result including RMSE (Root Mean Square Error), Median Error, Mean Error, and S.D. (Standard Deviation).

Our experimental environment and conditions are as follows: Laptop, Ubuntu16.04, Inter(R)Core(TM)i5-9300H@CPU2.4GHz, RAM-16 GB, and GPU-GTX1650.

In Section 4.1, we firstly verify and analyze the effectiveness of the dynamic point eliminate algorithm; In Section 4.2, we compare and analyze the performance of our SLAM system to construct a static semantic map on the dataset. Finally, the actual mapping effect is shown through the real scene experiment in Section 4.3.

4.1. Dynamic Object Eliminating Experiment

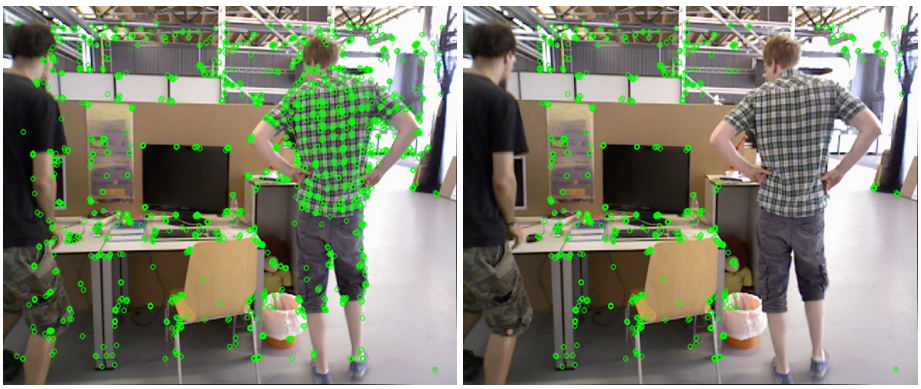
Figures 5 and 6 present the actual effect of the instance segmentation network and the dynamic point eliminating algorithm used in this article on the dataset.

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

Figure 5. Segmentation effect: (**a**) the raw image; (**b**) semantic segmentation result.

In this scene, there are two people walking around the desk. The left image in Figure 5 is the raw image of a certain frame in the scene, and the right image is the result of semantic segmentation. The left image in Figure 6 shows the feature point distribution in a certain frame of tracking. It can be seen that there are a large number of feature points on the moving person. The right image in Figure 6 is the feature point distribution after applying the algorithm in this article to eliminate the selected dynamic points.

For further analysis and comparison of the above experimental results, we calculated APE and RPE between the estimated trajectory and the real trajectory. The experimental results are shown in Tables 1–3, where fr3 represents that the dataset sequence it belongs to is freiburg3; sitting and walking represent two different character states, sitting is low dynamic and walking is high dynamic; xyz, rpy, static, and half halfsphere stand for four types of camera ego-motions [36]. For example, sit means that the person is sitting, and xyz means the camera moves along the *x*-*y*-*z*-axis.



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

Figure 6. Dynamic point elimination effect: (**a**) the feature point distribution in a frame of tracking; (**b**) the result after eliminating dynamic points.

Table 1. Results of Absolute Pose Error (APE includes RMSE [m], Median [m], Mean [m], S.D. [m]).

Sequences	ORB-SLAM3				Our System			
	RMSE	**Median**	**Mean**	**S.D.**	**RMSE**	**Median**	**Mean**	**S.D.**
fr3-walking-xyz	0.6687	0.5124	0.5823	0.3288	0.0150	0.0110	0.0128	0.0078
fr3-walking-rpy	0.8461	0.7803	0.7738	0.3424	0.0314	0.0203	0.0256	0.0183
fr3-walking-static	0.1072	0.0788	0.0933	0.0528	0.0073	0.0059	0.0065	0.0033
fr3-walking-halfsphere	0.5939	0.4562	0.5092	0.3055	0.0180	0.0150	0.0162	0.0079
fr3-sitting-static	0.0087	0.0068	0.0075	0.0044	0.0065	0.0048	0.0056	0.0033

Table 2. Results of Relative Pose Error (RPE includes RMSE [m], Median [m], Mean [m], S.D. [m]).

