machines

MDPI

Article

Robust Tracking and Clean Background Dense Reconstruction for RGB-D SLAM in a Dynamic Indoor Environment

Fengbo Zhu, Shunyi Zheng *, Xia Huang and Xiqi Wang

School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
* Correspondence: syzheng@whu.edu.cn

Abstract: This article proposes a two-stage simultaneous localization and mapping (SLAM) method based on using the red green blue-depth (RGB-D) camera in dynamic environments, which can not only improve tracking robustness and trajectory accuracy but also reconstruct a clean and dense static background model in dynamic environments. In the first stage, to accurately exclude the interference of features in the dynamic region from the tracking, the dynamic object mask is extracted by Mask-RCNN and optimized by using the connected component analysis method and a reference frame-based method. Then, the feature points, lines, and planes in the nondynamic object area are used to construct an optimization model to improve the tracking accuracy and robustness. After the tracking is completed, the mask is further optimized by the multiview projection method. In the second stage, to accurately obtain the pending area, which contains the dynamic object area and the newly added area in each frame, a method is proposed, which is based on a ray-casting algorithm and fully uses the result of the first stage. To extract the static region from the pending region, this paper designs divisible and indivisible regions process methods and the bounding box tracking method. Then, the extracted static regions are merged into the map using the truncated signed distance function method. Finally, the clean static background model is obtained. Our methods have been verified on public datasets and real scenes. The results show that the presented methods achieve comparable or better trajectory accuracy and the best robustness, and can construct a clean static background model in a dynamic scene.

Keywords: dynamic scene; semantic SLAM; RGB-D; mask refinement; background reconstruction

Citation: Zhu, F.; Zheng, S.; Huang, X.; Wang, X. Robust Tracking and Clean Background Dense Reconstruction for RGB-D SLAM in a Dynamic Indoor Environment. *Machines* **2022**, *10*, 892. https://doi.org/10.3390/machines10100892

Academic Editor: Antonios Gasteratos

Received: 7 August 2022
Accepted: 21 September 2022
Published: 3 October 2022

1. Introduction

Visual simultaneous localization and mapping (SLAM) plays an important role in autonomously unknown environment exploration and map construction using robots and other equipment [1,2]. Currently, most SLAM systems work well on scenes based on static assumptions [3], but in real scenes, there will inevitably be moving objects, and moving objects will bring great challenges to SLAM. On the one hand, dynamic objects may cause errors of data association during tracking, thereby causing tracking and mapping to fail. On the other hand, it is difficult to accurately distinguish dynamic regions from static backgrounds, and some dynamic objects will be reconstructed in the map, thus leading to mapping errors. Such a map cannot be used for robot navigation.

Most of the improved SLAM methods for dynamic scenes are based on ORB-SLAM2 [4]. The idea of these improved methods is to exclude the dynamic feature points and use only static background points for tracking and mapping. In traditional methods, the motion consistency check method is used to distinguish between dynamic points and static points, but due to the lack of prior information on dynamic objects, tracking will degenerate when the dynamic objects move at a low speed. Currently, image segmentation [5,6] and object detection methods [7] based on deep learning are used to recognize dynamic objects, and the feature points in the dynamic area are filtered out. Some scholars combine deep learning and geometric methods to further improve tracking accuracy; these methods can eliminate

most of the dynamic points, and the rare remaining dynamic points can be removed by using bundle adjustment optimization, but when the dynamic object occupies a larger foreground or a larger part of the image, the points in the dynamic region are removed and the remaining few unevenly distributed static points may cause unstable tracking; in these cases, some structural information in the scene can be used to help improve tracking accuracy and robustness.

A clean background map can make robot navigation much more convenient. Currently, there are relatively few studies on reconstructing the dense static background of dynamic scenes. Dynamic scenes are often more complex than static scenes, and reconstructing a clean map without the dynamic object is a great challenge. The segmentation results of deep learning suffer from over-segmentation or under-segmentation [8]; connected component analysis based on depth information will inevitably divide the dynamic area into the static area when the dynamic object is in close contact with the static background; the Bayesian probabilistic update method cannot reconstruct the static background well when dynamic objects are in close contact with the background and the dynamic objects move slowly. Overall, the current reconstruction quality and effect still need to be further improved.

In this paper, we propose a method for high-precision and robust tracking in a dynamic environment and a method for reconstructing scene models without dynamic objects. An overview of our method is shown in Figure 1. Our method is divided into two stages. The first stage is based on the ORB-SLAM2 [4] system, and the workflow of this stage is indicated by the blue arrow. First, points, lines, and plane features are extracted from each frame; Mask-RCNN [5] and the connected component analysis method are used to accurately detect dynamic objects; the features in the dynamic area are dropped out; and the points, lines, and planes in the static area are used to construct an optimization equation to improve the tracking accuracy. When tracking and sparse reconstruction end, multiview projection methods are used to further optimize the mask. The workflow of the second stage is indicated by the green arrow, and the pose and mask of each frame in the first stage are used as the initial value of dense tracking. Then, the static background depth map in the current pose can be obtained subtly and precisely by using the ray-casting method. The pending area is obtained by comparing the depth map captured by the red green blue-depth (RGB-D) sensor and the depth map captured by the ray-casting method. Some region-handling methods are designed to divide static regions and dynamic regions in the pending area, and static regions are fused into the background by using the truncated signed distance function (TSDF) method. Finally, a clean background model is obtained. Our contributions are summarized as follows:

- A dynamic feature-removal method is proposed, which is based on a dynamic object mask and depth and projection error check. Dynamic object masks are obtained by Mask-RCNN and further optimized by using the connected component analysis method and a reference frame-based optimization method;
- We build an optimization model based on points, lines, and plane features to obtain higher trajectory accuracy and more robust tracking;
- We propose a static background reconstruction method, which uses the reconstructed information and ray-casting method to determine the pending regions. Some region-handling methods are designed to extract static regions from the pending regions, and static regions are used to reconstruct the static background model.

The remainder of the paper is organized as follows: Section 2 provides a brief overview of related work; Section 3 presents the proposed method in detail; Section 4 shows the experimental results, including comparisons by trajectory accuracy, tracking stability, and dense background reconstruction effect; Section 5 summarizes the whole work and the future outlook.

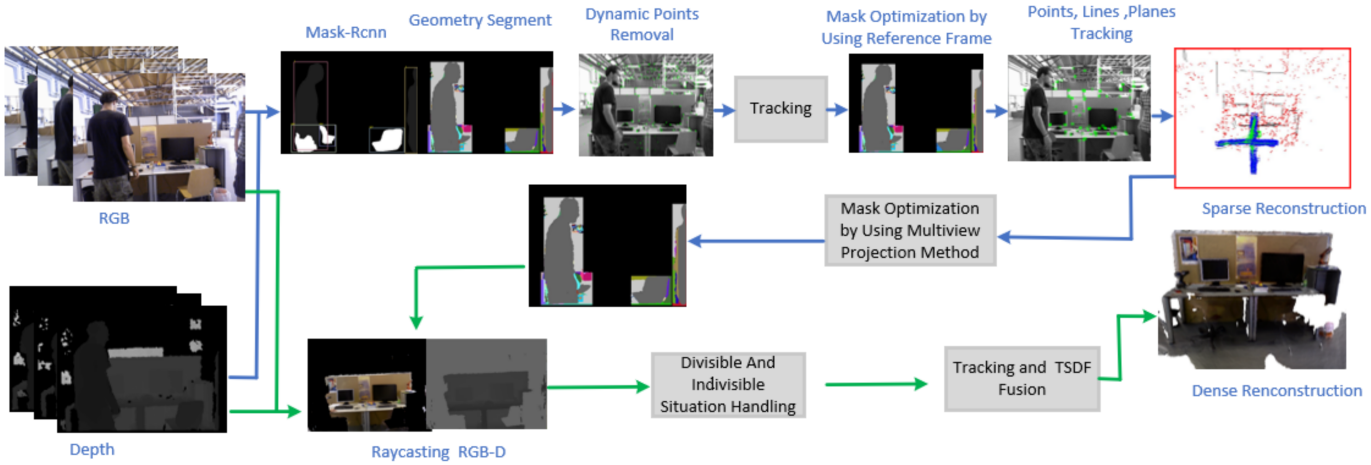


Figure 1. System overview.

2. Related Works

Most SLAM systems are based on static environment assumptions [3]. In the dynamic environment, these SLAM systems easily interfere with dynamic objects. The motion consistency check method is used to distinguish between dynamic points and static points. In Dai's method [9], the dynamic points were distinguished based on the principle that the movement of the feature points in the static background and the movement of the feature points in the dynamic area were inconsistent, thus causing the length of the connecting edge of the static point and the dynamic point to change. Sun [3] used sequence depth map differences and quantized depth images to classify dynamic regions and static regions, but the dynamic object area was not obtained when the moving object was stationary. The optical flow method can model points with different motion speeds. Wang [10] divided the points with different optical-flow motion-description vectors into several groups and used the random sample method to evaluate the current frame pose; the feature points in the group with the most numerous inner points were selected as static points, and the other points were selected as dynamic points. Liu [11] used dense optical flow to predict the semantic label of a region and reduced the influence of the dynamic objects according to the velocity of each landmark. Cheng [12] used essential matrix evaluation and optical flow vectors to determine whether a feature point was a dynamic point. Brasch [13] adopted a probabilistic model to reduce the influence of dynamic feature points; this method was affected when the dynamic object moved slowly. Wang [14] and DMS-SLAM [15] were based on the assumption that static feature points outnumber dynamic feature points; when dynamic objects occupied a larger region in the image, this method would degenerate. Given the development of deep learning, nowadays deep learning methods can be used to detect and segment dynamic objects accurately. Dyna-SLAM [16] used Mask-RCNN to obtain dynamic object masks, and simultaneously, dynamic points were identified by using multiview methods; then, these dynamic points were used to extract the masks of the dynamic object by the regio- growing method, and the feature points in nondynamic regions were used for tracking and sparse reconstruction. PLD-SLAM [17] combined deep learning segmentation and k-means methods [18] to obtain dynamic points in dynamic regions, and dynamic points were further filtered out by using depth and epipolar geometry constraints. The points and lines in the static region were used to improve the tracking accuracy. SGC-VSLAM [19] used deep learning object detection and geometric constraints to segment dynamic objects, and the optical flow method was used to optimize the dynamic object mask, while the feature points in the nondynamic area were used for tracking and sparse reconstruction. Han [20] used semantic segments and optical flow to filter out dynamic points. SOF-SLAM [21], SDF-SLAM [22], DS-SLAM [23], DM-SLAM [24], and OFM-SLAM [25] used deep learning segmentation combined with optical flow and other geometric constraints methods, such as epipolar constraints, to filter out dynamic points.