Sequences	ORB-SLAM3				Our System			
	RMSE	**Median**	**Mean**	**S.D.**	**RMSE**	**Median**	**Mean**	**S.D.**
fr3-walking-xyz	0.0255	0.0163	0.0207	0.0148	0.0121	0.0080	0.0099	0.0069
fr3-walking-rpy	0.0281	0.0180	0.0221	0.0172	0.0197	0.0123	0.0153	0.0124
fr3-walking-static	0.0290	0.0065	0.0128	0.0260	0.0066	0.0051	0.0057	0.0032
fr3-walking-halfsphere	0.0236	0.0145	0.0188	0.0143	0.0128	0.0093	0.0107	0.0069
fr3-sitting-static	0.0048	0.0037	0.0041	0.0024	0.0056	0.0042	0.0049	0.0027

Table 3. Percentage of APE and RPE reduction.

Sequences	Improvements (APE)				Improvements (RPE)			
	RMSE	**Median**	**Mean**	**S.D.**	**RMSE**	**Median**	**Mean**	**S.D.**
fr3-walking-xyz	97.76%	97.85%	97.80%	97.63%	52.55%	50.92%	52.17%	53.38%
fr3-walking-rpy	96.29%	97.4%	96.69%	94.66%	29.89%	31.67%	30.77%	27.91%
fr3-walking-static	93.19%	92.51%	93.03%	93.75%	77.24%	21.54%	55.47%	87.69%
fr3-walking-halfsphere	96.97%	96.71%	96.82%	97.41%	45.76%	35.86%	43.09%	51.75%
fr3-sitting-static	25.29%	29.41%	25.33%	25.00%	-	-	-	-

Tables 1 and 2 are the results obtained through experiments with different datasets. Table 3 further illustrates the experimental results, in which the improvements represent that the obtained error after the algorithm processing in this article reduces the percentage of the original error. In addition, the percentage of average APE and RPE reduction are shown in Tables 4 and 5. In Tables 4 and 5, compared with the original ORB-SLAM3 on different types of dataset sequences of fr3-walking, after processing the dynamic point elimination algorithm, the APE is greatly reduced, and the RPE also has a more obvious

reduction. Especially on fr3-walking-xyz, APE decreases by 97.78% on average, and RPE decreases by 52.33% on average. It is noted that, since only part of the human body is moving in the low dynamic dataset, when the visual odometry is tracking, the static part of the human body still provides pose estimation information. While the ultimate goal of this article is to further construct a static semantic map by eliminating the influence of dynamic objects, our focus is on the effect of static semantic mapping at the end. Thus, we only use the low-dynamic dataset fr3-sitting-static to illustrate its effect here. Although the APE has only a small decrease and RPE has not changed much, the precision of the static semantic map in low-dynamic scene is ensured. From the point of view of the dynamic point elimination effect, our algorithm not only improves the pose accuracy, but also improves the robustness of the system to the dynamic environment, and is conducive to construct the static map consistent with the environment of the dynamic scene.

Table 4. Percentage of average APE reduction.

Sequences	ORB-SLAM3 APE	Our APE	Improvements
	Average Value	Average Value	(APE)
fr3-walking-xyz	0.5230	0.0116	97.78%
fr3-walking-rpy	0.6856	0.0239	96.51%
fr3-walking-static	0.0830	0.0057	93.13%
fr3-walking-halfsphere	0.4662	0.0142	96.95%

Table 5. Percentage of Average RPE Reduction.

Sequences	ORB-SLAM3 RPE	Our RPE	Improvements
	Average Value	Average Value	(RPE)
fr3-walking-xyz	0.0193	0.0092	52.33%
fr3-walking-rpy	0.0213	0.0149	30.04%
fr3-walking-static	0.0185	0.0051	72.43%
fr3-walking-halfsphere	0.0178	0.0099	44.38%

In order to more intuitively show the effectiveness of the dynamic point eliminating algorithm in this article and the improvement of ORB-SLAM3's pose accuracy and system robustness in a dynamic environment, we take the freiburg3-walking-xyz dataset as an example. Under the same experimental conditions, we compared the real trajectory of the dataset, the estimated trajectories of the original ORB-SLAM3, DS-SLAM, and DynaSLAM with the estimated trajectory of our algorithm. As shown in Figures 7 and 8, the real trajectory of the original dataset is groundtruth, Our represents the result of our algorithm, and the comparison among these trajectories are drawn, respectively.