Dynamic-SLAM [26] used the bounding box compensation algorithm to compensate for the area removed by deep learning segmentation, and the scene information was fully used to improve the tracking robustness. Liu [27] used static probability and the static observation number as the weight of the feature point, and the improved random sample consensus method was used to filter out the dynamic point. Xie [28] used the depth learning method to segment dynamic objects, and the dynamic region inpainting method and optical flow method were used to eliminate the interference of dynamic feature points.

Currently, most scholars focus on dynamic scene tracking and sparse map construction, while relatively few scholars focus on clean-background dense reconstruction. Due to the complex characteristics of dynamic scenes, building a clean, high-quality, dense background of a dynamic scene is challenging. Fan and Zhang [8] used instance segments to obtain the mask of a dynamic object; depth value analysis and mask expansion methods were used to divide the dynamic area and the static area, and the points in the static area were used to construct the point cloud map. RS-SLAM [29] used octrees to construct scenes, and the labels and confidence of a voxel were used to determine whether a voxel was dynamic. DDL-SLAM [30] and Zhang [31] used object-detection methods to detect dynamic objects, where multiview constraints were used to further optimize the dynamic area mask, and static areas were used for octree-based reconstruction. Some algorithms track and reconstruct each instance individually. MaskFusion [32] and MID-FUSION [33] used Mask-RCNN and geometric methods to obtain the mask of each object, where the object was tracked and reconstructed separately, then, finally, the semantic map was obtained. StaticFusion [34] divided the depth map into different clusters; each weighted cluster was used to construct an error optimization function according to the photometric and geometric consistency constraint. This method could simultaneously estimate camera motion and segment static objects in the current frame; finally, the temporally consistent data was used to construct a dense map. Refusion [35], as a voxel-based real-time dynamic-scene-reconstruction method, used the distance between the point and the fusion surface to determine the dynamic pixel point. Although this method had better tracking and reconstruction capabilities, dynamic objects would still be reconstructed when the dynamic object was static or moved at low speeds in the scenes.

3. Proposed Methods

In this paper, we propose a two-stage algorithm for robust and high-precision tracking and clean background reconstruction. The first stage is the sparse reconstruction stage based on ORB-SLAM2, and the second stage is the dense background reconstruction stage based on TSDF fusion. The pipeline of the first stage is shown in Figure 2, where the position of the proposed method in the ORB-SLAM2 framework is indicated by the green boxes. We use Mask-RCNN and connected component analysis methods and a reference frame-based method to obtain the dynamic object mask. In tracking, the points, lines, and planes in the nondynamic area are used to construct an optimization model to improve the tracking accuracy and stability, and a multiview projection method is used to further optimize the mask after the tracking is completed. The first stage can provide an initial value for the second stage. The method used in the first stage will be introduced in detail.

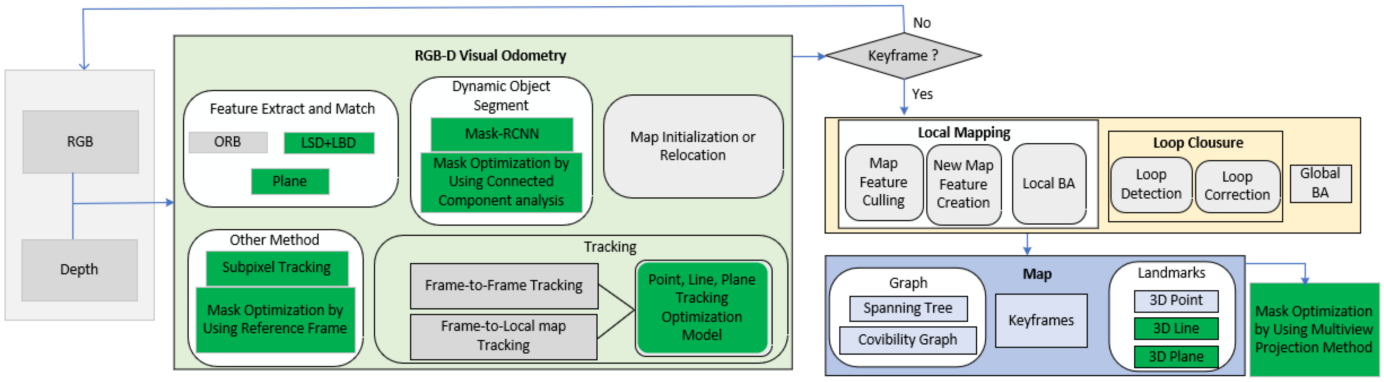
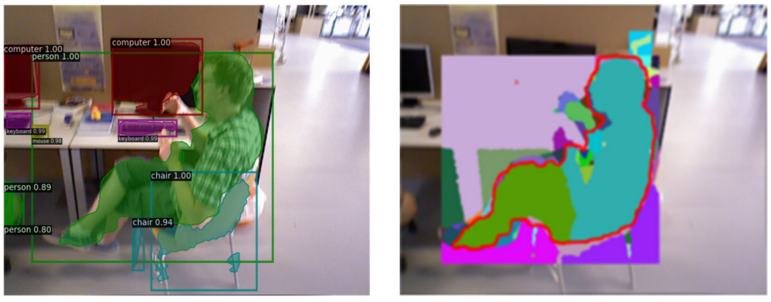


Figure 2. Overview of the first stage.

3.1. Subsection Dynamic Object Mask Generation

3.1.1. Dynamic Object Mask Extraction Using Mask-RCNN

To exclude the interference of dynamic objects, we first need to know the region of the dynamic object. In a scene where dynamic objects move slowly or are in close contact with the background, the depth-based connected component analysis method cannot accurately obtain the dynamic object region. Mask-RCNN is an image-based instance-level segmentation algorithm that can provide prior information of dynamic objects in the scene. Compared with semantic segmentation, a bounding box of a dynamic object can be provided by Mask-RCNN; this bounding box contains most of the object's information, so this bounding box can provide a good range limit for depth-based connected component methods. Mask-RCNN is used to segment the dynamic objects in this paper. As shown in Figure 3a, people and chairs are segmented as dynamic objects; the segmentation results of deep learning suffer from over-segmentation or under-segmentation. In the mask, some static background areas are divided into dynamic foreground areas, and some dynamic foreground objects are divided into static backgrounds. When the dynamic area is divided into the static background, the feature points in this dynamic region will be regarded as static points and participate in tracking, thus affecting the tracking accuracy. Therefore, it is necessary to process the dynamic object mask extracted by Mask-RCNN.



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

Figure 3. Dynamic region segment. (**a**) The result of Mask-RCNN. (**b**) The mask refined by the connected component analysis method.

3.1.2. Mask Optimization by Connected Component Analysis Method

Mask-RCNN only uses image information for instance segmentation; in RGB-D data, the depth information can be used to further optimize the mask segmented by Mask-RCNN. In the depth map, when the dynamic area and its adjacent static background have a large depth difference, it is easy to separate dynamic objects from the scene; in some cases, when people put their hands on the table, the depth difference between the dynamic area and its adjacent static background area is relatively small, and it is difficult to accurately remove dynamic objects. Compared with the smooth area in an object, the normal and variance

at the junction of the dynamic area and its adjacent static background will differ more. Therefore, the depth difference $\varnothing_d$, normal difference $\varnothing_e$, and variance difference $\varnothing_\sigma$ are used to express the difference of a pixel, as shown in Formula (1).

$$\varnothing = \varnothing_d + \varnothing_\sigma + \gamma_1 \cdot \varnothing_e \tag{1}$$

where γ_1 is the weight of $\varnothing_e$. $\varnothing_d$, $\varnothing_e$ and $\varnothing_\sigma$ are expressed as follows:

$$\varnothing_d = \max_{i \in N} |(\boldsymbol{v_i} - \boldsymbol{v}) \cdot \boldsymbol{n}| \tag{2}$$

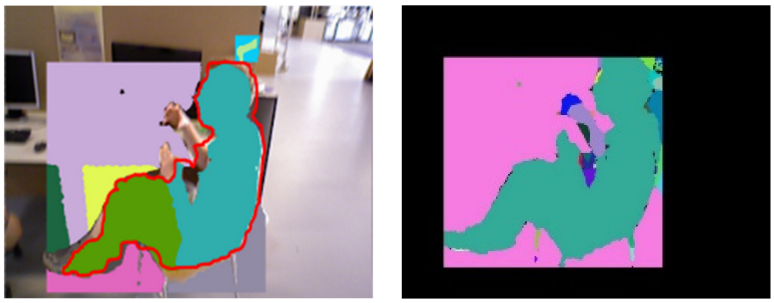
$$\varnothing_e = \max_{i \in N} \begin{cases} 0 & if((\boldsymbol{v_i} - \boldsymbol{v}) \cdot \boldsymbol{n}) < 0 \\ 1 - (\boldsymbol{n_i} \cdot \boldsymbol{n}) & else \end{cases} \tag{3}$$

$$\varnothing_\sigma = \sqrt{\sum_{i=1}^{N} \frac{(\boldsymbol{v_i} - \boldsymbol{v})^2}{N}} \tag{4}$$

where $\boldsymbol{v}$ represents the point on the depth map, N represents the neighborhood point index sets of point $\boldsymbol{v}$, and $\boldsymbol{v_i}$ represents the neighborhood point of $\boldsymbol{v}$. γ_1 is set to 5 in our work.

Formula (1) is used to calculate the weight value of every pixel. As shown in Figure 3b, the region in the red contour is the dynamic object segmented by Mask-RCNN, and the part in the bounding box but outside the red contour is the nondynamic recognition region. The connected component method based on the $\varnothing$ value is applied in these two regions to obtain every area, and different areas are marked with different colors.

In Figure 4, in the region segmented by Mask-RCNN, dynamic objects occupy a larger proportion, and in the region that is in the bounding box but outside the red contour, the static background accounts for a larger proportion. We regard the larger areas in these two regions as static objects and dynamic objects, as shown in Figure 4a. The main part of the dynamic object inside the red contour is shown, and the main part of the static region in the bounding box but outside the red contour is shown.



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

Figure 4. Region analysis. (**a**) Static and dynamic main area determination. (**b**) The result of region merging.

When there are more points with consistency $\varnothing$ on the boundary of two regions, the two regions may be the same region. Assuming that the two regions are A and B, A is the small area; the number of boundary points with consistent depth in region A is NC, amd the number of remaining boundary points in region A is ND.

$$\mathrm{NC} > \mathrm{ND} \tag{5}$$

When the conditions in Formula (5) are satisfied, the two regions are considered to be the same connected components, and A will be merged into B. This condition is used as the boundary condition.