In Figure 7, the estimated trajectory of the original ORB-SLAM3 deviates the most from the real trajectory and is most affected by dynamic objects; DS-SLAM improves the influence of dynamic objects to a certain extent; but the deviation between DynaSLAM, our algorithm, and the real trajectory is the smallest, effectively reducing the impact of dynamic objects. As a result, after using the algorithm proposed in this article, the estimated trajectory and the real trajectory can be well fitted, which improves the trajectory accuracy and robustness of the ORB-SLAM3, and therefore enhances the global consistency of the mapping.

In Figure 8, from the comparative analysis of the three different directions of these trajectories, we can intuitively see the deviation in different pose directions at different moments: the estimated trajectory of ORB-SLAM3 obviously has large deviations in the three directions of x, y, and z to the real trajectory; the degree of deviation of the estimated trajectory of DS-SLAM in the x, y, and z has been improved. Similarly, DynaSLAM and our algorithm fit the real trajectory well in all directions. Then, we further evaluate the

performance of DS-SLAM, DynaSLAM, and our algorithm on APE and RPE through EVO [37] (Evaluation of Odometry). The results are shown in Tables 6 and 7.

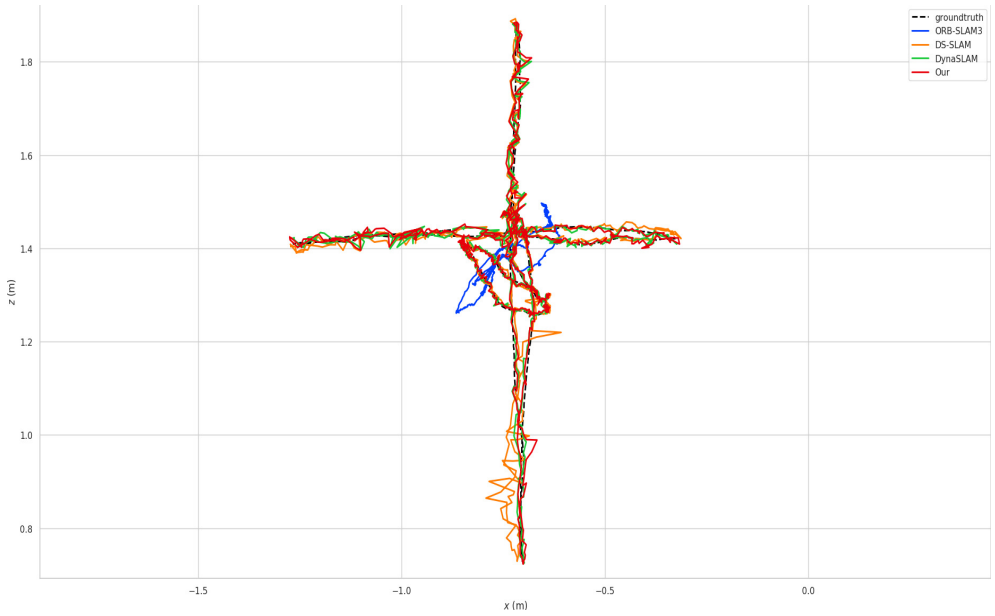


Figure 7. Comparison of trajectory.