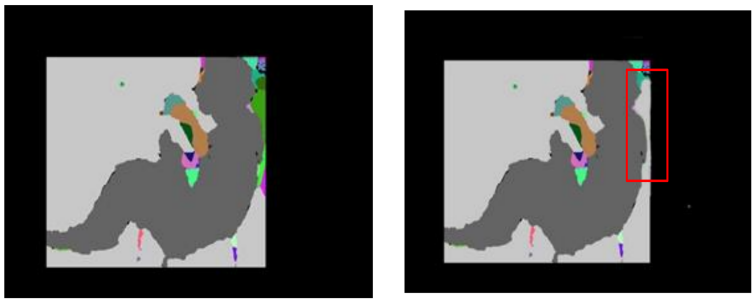
After the main part of the static region and the main portion of the dynamic objects are determined, the remaining small areas should be merged into the main area to optimize

the mask. In the bounding box, we use boundary conditions to determine whether a region can be merged into its adjacent regions. Some areas are merged into adjacent main regions according to the boundary conditions; in addition, we consider that the outside of the bounding box is a static region. If an area is adjacent to a static region outside the bounding box and meets the boundary conditions with this adjacent static region, the area is considered to be a static area. As shown in Figure 4b, after processing by our method, most of the areas within the bounding box are identified, and the remaining areas that cannot be merged are still regarded as undetermined areas.

3.1.3. Mask Optimization Using a Reference Frame

We extract feature points, lines, and planes of the current frame and use the mask to remove the feature in the dynamic region, and the feature points, lines, and planes in the static region are used for lightweight tracking. After lightweight tracking, the pose of the current frame can be obtained. Each point in the undetermined area on the current frame is projected to the reference frame depth map according to the pose of the current frame to find the corresponding point. If the point has a consistent depth with the corresponding point and the corresponding point is in the static region, this point is viewed as a static point. If the proportion of static points in an area exceeds the proportion of dynamic points, the connecting region is considered to be a static area; otherwise, the connecting region is still an area that cannot be identified.

Figure 5a is another color representation of Figure 4b. Figure 5b shows the result of the mask refined by using the reference frame. After the reference frame is used to determine the undetermined area, part of the undetermined area is divided into the background, which is conducive to tracking and reconstruction.



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

Figure 5. Mask optimization. (**a**) Mask merging. (**b**) Mask optimization using a reference frame.

3.2. Dynamic Features' Removal

3.2.1. Dynamic Features' Removal by a Mask

Most dynamic objects fall within the bounding box extracted by Mask-RCNN. Therefore, only a few feature points are dynamic points outside the bounding box. After the dynamic region segmented by Mask-RCNN is optimized by the connected component analysis method, we use the points, lines, and planes outside the dynamic region to perform lightweight tracking. The pose of the current frame is obtained, and based on the pose, the reference frame optimization method is used to optimize the mask, and the mask is used to further filter out the feature points, lines, and planes that fall within the mask.

3.2.2. Depth Constraint and Projection Error Check

We use points, lines, and planes in the static area for lightweight tracking. Since some dynamic regions failed to be divided into the dynamic mask, these dynamic regions will participate in the tracking. The depth and position of these feature points will change with the movement of the dynamic object, so while bundle adjustment is used to optimize the

pose of the current frame, depth error d_{proj} and projection error d_{depth} are used to further filter out the dynamic points.

$$d_{proj} = \|\boldsymbol{p} - \Omega(\boldsymbol{R}\cdot\boldsymbol{P^i} + \mathbf{t})\| \tag{6}$$

$$d_{depth} = \|D_p - \mathrm{z}(\boldsymbol{R}\cdot\boldsymbol{P^i} + \mathbf{t})\| \tag{7}$$

As shown in Figure 6, $o-xyz$ and $o_1-x_1y_1z_1$ represent the reference frame coordinate system and the current frame coordinate system, respectively; $\boldsymbol{R}$ and $\mathbf{t}$ represent the rotation matrix and translation vector of the current frame; Ω represents the perspective projection transformation; z( ) represents the z value of a point; p and p' represent the matched feature points; p^i represents the projection position of point P in the current frame; D_p represents the depth value of point p_0 on the depth map; d_{proj} represents the distance between p_0 and p; and d_{depth} represents the difference between D_p and the depth value of P in the current frame coordinate system.

Assuming that τ_{proj} and τ_d are the established projection error threshold and depth error threshold, respectively, if $d_{proj} > \tau_{proj}$ or $d_{depth} > \tau_d$, the point is considered to be a dynamic point, and the point will be filtered out. τ_{proj} is the same as the feature point projection error threshold in ORB-SLAM2.

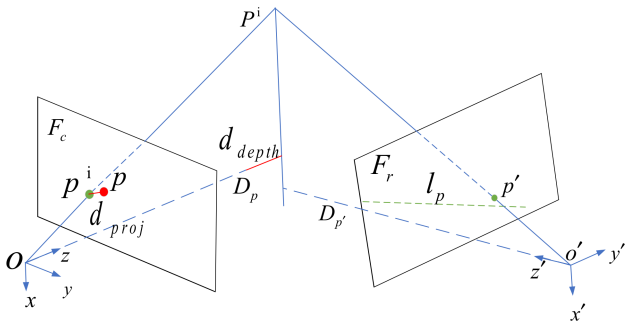


Figure 6. Depth error and projection error.

3.3. Subpixel Tracking

ORB-SLAM2 uses pixel-level corner points for tracking; in some undulating areas, such as mouse lines and keyboard corners, differences in viewing angles may cause position deviations of matching points during tracking. In this paper, the subpixel optimization algorithm is used to further optimize the position of the matching point on the current frame. Considering the matching speed, the quadratic polynomial equation fitting method is used to obtain the subpixel feature match point, which is expressed as follows:

$$z = a + bx + cy + dx^2 + ey^2 + fxy,\ x, y \in [-1,\ 1] \tag{8}$$

where x and y are the coordinate positions of the pixels in the neighborhood 3×3 window centered on the current point. Therefore, the value range of x, y is $[-1, 1]$, where z represents the zero-mean normalized cross-correlation (ZNCC) similarity value of the point (x, y), (a, b, c, d, e, f) are the coefficients, which can be obtained by the linear least-squares method, and the extreme point position is obtained through the first derivative of the equation. During tracking, after the feature points of the current frame and the reference frame are matched, Formula (8) is used to optimize the feature points' position in the current frame. If the refined point is within the neighborhood window, the optimization is considered to be successful.

3.4. Optimization Model of Points, Lines, and Planes

When the dynamic object occupies a large foreground area, the dynamic mask is used to remove the dynamic area. The remaining static points are few and unevenly distributed,

which may lead to unstable tracking and decreased accuracy. Therefore, we make full use of line and plane information in indoor scenes to improve the robustness and accuracy of tracking.

3.4.1. Line Structure Constraints

The line segment detector (LSD) [36] algorithm is used to extract and describe the line on the RGB image. In tracking, the line is matched by the line description. The 2D line on the image and the corresponding three-dimensional line in space can be used to construct constraint equations to maintain the line structural consistency and improve he tracking accuracy. As shown in Figure 7, assuming that the endpoints of the line in the space are $\boldsymbol{P_s}$ and $\boldsymbol{P_e}$, O_1, O_2, and O_3 are the camera centers of the three frames, and $\boldsymbol{l_1}$, $\boldsymbol{l_2}$, and $\boldsymbol{l_3}$ are the projections of $\boldsymbol{P_s}$ and $\boldsymbol{P_e}$ on the three images. The endpoints of the line $\boldsymbol{l}$ extracted by LSD on the image are p_s and p_e, and the projection points of $\boldsymbol{P_s}$ and $\boldsymbol{P_e}$ on the image are $\boldsymbol{p'_s}$ and $\boldsymbol{p'_e}$; $\boldsymbol{p'_s} = (p'_{sx}, p'_{sy}, p'_{sz})$, and $\boldsymbol{p'_e} = (p'_{ex}, p'_{ey}, p'_{ez})$. The 3D line in space can be expressed as $(\boldsymbol{P_s}, \boldsymbol{P_e} - \boldsymbol{P_s}/\|\boldsymbol{P_e} - \boldsymbol{P_s}\|)$), and the line l can be calculated by the cross-product, which is expressed as $\boldsymbol{L} = p_s \times p_e$. Suppose that $\boldsymbol{L} = (a, b, c)$. The, then projection distance d can be expressed as in Formula (9):

$$d_L(l, L) = \frac{\mathrm{a} \cdot p'_{sx} + b \cdot p'_{sy} + c \cdot p'_{sz}}{\sqrt{a \cdot a + b \cdot b}} + \frac{\mathrm{a} \cdot p'_{ex} + b \cdot p'_{ey} + c \cdot p'_{ez}}{\sqrt{a \cdot a + b \cdot b}} \tag{9}$$

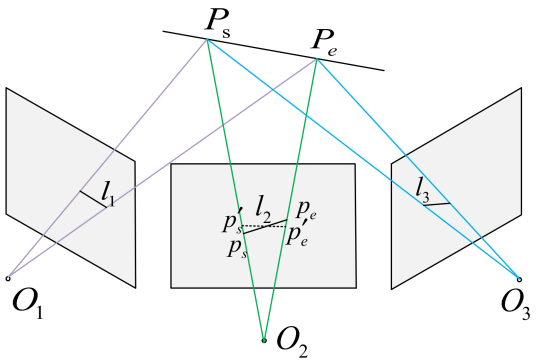


Figure 7. Line constraint.

3.4.2. Plane Structure Constraints

There is also some plane structure information in the indoor scene, such as that concerning the floor, wall, and desktop. These planes can be used as structural constraints to assist in tracking; the cascaded plane extraction method [37] is used to extract planes from each depth map. Considering that a plane with only a few support points may be noise, the planes with extracted plane support pixels of less than 5000 points are deleted. During tracking, the initial pose (rotation matrix $\mathbf{R}$ and translation vector $\boldsymbol{t}$) of the current frame can be obtained through the uniform motion model. Assume that $\boldsymbol{n^c_{predict}}$ and $d^c_{predict}$ represent the parameters of the plane normal and the distance from the origin to the plane in the current frame coordinate system, $\boldsymbol{n}^W$ and d^W represent the parameters of a plane in the map, and p is a point in the depth map of the current frame. Formulas (10) and (11) are used to determine whether two planes are similar.

$$\|(\mathbf{R}^{-1} \cdot \boldsymbol{n^c_{predict}})^T \cdot \boldsymbol{n}^W\| > \tau_\theta \tag{10}$$

$$(\boldsymbol{n}^W)^T \cdot \left(\mathbf{R}^{-1} \cdot \boldsymbol{p} - \mathbf{R}^{-1} \cdot \boldsymbol{t}\right) + d^W < \tau_{dis} \tag{11}$$

The angle threshold τ_θ and point-to-face distance threshold τ_{dis} are, respectively, set as 0.9 and 0.02 m in our experiment. When the plane in the current frame and the plane

in the map meet the normal and distance constraints in Formulas (10) and (11), the two planes are considered to be the matched plane.

Assuming that the plane is represented by (φ, θ, d), $\boldsymbol{\pi}^c$ represents the current frame plane, $T(\pi^c)$ represents the transformation matrix that can transform the plane from the current frame coordinate system to the world coordinate system, and $\boldsymbol{\pi}^w$ represents the map plane in the world coordinate system. The specific formula is as follows:

$$q(\pi) = \left(\varphi = \arctan\left(\frac{n_y}{n_x}\right), \theta = \arcsin(\boldsymbol{n}_z), d\right) \tag{12}$$

$$e^{\pi}(\pi_c, \pi_W) = q(T(\boldsymbol{\pi}^c)) - q(\boldsymbol{\pi}^w) \tag{13}$$

The difference $e^{\pi}(\boldsymbol{\pi}_c, \boldsymbol{\pi}_W)$ between the current frame plane and the corresponding 3D plane in the map are used as constraints in the pose optimization.

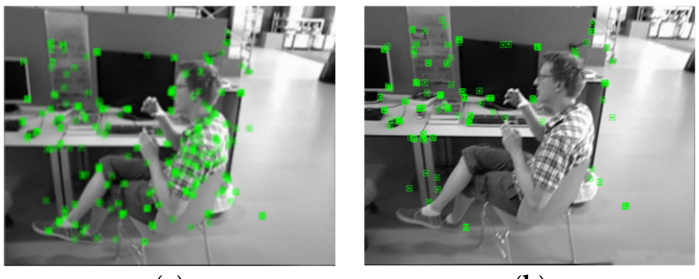
3.4.3. Sparse Tracking and Reconstruction in Dynamic Scenes

As Figure 8c shows, our method removes the dynamic feature points well, thereby providing a good basis for tracking.

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

Figure 8. Results of feature point extraction. (**a**) The original RGB image. (**b**) The feature point extraction results in the ORB-SLAM2 system. (**c**) The feature point extraction results of our method.

Figure 9a shows that in the tracking result of the ORB-SLAM2 system, feature points in the dynamic regions are also involved in the tracking, and these dynamic points will cause some interference with the tracking. As shown in Figure 9b, dynamic points no longer participate in tracking when using our method. Compared to the ORB-SLAM2 system, our method can eliminate the interference of dynamic points.



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

Figure 9. Tracking results. (**a**) The tracking result of ORB-SLAM2. (**b**) The tracking result of our method.

During tracking and sparse reconstruction, Mask-RCNN and the connected component analysis method and the referenced frame-based method are used to extract the dynamic area mask; the ORB feature extraction algorithm, LSD, and plane extract algorithm are used to extract the feature points, lines, and planes; the features in the static region are used for lightweight tracking, and the mask is refined by using the reference frame. Points, lines, and planes that fall in the dynamic region are removed; then, the points, lines, and

planes in the static region are used to construct error constraint equations for tracking, local graph optimization, and global optimization. The global optimization equation is shown as Formula (14):

$$\begin{aligned}\{\boldsymbol{P^u}, \boldsymbol{L^v}, \boldsymbol{\pi^h}, (\boldsymbol{R_s}, \boldsymbol{t_s}) | \quad & u \in \mathcal{K}_p, \boldsymbol{v} \in \mathcal{K}_l, h \in \mathcal{K}_\pi, s \in N_k\} \\ = \quad & argmin(\sum_{b=1}^{N_k} \sum_{i=1}^{N_p} \rho(E_p(b,i)) + \sum_{b=1}^{N_k} \sum_{j=1}^{N_l} \gamma_{2\cdot} \rho(E_{line}(b,j)) \\ + \quad & \sum_{b=1}^{N_k} \sum_{k=1}^{N_k} \gamma_{3\cdot} \rho(E_{plane}(b,k)))\end{aligned} \tag{14}$$

where ρ represents the robust Huber kernel function, Ω represents the perspective projection transformation, $\boldsymbol{P^u}$ represents the map points, $(\boldsymbol{R}_s, \boldsymbol{t}_s)$ represents the frame poses to be optimized, L^v represents the map lines to be optimized, $\boldsymbol{\pi}^h$ represents the planes to be optimized, $\mathcal{K}_p$ represents the index set of the map points, $\mathcal{K}_l$ represents the index set of the total map line, $\mathcal{K}_\pi$ represents the index set of the total map planes, N_k represents the number of keyframes, N_p represents the number of map points, γ_2 and γ_3 represent the weights, N_l represents the number of map lines, and N_π represents the number of map planes. $\gamma_{2\cdot}$ and $\gamma_{3\cdot}$ are defined by using each iteration's residuals of error terms related to line and plane features.

The feature point projection error $E_p(b,i)$, line projection error $E_{line}(b,j)$, and plane error $E_{plane}(b,k)$ are expressed as follows:

$$E_p(b,i) = \|\boldsymbol{p} - \Omega\left(\boldsymbol{R}_b \boldsymbol{P^i} + \boldsymbol{t}_b\right)\| \tag{15}$$

$$E_{line}(b,j) = d_L(\boldsymbol{l}_b^j, \boldsymbol{L}^j) \tag{16}$$

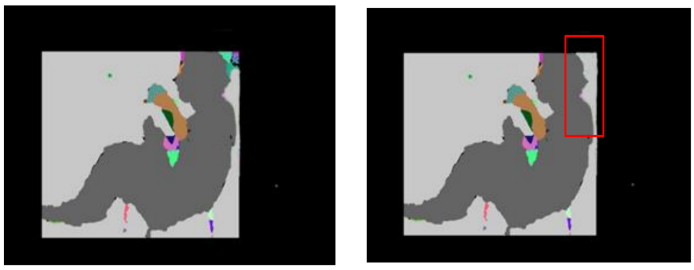
$$E_{plane}(b,k) = e^\pi\left(\boldsymbol{\pi}_{bc}^k, \boldsymbol{\pi}_W^k\right) \tag{17}$$

3.5. Mask Optimization by Multiview Projection

When the sparse reconstruction finishes, we use the multiview projection method to optimize the mask of the dynamic object. First, both the distance and the rotation angles between the current frame Fc and another keyframe Fn are used to build $dist(Fc, Fn)$, which is similar to the method of selecting adjacent frames in Dyna-SLAM [16]. The five keyframes closest to the current frame are selected for mask optimization.

After the neighboring five frames of each frame are selected, similar to the method used in the "Mask Optimization by Using a Reference Frame" section, each undetermined region on the current frame is projected onto the five reference frames, and the proportion of static points in the connected area is counted in every reference frame. If the proportion of static points in any of the five statistics exceeds the proportion of dynamic points, the area is considered to be a static area.

Figure 10 shows that after multiview projection, some undetermined areas are accurately determined to be background areas, and the mask quality is further improved. The remaining area is relatively small, and it is found in the experiment that these areas have little effect on tracking. Therefore, our method can obtain a higher-quality mask, which can provide a good initial value for dense reconstruction.



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

Figure 10. Mask refined by the multiview method. (**a**) Mask before the multiview optimization. (**b**) Mask after the multiview optimization.

3.6. Detailed Overview of Dense Background Reconstruction in Dynamic Scenes

This paper proposes a method for the dense reconstruction of static backgrounds in dynamic scenes. As shown in Figure 11, the pose and mask of each frame obtained from the first stage are used to perform initial dense tracking, and the obtained pose is used to generate the static background depth map in the current pose by using the ray-casting method. The pending area is obtained by comparing the depth map generated by the ray-casting method and the depth map captured by the RGB-D sensor. We designed some region-handling methods to classify the dynamic and static regions in the pending area. The static regions are used to refine the dense tracking and are fused into the dense background map by using the voxel-based TSDF method. Finally, a clean and dense background model is obtained.

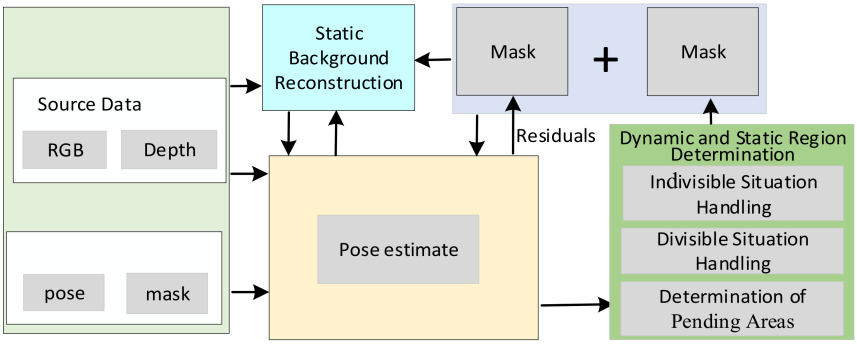


Figure 11. Dense reconstruction process.

3.7. Determination of Pending Areas

In the reconstruction of the first frame, the dynamic object mask extracted in the first stage is used to remove the dynamic object, and the non-dynamic area in the first frame is reconstructed. In the following frames, we make full use of the reconstructed background information to assist in determining the pending regions in the current frame. In the initial tracking stage, the mask and pose of the current frame in the sparse reconstruction are used as the initial value. Since the mask removes most of the dynamic objects, an accurate camera pose can be obtained in the initial dense tracking stage. By using the ray-casting method with the current camera pose, a depth map of the static background map can be obtained in the same pose of the current frame.

Suppose that $\boldsymbol{D}_{\mathrm{C}}$ represents the current frame depth map and $\boldsymbol{D}_{\mathrm{R}}$ represents the depth map generated by using the ray-casting method. Where there is a difference between the depth values of $\boldsymbol{D}_{\mathrm{R}}$ and $\boldsymbol{D}_{\mathrm{C}}$, the region may be a dynamic region or a newly static region. The depth difference area is calculated by Formula (18).

$$d\_sub = |\boldsymbol{D}_{\mathrm{R}} - \boldsymbol{D}_{\mathrm{C}}| \tag{18}$$

where d_sub represents the difference between $\boldsymbol{D}_{\mathrm{R}}$ and $\boldsymbol{D}_{\mathrm{C}}$. Suppose $\boldsymbol{M}_{\mathrm{C}}$ represents the mask of the depth map of the current frame, which indicates the state of each pixel. The point (u, v) in $\boldsymbol{D}_{\mathrm{C}}$ with an invalid depth value or depth value greater than the value of the sensor measurement range is evaluated as an invalid point, and $\boldsymbol{M}_{\mathrm{C}}(\mathrm{u},\ \mathrm{v})$ is set to an invalid point. The point (u, v) with a depth that is valid in $\boldsymbol{D}_{\mathrm{C}}$ but invalid in $\boldsymbol{D}_{\mathrm{R}}$ is evaluated as a pending area, and $\boldsymbol{M}_{\mathrm{C}}(\mathrm{u},\ \mathrm{v})$ is set as an undetermined point. When the depth values of the point (u, v) in $\boldsymbol{D}_{\mathrm{R}}$ and $\boldsymbol{D}_{\mathrm{C}}$ are all valid, the point can be divided by the following formula:

$$\boldsymbol{M}_{\mathrm{C}}(\mathrm{u},\mathrm{v}) = \begin{cases} 1 & d\_sub(u,\boldsymbol{v}) > \tau_d \\ 0 & \text{else} \end{cases} \tag{19}$$

If $\boldsymbol{M}_C(u,\boldsymbol{v})$ is marked as 1, then $\boldsymbol{M}_{\mathrm{C}}(\mathrm{u},\mathrm{v})$ is a dynamic point; otherwise, it is a static point. We divide the pending area into $\boldsymbol{M}_C^D$. Figure 12 show the results of the ray-casting. Figure 13 shows invalid data and region classification. From Figures 12b and 13a, we can see that $\boldsymbol{M}_C^D$ contains the following types of areas:

$$Patch(NBR) + Patch(NDR) + Patch(DR) + Patch(IR) + Patch(NR) = \boldsymbol{M}_C^D \tag{20}$$

$Patch(NBR)$ represents the newly added background areas, $Patch(NDR)$ represents the newly added dynamic areas, $Patch(IR)$ represents an invalid depth value area, $Patch(NR)$ represents noise blocks, and $Patch(DR)$ represents the dynamic area that has been reconstructed.