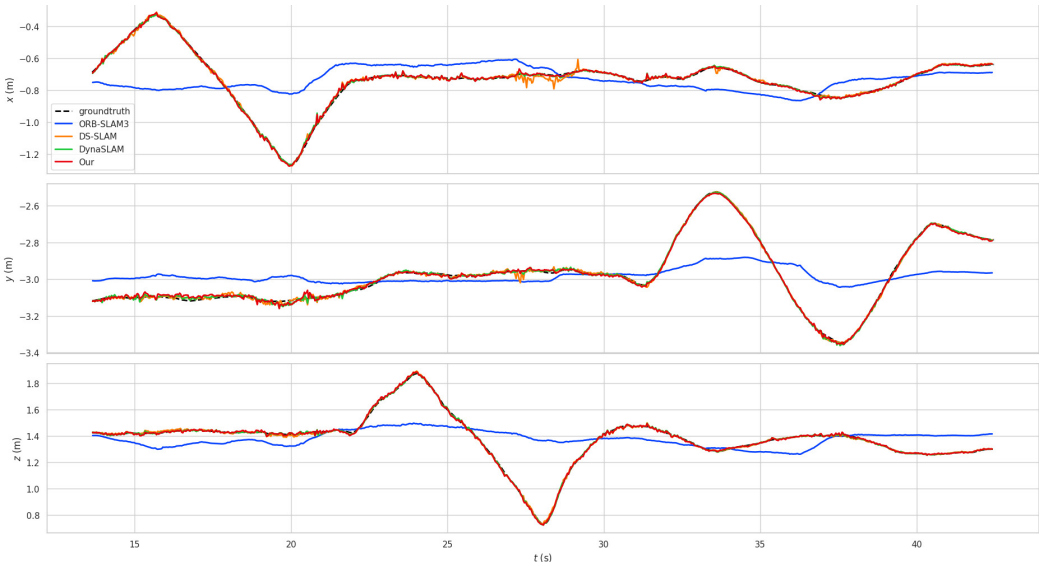


Figure 8. Comparison of tracks in different directions.

Table 6. APE comparison (APE includes RMSE [m], Median [m], Mean [m], S.D. [m]).

Algorithm	RMSE	Median	Mean	S.D.
DS-SLAM	0.0171	0.0118	0.0140	0.0098
DynaSLAM	0.0133	0.0097	0.0112	0.0071
Our	0.0150	0.0110	0.0128	0.0078

Table 7. RPE comparison (RPE includes RMSE [m], Median [m], Mean [m], S.D. [m]).

Algorithm	RMSE	Median	Mean	S.D.
DS-SLAM	0.0139	0.0080	0.0105	0.0091
DynaSLAM	0.0122	0.0082	0.0098	0.0073
Our	0.0121	0.0080	0.0099	0.0069

In addition, Table 8 shows the mean tracking time of DS-SLAM, DynaSLAM, and our SLAM system. Although the real-time performance of DS-SLAM is good, it is not as good as DynaSLAM and our SLAM system in improving the impact of dynamic objects. DynaSLAM and our system are comparable in reducing the impact of dynamic objects, and the gap between them is small in order of magnitude. However, our real-time performance is better than DynaSLAM. These results also show that our algorithm improves the robustness of ORB-SLAM3 system in dynamic scenes.

Table 8. Mean tracking time.

Algorithm	Mean Tracking Time (ms)
DS-SLAM	102.9
DynaSLAM	594.4
Our SLAM	371.6

Table 9 shows the running time consumption table of the main algorithm modules of our SLAM system. These results are obtained by averaging the time of the algorithm running 10 times. The first column is the processing time of semantic segmentation; the second column is the time required to calculate the global dense optical flow and dynamic-static mask; the third column is the time required to dynamic points elimination.

Table 9. Algorithm running time consumption.

Algorithm Name	Instance Segmentation	Compute Dynamic-Static Mask	Dynamic Points Elimination
Time (ms)	336.098	19.461	2.452

According to all of the analyses of the experimental results above, it is not difficult to find that our algorithm effectively improves the performance degradation of ORB-SLAM3 in the tracking process caused by the movement of dynamic objects. However, for the consideration of lightweight platform application, the real-time performance of the algorithm on our platform needs to be further improved.

4.2. Dataset Experiment

This section conducts experiments in a low-dynamic dataset scene (there are two people sitting on a chair in the scene, and the body is moving locally) and a high-dynamic dataset scene (there are two people walking around the desk in the scene). In order to fully compare the effect of the algorithm in the work when constructing a static semantic map, we firstly use ORB-SLAM3 to construct an original sparse point cloud map. The results are shown in Figure 9.

In Figure 9, (a) is a frame image in the low dynamic dataset, and (b) is its corresponding sparse point cloud map. (c) is a frame in the high-dynamic dataset, and (d) is its corresponding sparse point cloud map.

After the dense mapping, the mapping results of the low-dynamic dataset and the high-dynamic dataset are further compared without using the algorithm and using the algorithm of our work, as shown in Figure 10. The results without the proposed algorithm

are in the left column, and their dense point cloud maps do not have semantic information. The right column corresponds to the semantic maps when using our algorithm in the article.