$\boldsymbol{M}_C^S$ is the region of static points in $\boldsymbol{M}_C$. From Figures 12b and 13a, we can see that it contains the following two types.

$$Patch(SDR) + Patch(SBK) = \boldsymbol{M}_C^S \tag{21}$$

$Patch(SDR)$ represents the dynamic area in the static background. These areas are generated because when a dynamic object appears in the scene for the first time, the bounding box extracted by deep learning is inaccurate, thus causing a very small area of the dynamic object to leak into the static background. $Patch(SBK)$ represents the static background that has been reconstructed.

In $\boldsymbol{M}_C$, $Patch(IR)$ represents the region of invalid points, which does not need to be processed; the regions that generally occupy a large area in $Patch(SBK)$ and $Patch(DR)$ are viewed as the static and dynamic main body, respectively, which are relatively easy to identify. The main focus of our method is how to deal with $Patch(NBR)$, $Patch(NDR)$, $Patch(NR)$, and $Patch(SDR)$; these four areas are caused by the movement of dynamic objects and the camera. In continuous frames, these areas are generally relatively small, as shown in Figure 13b. This article will introduce how to address these four types of areas.

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

Figure 12. Result of ray casting. (**a**) The depth map of the current frame. (**b**) The reconstructed background. (**c**) The depth map generated by the ray-casting method in the same pose as the current frame.

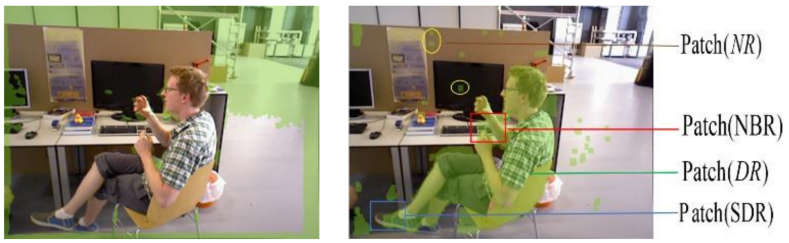


Figure 13. Region classification. (**a**) Invalid depth area and area beyond the effective measurement distance in M_C^D. (**b**) $Patch(NR)$, $Patch(NBR)$, and $Patch(DR)$.

3.8. Divisible and Indivisible Situation Handling

In $\boldsymbol{M}_C^D$, the connected component analysis method is used to obtain all connected regions, and the region with the largest area is considered the dynamic main region *Patch(DR)*, while the other connected regions include $Patch(NBR)$, $Patch(NDR)$, $Patch(NR)$, and $Patch(SDR)$. The points in $\boldsymbol{M}_C^D$ are used to perform region growth in $\boldsymbol{M}_C^S$ to obtain *Patch (SDR)* in the reconstructed region. Patches that contain only dynamic regions or only static regions are considered to be divisible regions, while patches that contain both dynamic and static regions are considered to be indivisible patches. Since the change between the two consecutive frames is very small, the new regions generated by the change of the angle of view should be small, so we set an area threshold τ_{area} where range is 6000–10,000, and only the patches with an area smaller than τ_{area} are processed. When the area of a region is less than τ_{area}, we simply view it as a divisible area.

3.8.1. Small Noise Block Removal

$Patch(NR)$ comprises discrete small blocks generated due to the unstable depth value of the object edge or areas with reflective surfaces in the static background. Since the area of each noise block is small, as shown in Formula (22), the morphological erosion method is used to remove small noise areas, and the morphological dilate method is used to restore other areas. Supposing that *Mask* contains $\boldsymbol{M}_C^D$ and $\boldsymbol{M}_C^S$, the formula is as follows:

$$Mask \leftarrow dilate(erode(Mask)) \tag{22}$$

3.8.2. Dynamic Block $Patch(SDR)$ Removal

When a new dynamic object enters the scene for the first time, $Patch(SDR)$ will be reconstructed because the dynamic object cannot be completely segmented by deep learning and connected component analysis. If $Patch(SDR)$ is not processed, in the next few frames, the newly added area adjacent to $Patch(SDR)$ will be determined to be a static region, and these newly added areas will be reconstructed in the background, thus leading to a map error. *Patch (SDR)* is usually adjacent to an area in $\boldsymbol{M}_C^D$ and has a consistent depth with this adjacent region. According to this characteristic, *Patch (SDR)* is processed. Suppose that patch$(\boldsymbol{i}')$ represents a region in $\boldsymbol{M}_C^D$, patch$(\boldsymbol{i})$ represents an area in $\boldsymbol{M}_C^S$ and is adjacent to patch(i'), N_{SC} represents the number of boundary points patch(i) at a consistent depth with the points in $\boldsymbol{M}_C^D$, and N_B represents the other bounding pixels of patch$(\boldsymbol{i})$. $\boldsymbol{S}_{\text{patch}(\boldsymbol{i})}$ represents the number of pixels in the block patch$(\boldsymbol{i})$, and $\boldsymbol{S}_{\text{patch}(\boldsymbol{i}')}$ represents the number of pixels in the block patch$(\boldsymbol{i}')$.

$$\boldsymbol{S}_{\text{patch }(i)} < \boldsymbol{S}_{\text{patch }(i')} \tag{23}$$

$$N_{SC} > N_B \tag{24}$$

If the constraints in Formulas (23) and (24) are satisfied, then patch(i) is merged into the block patch(i'), and the dynamic block $Patch(SDR)$ removal method can effectively prevent the reconstructed dynamic area from expanding further.

3.8.3. Static Block $Patch(NBR)$ Extraction in M_C^D

On consecutive frames, the static area newly added in each frame is generally small. Assuming that $patch(j')$ is an area in $\boldsymbol{M}_C^D$, this region may be connected to the static background and other dynamic scenes. The number of points that are on the boundary of $patch(j')$ and have consistent depth with its adjacent points in the static background region is N_{SL}^D, the number of points that are on the boundary of $patch(j')$ and have consistent depth with its adjacent points in the dynamic region is N_{DL}^D, and the number of valid boundary points of $patch(j')$ is N_{VE}. Formulas (25) and (26) can be used to determine whether $patch(j')$ is a static area.

$$N_{SL}^D > N_{VE} \tag{25}$$

$$N_{SL}^D > N_{DL}^D \tag{26}$$

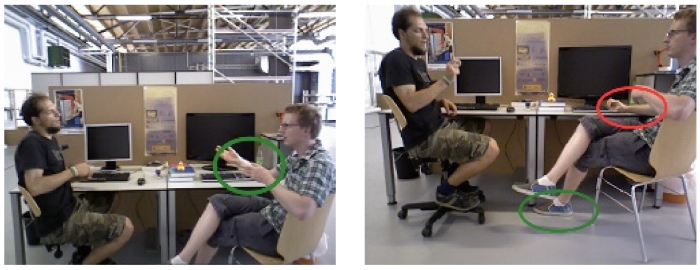
If area $patch(j')$ meets the conditions of Formulas (25) and (26), $patch(j')$ will be merged into the static area.

3.8.4. Floating Block $Patch(SDR)$ Processing in the Bounding Box

Part of the dynamic object may differ in depth from the dynamic object; for example, the depth of the arm differs from the body, as shown in Figure 14a. Therefore, the method of counting the number of points with consistent depth on the boundary is no longer applicable. Instead, we count boundary points with the dynamic label. These regions generally have a small area compared with the main body. Therefore, boundary constraints and area constraints are used to filter part of the patch. Assuming that the number of bounding points with an adjacent point with a dynamic label is N_{SL} and the number of bounding points with an adjacent point with a dynamic label is N_{DL}, the discriminant formula can be expressed as follows:

$$N_{DL} > N_{SL} \tag{27}$$

If the condition in Formula (27) is met, the small area is considered to be a dynamic area, and the small region is merged into its neighboring dynamic area.



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

Figure 14. Divisible dynamic block and depth indistinguishable situation. (**a**) The right hand is unconnected to the main body. (**b**) The arms and feet are closely connected to the static background.

3.8.5. Indivisible Situation Handling

When the dynamic area is closely connected with the background and nonconnected with the people in depth (such as concerning the hand and the desktop in Figure 14b, where there is no obvious difference in depth), it is more difficult to distinguish which part is static. Generally, this small area may be merged into the static area by using our method. However, after the hand moves away after a while, the fusion of static background data of continuous multiple frames will make the reconstructed part of the dynamic object weaken or even disappear.

When a new area is closely connected with dynamic objects and this dynamic part is static for a long time, such as concerning the feet and the ground in Figure 14b, although the ground area will become increasingly larger as the depth camera moves, it is difficult to

accurately distinguish between the feet and the static ground. In this paper, the structural information in the scene is used to address this indivisible situation; the planes are extracted from the depth map, and the plane area will become increasingly larger as the camera moves. When the plane area is large enough, the plane area is judged to be a static area, and the points in the plane are fused into the static background map.

3.9. Bounding Box Tracking and Noncontact Static Region Extraction

The algorithm used in this paper can reconstruct only the area connected to the static background. Therefore, when some areas enter the scene, the algorithm cannot reconstruct areas that are static and that are not in contact with the background that has been reconstructed. As the angle of view moves, most areas in the current frame have not been reconstructed, and dense tracking may fail. We use the bounding box tracking method to process this problem.