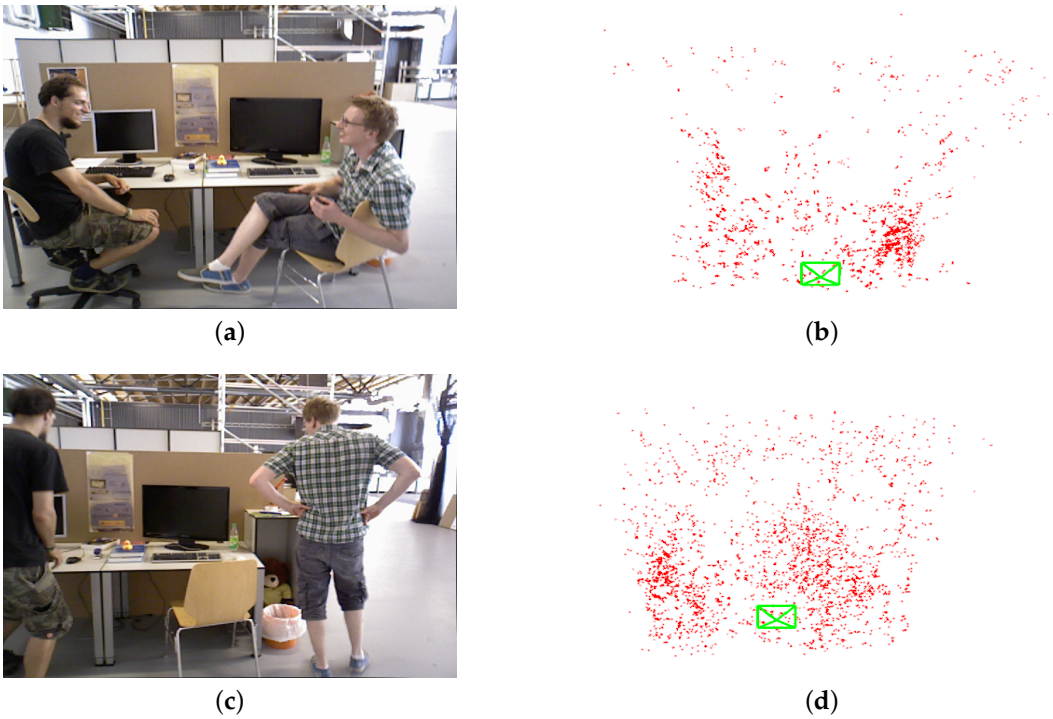


Figure 9. Original ORB-SLAM3 algorithm sparse point cloud map: (**a**) a frame image in a low dynamic dataset; (**b**) sparse point cloud map of a low dynamic dataset; (**c**) a frame in a high-dynamic dataset; (**d**) sparse point cloud map of a high-dynamic dataset.

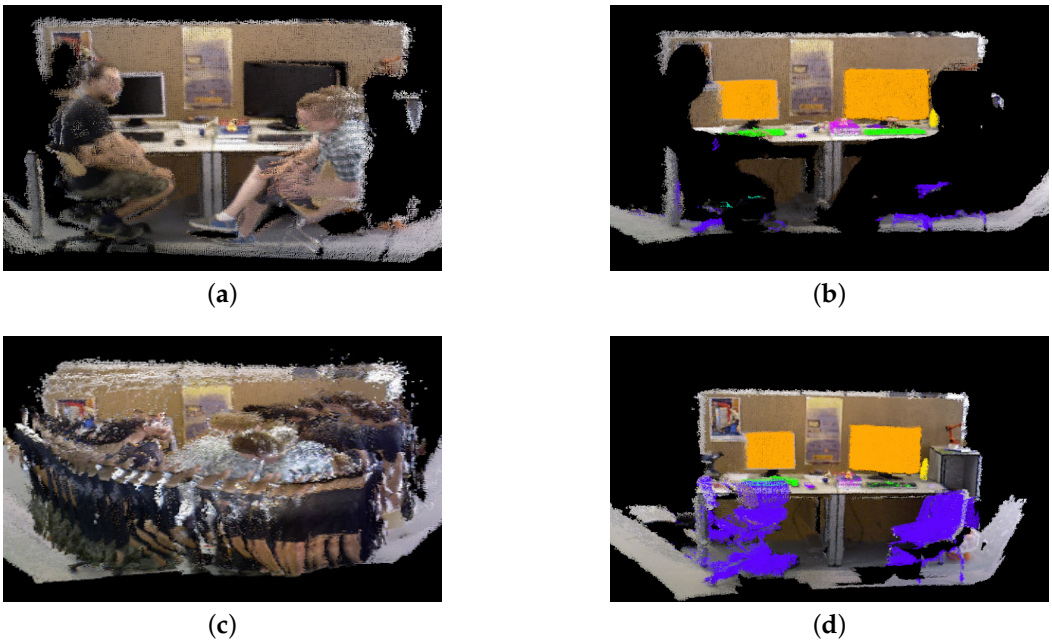
Specifically, in Figure 10, (a) is the map of dense point cloud in a low dynamic dataset not using our algorithm, in which it can be seen that the hands and heads of characters appear obviously redundant; (b) is the corresponding dynamic object eliminating and semantic mapping effect when implementing the proposed algorithm in this article. Clearly, the redundant information is reduced in the figure, and the semantic information from instance segmentation network is given to the objects in the environment. Among them, orange represents the computer screen, green represents the keyboard, yellow represents the bottle, light purple represents the book, dark purple represents the mouse, and dark blue represents the chair; (c) is the result for high dynamic scenes' dense point cloud mapping, and, due to the large movement of people in the scene, a large amount of redundancy can easily be found in the point cloud map; While compared with (c), (d) is the mapping effect after using our algorithm to eliminate the dynamic semantic characters, which effectively reduces the map redundancy, while having the semantic information of different colors. From the comparison in Figure 10, it can be concluded that the entire environment map with the elimination of dynamic object not only reduces the map redundancy information brought by dynamic object, but also has semantic information, and it can provide semantic support for high-level tasks.

4.3. Real Scene Test

Finally, we test our system in the actual laboratory. The hardware platform is as mentioned above. The depth camera is Astra-Pro, and the experimental results are shown in Figure 11.

In Figure 11, (a) shows the low dynamic scene in the laboratory (part of the figure's body moves), and (b) gives the static semantic map generated by the algorithm in this article of (a). In the figure, dark blue represents chair, orange represents the display screen,

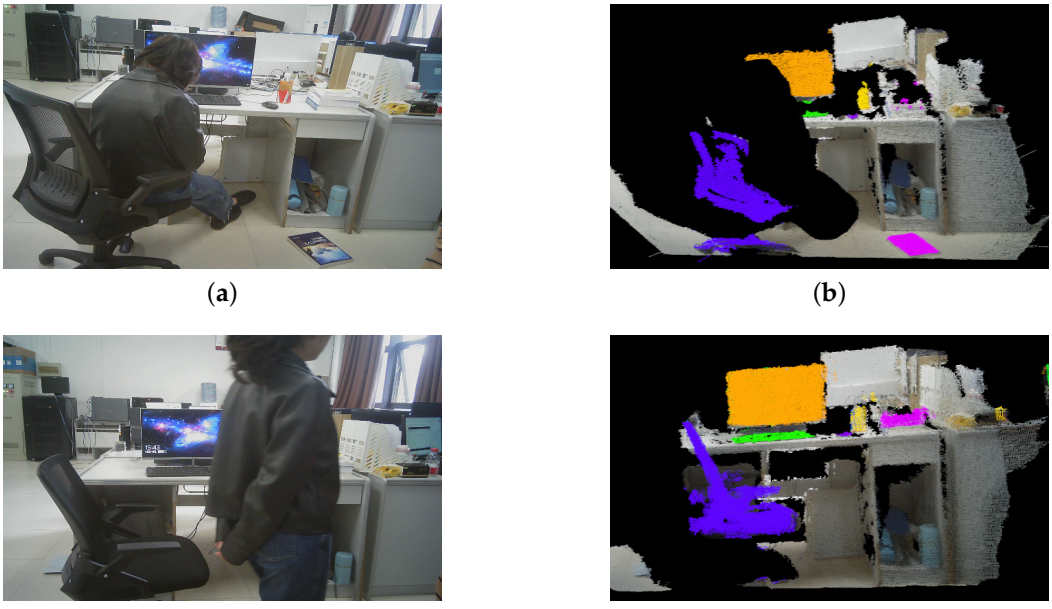
dark purple represents books, light purple represents the mouse, yellow represents the bottle, and green represents the keyboard. Similarly, (c) is a highly dynamic scene in the laboratory (the figure is walking back and forth in front of the desk), and (d) is a static semantic map constructed by the algorithm in this article about (c), whose semantic meaning is consistent with that of (b). From the analysis of the results with the dynamic point elimination algorithm running in the actual scene, it can be seen that the algorithm proposed in this article has effectively eliminated the dynamic object and completed the construction of the static semantic map.