To obtain a good bounding box, the ID, mask, and bounding box obtained by deep learning segmentation are first merged with the projected area of the dynamic area of the previous frame. Assume that $PBBX_{(i)}$ represents the bounding box of the projected area of the i-th object in the previous frame, $BBX_{(j)}$ represents the bounding box of the j-th object in the current frame generated by using Mask-RCNN segmentation, $PID(j)$ represents the object ID of $PBBX_{(i)}$, and $ID_{(j)}$ represents the object ID of $BBX_{(j)}$. When the overlap of $PBBX_{(i)}$ and $BBX_{(j)}$ is large, the objects in the two bounding boxes are considered to be the same object, and the combined bounding box can be identified as $BBX^{'} = PBBX_{(i)} \Big| BBX_{(j)}$.

$$area(PBBX_{(i)}) > \gamma_4 * area(BBX_{(j)}) \tag{28}$$

$$area(BBX_{(i)}) > \gamma_4 * \text{area}(PBBX_{(j)}) \tag{29}$$

$$ID_{(i)} = PID(j) \tag{30}$$

where the value range of γ_4 is (0.5, 1), and if the $PBBX_{(i)}$ and $BBX_{(j)}$ satisfy the constraints in Formulas (28)–(30), these two boundary boxes can be merged. Suppose that $DBBX_{(i)}$ represents the area constituted by the connected component domain analysis based on depth in $\boldsymbol{M}_C^D$, $rectU(i,j) = DBBX_{(i)} \Big| BBX^{'}{}_{(j)}$ represents the union of two regions, $rectA(i,j) = DBBX_{(i)}$ & $BBX^{'}{}_{(j)}$ represents the intersection of the two regions, and $IOU_{occp(i,j)} = \text{area}(rectA(i,j))/\text{area}\left(BBX^{'}{}_{(j)}\right)$ represents the proportion of the intersection region in the area $BBX^{'}{}_{(j)}$. If the ratio of the area of $rectA(\boldsymbol{i},j)$ to the area of $DBBX_{(\boldsymbol{i})}$ and the area of $BBX^{'}{}_{(j)}$ is more than γ_4, or when IOU_occp(i,j) is very close to 1, the two bounding boxes can be combined If there is no intersection between $BBX^{'}{}_{(j)}$ and any bounding box in $DBBX$, the area in $BBX^{'}{}_{(j)}$ may be a new area, and its ID is assigned as -1.

The ID and region bounding box of multiple continuous frames are used to determine whether the area is static, as shown in Figure 15. If no ID is allocated in consecutive frames in the red bounding box, and it is not connected to the edge of other areas, when the camera moves left continuously, the tracking will fail. The number of consistent depth pixels between the area in the bounding box of the current frame and the projection point of the corresponding area from the last frame has a proportion exceeding 0.8, and these two regions have the same ID. If this statistic is satisfied in multiple consecutive frames, the region is considered to be a static area.

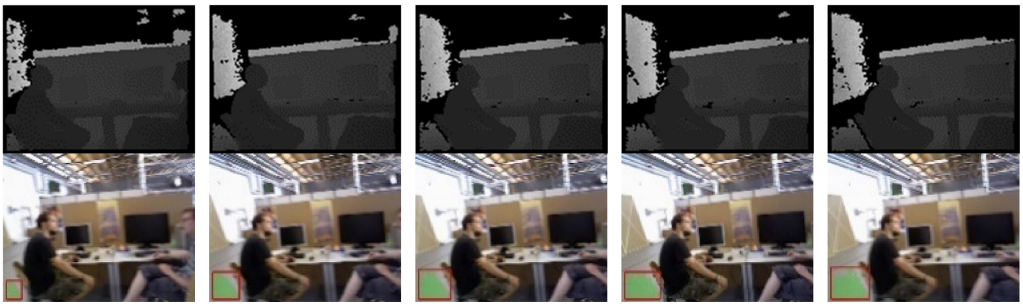


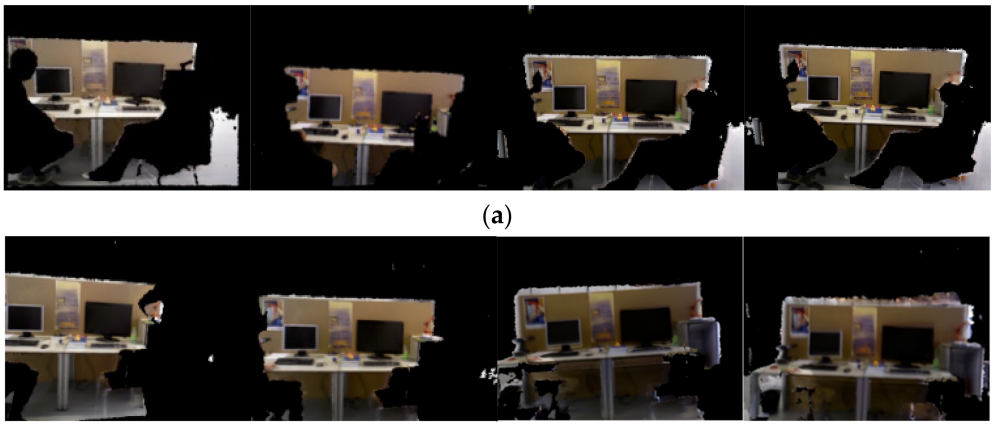
Figure 15. Bounding box tracking and static region determination.

3.10. Mesh Generation and Postprocessing

In highly dynamic scenes, dynamic objects stay in a position for a short time, and the number of points falling into the corresponding voxel is fewer, so we delete the voxels with fewer fusion points. By using this method, most of the traces left by the highly dynamic objects can be removed.

By the marching cube method, the mesh is obtained from the SDF field, and the small components in the mesh with a small surface volume are removed. Through this method, some small dynamic areas or noises are removed.

Figure 16 shows the incremental reconstruction process. We can see that dynamic objects are well-removed during reconstruction, and the static background area is incrementally reconstructed. Our algorithm reconstructs a good background map on both less and highly dynamic scenes.



(**a**)

(**b**)

Figure 16. Incremental background reconstruction process. (**a**) Incremental dense reconstruction of less dynamic scenes. (**b**) Incremental dense reconstruction of highly dynamic scenes.

4. Experimental Results and Discussion

This section evaluates our method on the TUM dataset and real scenes. Real scene sequences are collected through Kinect v2. All experiments are implemented on a computer with a GTX1060 GPU, an Intel(R) Core (TM)i7-7700HQ CPU, 16 GB RAM, and a Windows-10 64-bit operating system.

To obtain a reliable depth difference threshold τ_d, this paper uses plane extraction [37] to obtain planes on each depth map, each plane is considered to belong to the same object, and τ_d is calculated based on the depth difference of adjacent pixels in the same plane. First, the distance from the point to the center of the camera is calculated, and the point is stored in a bucket with an interval of 5 cm according to the distance. Then, the depth difference between the point and its adjacent points is calculated. After calculating all points in the dataset, the maximum difference value in each bin is selected as the difference value. In this paper, the difference value is fitted as a straight line, which is represented as

$=-0.733x + 24.753$; x (expressed in millimeters) represents the distance from a point to the center of the camera, and τ_d indicates the depth difference threshold between the point and its neighboring point.

To evaluate the proposed method more comprehensively, the experiment is divided into three parts: trajectory error, tracking rate, and the effect of the reconstruction model. Specifically, the absolute trajectory error (ATE) and the relative pose error (RPE) are used to evaluate our trajectory accuracy, the tracking rate is used to evaluate the tracking stability of our method, and static background dense reconstruction results are used to show the reconstruction ability of our method in the dynamic scene. Our method is compared with a series of state-of-the-art methods.

4.1. Experiments on TUM Datasets

4.1.1. ATE Evaluation Experiment

Table 1 shows the ATE of eight sequences in the TUM datasets. The "-" symbol in the table indicates that no corresponding results were found in related papers. The results show that ORB-SLAM2 has good accuracy in less dynamic scenes and poor accuracy in highly dynamic scenes. Compared with the ORB-SLAM2 algorithm, our method has a 22% improvement on average in a less dynamic environment and a 32.3% improvement on average in a highly dynamic environment, which demonstrates that the algorithm in this paper can effectively improve the trajectory accuracy. The results of our method on eight datasets differ slightly from the ground truth, which indicates that there is no obvious tracking drift when using our method. Our method is also compared with the current state-of-the-art algorithms, such as DS-SLAM, OFM-SLAM, DM-SLAM, Dyna-SLAM, and PLD-SLAM, in the sequences (Fr3_s_static, Fr3_w_rpy, and Fr3_w_half). The algorithm proposed in this paper achieves the best results, which are 0.3 mm, 2.7 mm, and 2.3 mm higher than the results achieved by the current best algorithm. In the scene Fr3_w_xyz, our method has the same accuracy as that of the current best method. The accuracy of our method in the sequences (Fr3_s_xyz, Fr3_s_rpy, Fr3_s_half, and Fr3_w_static) is worse than the current best results.

Table 1. ATE evaluation accuracy results of eight sequences in the TUM dataset (M).

Sequence	DS-SLAM	OFM-SLAM	DM-SLAM	Dyna-SLAM	ORB-SLAM2	PLD-SLAM	Refusion	Xie [28]	Liu [27]	Our Method
Fr3_s_static	0.0065	0.0134	0.0063	0.0064	0.0083	0.0063	0.009	0.007	0.0086	**0.0060**
Fr3_s_xyz	-	0.0130	-	0.013	0.0095	0.0092	0.040	0.013	**0.0090**	0.0117
Fr3_s_rpy	0.0187	**0.0160**	0.0230	0.0302	0.019	0.0220	-	0.043	0.0204	0.021
Fr3_s_half	0.0148	0.0257	0.0178	0.0191	0.035	**0.0145**	0.110	0.019	0.0149	0.0173
Fr3_w_static	0.0081	0.041	0.0079	0.0080	0.390	**0.0065**	0.017	0.010	0.0108	0.016
Fr3_w_xyz	0.0247	0.306	0.0148	0.0158	0.614	0.0144	0.099	**0.014**	0.0156	**0.0140**
Fr3_w_rpy	0.4442	0.104	0.0328	0.0402	0.973	0.2212	0.104	0.033	-	**0.0303**
Fr3_w_half	0.303	0.307	0.0274	0.0274	0.789	0.0261	-	0.028	0.0359	**0.0227**

Figures 17 and 18 show the trajectory error image of the eight sequences, where the red lines represent the difference between the ground truth and the estimated value. The first row is the result of our algorithm, and the second row is the result of ORB-SLAM2. In the four highly dynamic datasets, the result of our method is close to the ground truth values, and the result of ORB-SLAM2 has a large deviation from the ground truth values. In the four less dynamic datasets, the results obtained by our method and the ORB-SLAM2 method are all close to the ground truth values. Figures 17 and 18 are consistent with the results in Table 1.

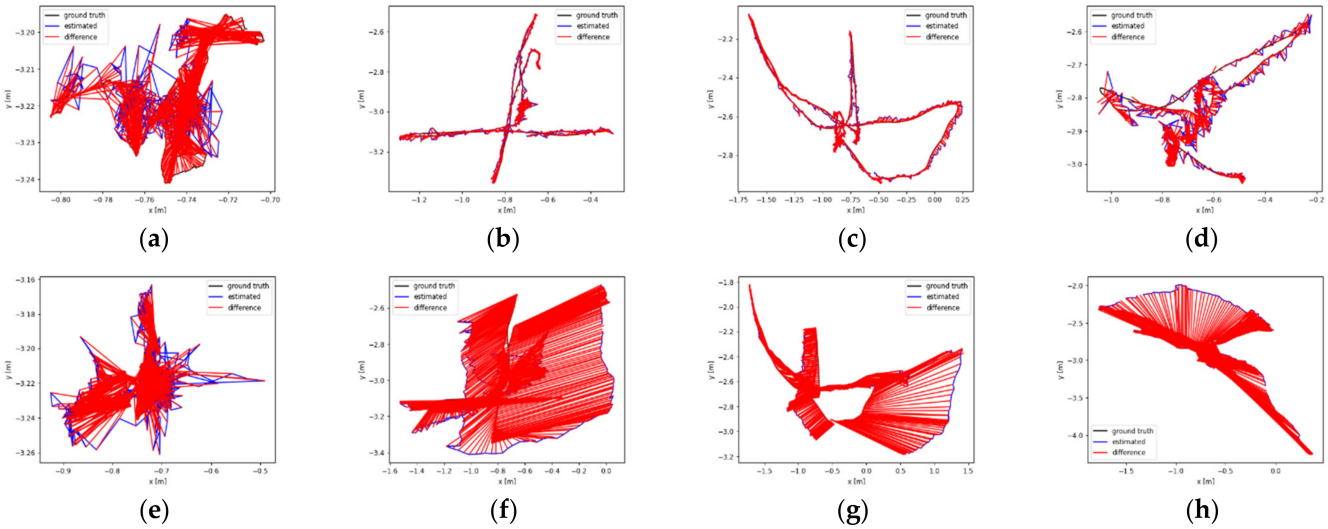


Figure 17. ATE comparison of four highly dynamic environment sequences in the TUM dataset. The first row shows the trajectories generated by our method. The second row shows the trajectories generated by ORB-SLAM2. (**a**) Fr3_w_static of our method. (**b**) Fr3_w_xyz of our method. (**c**) Fr3_w_half of our method. (**d**) Fr3_w_rpy of our method. (**e**) Fr3_w_static of ORB-SLAM2. (**f**) Fr3_w_xyz of ORB-SLAM2. (**g**) Fr3_w_half of ORB-SLAM2. (**h**) Fr3_w_rpy of ORB-SLAM2.

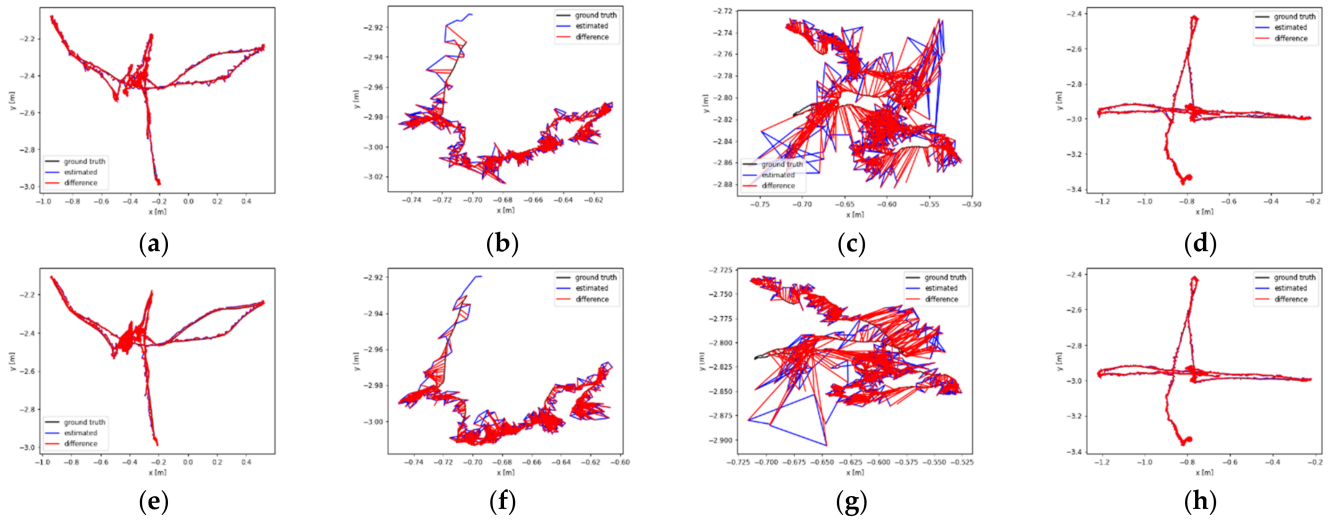


Figure 18. ATE comparison of four less dynamic environment sequences in the TUM dataset. The first row shows the trajectories generated by our method. The second row shows the trajectories generated by ORB-SLAM2. The four columns from left to right are Fr3_s_half, Fr3_s_static, Fr3_s_rpy, and Fr3_s_xyz. (**a**) Fr3_s_half of our method. (**b**) Fr3_s_half of our method. (**c**) Fr3_s_half of our method. (**d**) Fr3_s_half of our method. (**e**) Fr3_s_half of ORB-SLAM2. (**f**) Fr3_s_static of ORB-SLAM2. (**g**) Fr3_s_rpy of ORB-SLAM2. (**h**) Fr3_s_xyz of ORB-SLAM2.

4.1.2. RPE Evaluation Experiment

Table 2 shows the results of the relative translation error. On the sequences (Fr3_s_static, Fr3_w_xyz, Fr3_w_rpy, Fr3_w_half), the algorithm proposed in this paper has achieved the best results. On the scene Fr3_w_static, compared with the results published by Fan [8], the results obtained by our method are slightly worse.

Table 2. Relative translation error (M).

Sequence	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Fan [8]		Our Method	
	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
Fr3_s_static	0.0095	0.0046	0.0126	0.0067	0.0078	0.0038	0.0087	0.0038	**0.0073**	**0.0036**
Fr3_s_xyz	**0.0118**	**0.0057**	0.0147	0.0079	-	-	-	-	0.0143	0.0076
Fr3_s_rpy	**0.0264**	**0.0211**	0.0316	0.0191	-	-	-	-	0.0326	0.0185
Fr3_s_half	0.0229	0.0166	**0.0192**	**0.009**	-	-	-	-	0.0222	0.0107
Fr3_w_static	0.1928	0.1773	0.0089	0.0044	0.0102	0.0038	0.0102	0.0049	0.0144	0.0081
Fr3_w_xyz	0.4834	0.3663	0.0217	0.0119	0.0333	0.0229	0.0204	0.0107	**0.0182**	**0.0087**
Fr3_w_rpy	0.3880	0.2823	0.0448	0.0262	0.1503	0.1168	0.0616	0.0357	**0.0425**	**0.0239**
Fr3_w_half	0.3216	0.2629	0.0284	0.0149	0.0297	0.0152	0.0274	0.0140	**0.0243**	**0.0109**

Table 3 lists the relative rotation errors of several methods. In the sequences (Fr3_s_static, Fr3_w_xyz, and Fr3_w_rpy), the RMSE of our method is best. Although ORB-SLAM2 achieves the best results in the sequences (Fr3_s_xyz, Fr3_s_rpy, and Fr3_s_half), it performs poorly in highly dynamic scenes. In the scene Fr3_s_xyz, the results of our method are close to the current best results. In the sequences (Fr3_s_half, Fr3_s_rpy, Fr3_w_static, and Fr3_w_half), the results of our algorithm are worse than the current best results.

Table 3. Relative rotation error (radius).

Scene	Sequence	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Fan [8]		Our Method	
		RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.	RMSE	S.D.
Less Dynamic Scenes	Fr3_s_static	0.2881	0.1244	0.3416	0.1642	0.2735	0.1215	0.2782	0.1210	**0.2673**	**0.1178**
	Fr3_s_xyz	**0.4976**	**0.2772**	0.5162	0.2882	-	-	-	-	0.5062	0.2759
	Fr3_s_rpy	**0.7613**	**0.3954**	0.833	0.470	-	-	-	-	0.8206	0.4154
	Fr3_s_half	**0.576**	**0.2651**	0.649	0.3155	-	-	-	-	0.6945	0.3427
Highly Dynamic Scenes	Fr3_w_static	3.5991	3.2457	0.2612	0.1259	0.2690	0.1215	**0.2631**	**0.1119**	0.3336	0.1630
	Fr3_w_xyz	8.8419	6.6762	0.6284	0.3848	0.8266	0.2826	0.6227	0.3807	**0.6033**	**0.3749**
	Fr3_w_rpy	7.5906	5.4768	0.9894	0.5701	3.0042	2.3065	1.3831	0.8319	**1.0791**	**0.6658**
	Fr3_w_half	6.6515	5.3990	0.7842	0.4012	0.8142	0.4101	**0.7440**	**0.3459**	0.8101	0.3947

4.1.3. Results Analysis of the ATE and RPE

The ATE and RPE results show that most SLAM systems perform well in a less dynamic environment because the main area of the person is stationary or moving slightly, and the feature points on the person are used to participate in tracking; this can enhance tracking robustness and lower relative errors. This is also the reason why the RPE accuracy of ORB-SLAM2 on the two sequences Fr3_s_xyz and Fr3_s_rpy is higher than that of Dyna-SLAM, DS-SLAM, Fan [8], and our method. However, slow movement still affects the tracking. On the two sequences (Fr3_s_static and Fr3_s_half), compared with ORB-SLAM2, the method using mask optimization achieves better or comparable accuracy, thus reflecting that using the dynamic area mask to eliminate dynamic points can also adapt to the less dynamic scene.

On a highly dynamic scene, due to the interference of rapid human movement, the ORB-SLAM2 algorithm cannot accurately distinguish whether the feature point is a dynamic point or a static point and cannot perform correct tracking in a highly dynamic environment. When comparing ORB-SLAM2 and PLD-SLAM, Dyna-SLAM, and our method, the results show that the methods that use the dynamic object mask to filter out dynamic points can obtain better accuracy. When comparing Dyna-SLAM with PLD-SLAM and our method, we can find that the method that uses line structural constraints can obtain better ATE accuracy. Compared with PLD-SLAM, we use the line and plane structure constraints and the method of mask optimization to achieve better accuracy in the ATE and the translation part of RPE. Our method performs best on three of the four groups of highly

dynamic data, which shows that the mask optimization method and structural constraints method proposed in this paper can improve the tracking accuracy.