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

(**c**) (**d**)

Figure 10. Static Semantic Map of Dynamic dataset: (**a**) dense point cloud map of low dynamic dataset; (**b**) our algorithm results in a low dynamic dataset; (**c**) dense point cloud map of a high dynamic dataset; (**d**) our algorithm results in a high dynamic dataset.



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

(**c**) (**d**)

Figure 11. Static Semantic Map of Real Scene: (**a**) a frame in low dynamic scene; (**b**) our algorithm results in a low dynamic scene; (**c**) a frame in a high dynamic scene; (**d**) our algorithm results in a high dynamic scene.

5. Conclusions and Future Work

This article explores the solution to improve robustness and accuracy of ORB-SLAM3 in dynamic scenes. Through combining semantic information and global dense optical flow to eliminate dynamic points, it reduces the influence of the visual odometry on the pose estimation caused by the dynamic object. Compared with the original ORB-SLAM3, on different types of dataset sequences, both APE and RPE have been ameliorated to varying degrees, especially on fr3-walking-xyz, the APE decreased by 97.78% from the original average value of 0.523, and the RPE decreased by 52.33% from the original average value of 0.0193. In addition, compared with DS-SLAM and DynaSLAM, it makes a trade-off between speed and performance. At the same time, the fusion method of 2D semantic information and 3D pointcloud gives the map semantic information and reduces map redundancy successfully. As a feasible way of vSLAM to perceive the surrounding environment in higher level applications, this research provides a perceivable indoor environmental map with semantics for robots to understand surroundings. In the next work, we will put more emphasis on promoting the development of vSLAM to engineering, modifying the instance segmentation network based on wavelet transform, and investigating a variety of motion detection strategies to form strong constraints on the TensorRT platform to improve the precision of dynamic eliminating and the speed of the system.

Author Contributions: Conceptualization, R.Z., Y.L., R.F. and W.L.; formal analysis, R.Z., Z.Z., R.F. and W.L.; methodology, J.L.; project administration, J.L.; software, J.L.; supervision, Y.L. and Z.Z.; writing–original draft, J.L.; writing–review and editing, R.Z., Y.L., Z.Z., R.F. and W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Guizhou Provincial Science and Technology Foundation under Grant QKHJC-ZK[2021]Key001.

Data Availability Statement: "COCO dataset" at https://cocodataset.org (accessed on 5 August 2021). "TUM dataset" at https://vision.in.tum.de/data/datasets/rgbd-dataset/download (accessed on 5 August 2021).

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

References

1. Wang, S.; Wu, Z.; Zhang, W. An Overview of SLAM. In Proceedings of the Chinese Intelligent Systems Conference, CISC 2018, Wenzhou, China, 1 January 2019; pp. 673–681.
2. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef]
3. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
4. Carlos, C.; Richard, E.; Gomez, R.J.J. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, 1–17. [CrossRef]
5. Friedrich, F.; Davide, S. Visual odometry: Part II: Matching, robustness, optimization, and applications. *IEEE Rob. Autom. Mag.* **2012**, *19*, 78–90.
6. Jorge, F.-P.; Jose, R.A.; Juan Manuel, R.-M. Visual simultaneous localization and mapping: A survey. *Artif. Intell. Rev.* **2012**, *43*, 55–81.
7. Xia, L.; Cui, J.; Shen, R.; Xu, X.; Gao, Y.; Li, X. A survey of image semantics-based visual simultaneous localization and mapping: Application-oriented solutions to autonomous navigation of mobile robots. *Int. J. Adv. Rob. Syst.* **2020**, *17*, 1729881420919185. [CrossRef]
8. Smirnov, E.A.; Timoshenko, D.M.; Andrianov, S.N. Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks. *AASRI Procedia* **2014**, *6*, 89–94. [CrossRef]
9. Ross, G.; Jeff, D.; Trevor, D.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, 23–28 June 2014; pp. 580–587.
10. Vijay, B.; Alex, K.; Roberto, C. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495.
11. Ren, S.; He, K.; Ross, G.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]

12. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the 16th IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
13. Redmon, J; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
14. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the 14th European Conference on Computer Vision, ECCV 2016, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.
15. Raul, M.-A.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Rob.* **2015**, *31*, 1147–1163.
16. Lin, T.-Y.; Piotr, D.; Ross, G.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
17. Cui, L.; Ma, C.; Wen, F. Direct-ORB-SLAM: Direct Monocular ORB-SLAM. In Proceedings of the 2nd International Conference on Computer Information Science and Application Technology, CISAT 2019, Guangzhou, China, 30 August–1 September 2019; p. 032016.
18. Zhang, F.; Rui, T.; Yang, C.; Shi, J. LAP-SLAM: A Line-Assisted Point-Based Monocular VSLAM. *Electronics* **2019**, *8*, 2079–9292. [CrossRef]
19. Lianos, K.-N.; Schonberger, J.L.; Pollefeys, M.; Sattler, T. VSO: Visual Semantic Odometry. In Proceedings of the Computer Vision-ECCV 2018—15th European Conference, Munich, Germany, 8–14 September 2018; pp. 246–263.
20. Zhu, A.Z.; Atanasov, N.; Daniilidis, K. Event-Based Visual Inertial Odometry. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5816–5824.
21. 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 2018, Madrid, Spain, 27 December 2018; pp. 1168–1174.
22. Bescos, B.; Facil, J.M.; Civera, J.; Neira, J. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robot. Autom.* **2018**, *3*, 4076–4083. [CrossRef]
23. Deyvid, K.; Aljoša, O.; Jörg, S.; Leibe, B. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 1785–1792.
24. Palazzolo, E.; Behley, J.; Lottes, P.; Giguere, P.; Stachniss, C. ReFusion: 3D Reconstruction in Dynamic Environments for RGB-D Cameras Exploiting Residuals. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 7855–7862.
25. Rünz, M.; Agapito, L. Co-fusion: Real-time segmentation, tracking and fusion of multiple objects. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4471–4478.
26. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [CrossRef]
27. Qi, X.; Yang, S.; Yan, Y. Deep Learning Based Semantic Labelling of 3D Point Cloud in Visual SLAM. In Proceedings of the 2018 3rd International Conference on Automation, Control and Robotics Engineering, CACRE 2018, Chengdu, China, 19–22 July 2018; pp. 12–23.
28. Guan, P.; Cao, Z.; Chen, E.; Liang, S.; Tan, M.; Yu, J. A real-time semantic visual SLAM approach with points and objects. *Int. J. Adv. Rob. Syst.* **2020**, *17*, 1729881420905443. [CrossRef]
29. Yue, Y.; Zhao, C.; Wu, Z.; Yang, C.; Wang, Y.; Wang, D. Collaborative Semantic Understanding and Mapping Framework for Autonomous Systems. *IEEE/ASME Trans. Mechatronics* **2021**, *26*, 978–989. [CrossRef]
30. Qin, T.; Chen, T.; Chen, Y.; Su, Q. AVP-SLAM: Semantic Visual Mapping and Localization for Autonomous Vehicles in the Parking Lot. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 5939–5945.
31. Li, W.; Gu, J.; Chen, B. Incremental Instance-Oriented 3D Semantic Mapping via RGB-D Cameras for Unknown Indoor Scene. *Discret. Dyn. Nat. Soc.* **2020**, *2020*, 2528954.
32. ORBSLAM2_with_Pointcloud_Map. Available online: https://github.com/gaoxiang12/ORBSLAM2_with_pointcloud_map (accessed on 19 July 2021).
33. Andreas, T.; Karl-Peter, F.; Hannes, F. REST-Net: A dynamic rule-based IDS for VANETs. In Proceedings of the 7th IFIP Wireless and Mobile Networking Conference, WMNC 2014, Vilamoura, Portugal, 20–22 May 2014; pp. 1–8.
34. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In Proceedings of the 13th European Conference on Computer Vision, ECCV 2014, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
35. Farnebäck, G. Polynomial Expansion for Orientation and Motion Estimation. Ph.D. Dissertation, Linköping University Electronic Press, Linköping, Sweden, 2002.
36. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 25th IEEE/RSJ International Conference on Robotics and Intelligent Systems, IROS 2012, Vilamoura, Algarve, Portugal, 7–12 October 2012; pp. 573–580.
37. evo. Available online: https://github.com/MichaelGrupp/evo (accessed on 31 July 2021).