4.1.4. Evaluation of Tracking Rate

The tracking rate is defined as the successfully tracked ratio (STR), which represents the ratio between the number of successfully tracked frames (NSTF) and the total number of frames. STR reflects tracking stability.

Table 4 shows the tracking rates of ORB-SLAM2, Dyna-SLAM, DS-SLAM, Fan [8], and our algorithm. The table shows that our method tracked all the frames in the sequences (Fr3_s_static, Fr3_s_rpy, Fr3_s_half, Fr3_w_static, Fr3_w_xyz, and Fr3_w_half). Only in scene Fr3_w_rpy was the successfully tracked ratio of our method less than best because the algorithm proposed in this paper eliminates dynamic objects, and dynamic objects no longer participate in tracking. When dynamic objects occupy a larger part of the image, tracking will be less stable. The tracking results of eight sequences show that our method has achieved the best tracking rate as a whole. Compared with ORB-SLAM2, our method improves the successfully tracked ratio by 0.9% on average, and compared with Dyna-SLAM, our method improves the tracking rate by 17.8% on average, thus reflecting that our method uses accurate mask segmentation and fully uses the line and plane information in the scene, thereby improving the robustness of tracking.

Table 4. Tracking rate of eight sequences in the TUM dataset.

Scene	Sequence	Total	ORB-SLAM2		Dyna-SLAM		DS-SLAM		Fan [8]		Our Method	
			NSTF	STR	NSTF	STR	NSTF	STR	NSTF	STR	NSTF	STR
Less Dynamic Scenes	Fr3_s_static	679	675	99.4%	675	99.4%	676	99.6%	676	99.6%	**679**	**100%**
	Fr3_s_xyz	1219	1219	100%	1219	100%	-	-	-	-	**1219**	**100%**
	Fr3_s_rpy	795	773	97%	760	96%	-	-	-	-	**781**	**98%**
	Fr3_s_half	**1074**	**1074**	**100%**	**1074**	**100%**	-	-	-	-	**1074**	**100%**
Highly Dynamic Scenes	Fr3_w_static	717	714	99.6%	375	52.3%	714	99.6%	714	99.6%	**717**	**100%**
	Fr3_w_xyz	827	809	97.8%	757	91.5%	826	99.9%	826	99.9%	**827**	**100%**
	Fr3_w_rpy	866	825	95.3%	546	63.1%	**864**	**99.8%**	**864**	**99.8%**	858	98%
	Fr3_w_half	1021	942	99.3%	525	51.4%	1018	99.7%	1018	99.7%	**1021**	**100%**
Average STR		900	879	98.6%	741	81.7%	-	-	-	-	**897**	**99.5%**

4.1.5. Comparison of the Error Metric and Tracking Rate between the Items of the Proposed Method

To verify the contributions of different modules in our method to the stability and accuracy of tracking, an experiment was carried out, and the results are shown in Table 5. Our method is based on ORB-SLAM2, M represents the use of Mask-RCNN, CR represents the mask optimization method using connected component analysis and reference frame-based method, and LP represents the use of line and plane constraints. We can see that the performance of M + CR + LP is the best for all the sequences in terms of the tracking accuracy and successful tracking ratio. The results of the M + CR method are better than the results of method M because the mask segmented by M has an under-segmentation phenomenon, some dynamic points participate in tracking. M + CR refines the mask extracted by M, and these dynamic points are further excluded, so the overall accuracy is improved. After the mask optimization, some feature points with small movements on the moving person and the chair are also removed, so the overall accuracy is improved. In the sequence (Fr3_s_xyz, Fr3_s_half, Fr3_w_xyz, Fr3_w_half), the performances of M + CR and M + CR + LP are better than the performance of M in the tracking accuracy. In the sequence (Fr3_s_static and Fr3_w_static), we note that the results of the method M + CR + LP have slightly improved compared to the results of method M + CR. Yet, the contribution of the LP term is not so great because in these two sequences the camera is approximately

stationary relative to the scene, which helps predict the next pose of the camera and the feature point matching. In the static regions, some static feature points can be tracked from the beginning to the end, and in the dynamic regions, the feature points with large errors can be easily removed by projection error and a depth check. In the experiment, after mask optimization, some feature points with small movements on the moving person and the chair were also removed, so the overall accuracy was slightly improved.

Table 5. Comparison of the error metric and tracking rate between the items of the proposed method in the TUM dataset.

Sequence	Total Frame	M			M + CR			M + CR + LP		
		ATE	NSTF	STR	ATE	NSTF	STR	ATE	NSTF	STR
Fr3_s_static	679	0.0068	679	100%	0.0063	676	100%	**0.0060**	679	**100%**
Fr3_s_xyz	1219	0.0132	1219	100%	0.0126	1216	100%	**0.0117**	1219	**100%**
Fr3_s_rpy	795	0.0273	748	94.1%	0.0228	756	95.1%	**0.021**	781	**98%**
Fr3_s_half	1074	0.0207	1074	100%	0.0197	1074	100%	**0.0173**	1074	**100%**
Fr3_w_static	717	0.0172	717	100%	0.0167	717	100%	**0.016**	717	**100%**
Fr3_w_xyz	827	0.0212	827	100%	0.0182	827	100%	**0.0140**	827	**100%**
Fr3_w_rpy	866	0.0332	820	94.7%	0.0342	838	96.8%	**0.0303**	858	**98%**
Fr3_w_half	1021	0.0459	1021	100%	0.0366	1021	100%	**0.0227**	1021	**100%**

All frames can be tracked successfully on all sequences except for the sequences (Fr3_s_rpy and Fr3_s_rpy). The sequences (Fr3_s_rpy and Fr3_s_rpy) are a great challenge for robust tracking because of their complex camera movement. In the Fr3_w_rpy sequence, compared to method M, although more frames are tracked successfully by using M + CR, the tracking accuracy has decreased. The method M + CR + LP improves the tracking accuracy and tracks more frames, which indicates that the point, line, and plane constraints proposed can improve the tracking accuracy and stability.

4.1.6. Dense Reconstruction of the TUM Dataset

The eight sequences of the TUM dataset were reconstructed by using our method and the Refusion system. The results are shown in Figure 19; whether in the highly or lowly dynamic scene, our method can remove the dynamic objects well and obtain a clean reconstruction result for the static background. In the reconstruction results of Refusion, dynamic objects are reconstructed. Among the four highly dynamic sequences, there are many overlapping traces of human movement in the reconstruction results of highly dynamic scenes. In the results of owly dynamic scenes, dynamic objects such as people and chairs are all reconstructed. This experiment shows that our method can reconstruct the static background well.

Figure 19. Results of the static background reconstruction. The first row shows the result reconstructed by our method. The second row shows the results reconstructed by ORB-SLAM2. The eight columns from left to right are Fr3_w_static, Fr3_w_xyz, Fr3_w_rpy, Fr3_w_half, Fr3_s_xyz, Fr3_s_static, Fr3_s_rpy, and Fr3_s_half.

4.2. Experiments on Real Scenes

To prove the effectiveness of our method, we collected actual scene data for related experiments. Office desktops were used as the experimental scene. We fixed the depth camera in a position so that the ground truth pose of the depth camera was the identity matrix, and the person moved in front of the camera. To verify that our algorithm can work well in not only a highly dynamic environment but also a lowly dynamic environment, we designed two experiments: one is a highly dynamic scene, in which people move quickly in the scene; the other is a lowly dynamic scene, in which people move slowly around one position. In these two scenes, people also sometimes make contact with the static background. Kinect v2 is used to capture data sequences, and the sequences are collected at a frame rate of 30 Hz with a resolution of 424 × 512 pixels.

The experimental results are shown in Table 6. In a lowly dynamic scene, as shown in the left image in Figure 20a,c, ORB-SLAM2 is disturbed by dynamic objects. The path gradually deviates from the ground truth value in a circular shape. The left image in Figure 20a,c also shows that the result of ORB-SLAM2 has a large deviation from the ground truth value. As Figures 19d and 20b show, the ATE of our method is very close to the ground truth value in these two sequences. This paper also compares the effects of dense reconstruction. As shown in Figure 21, our method reconstructs the static background well in both highly dynamic scenes and lowly dynamic scenes, while part or all of the person is reconstructed by using the Refusion algorithm.

Table 6. ATE evaluation accuracy results on real scenes (M).

Sequence	ORB-SLAM2	Our Method
Lowly dynamic real scenes	0.1792	**0.004**
Highly dynamic real scenes	0.421	**0.0008**

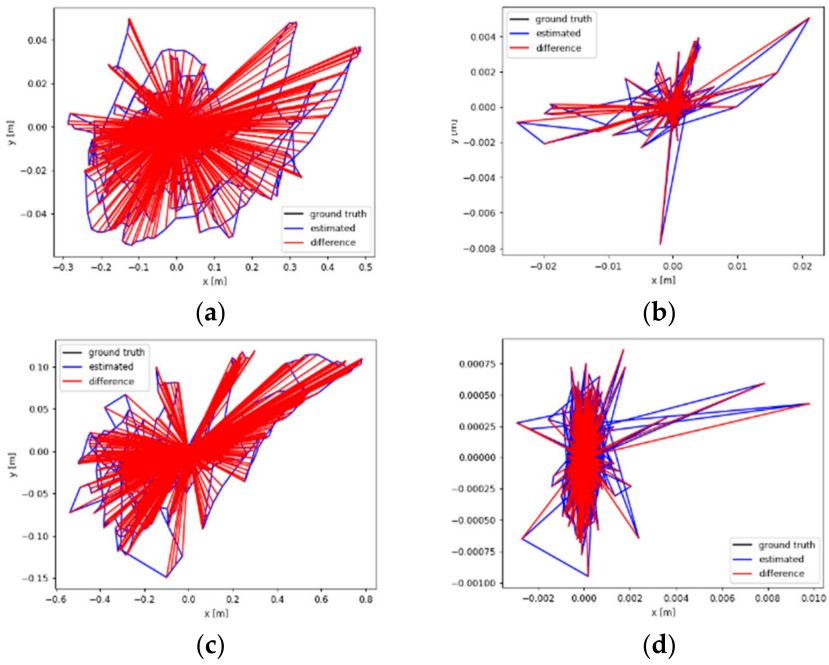


Figure 20. ATE for real scenes. The left column shows the results of ORB-SLAM2, and the right column shows the results of our method. (**a**) ATE of ORB-SLAM2 in lowly dynamic real scenes. (**b**) ATE of our method in lowly dynamic real scenes. (**c**) ATE of ORB-SLAM2 in highly dynamic real scenes. (**d**) ATE of our method in highly dynamic real scenes.

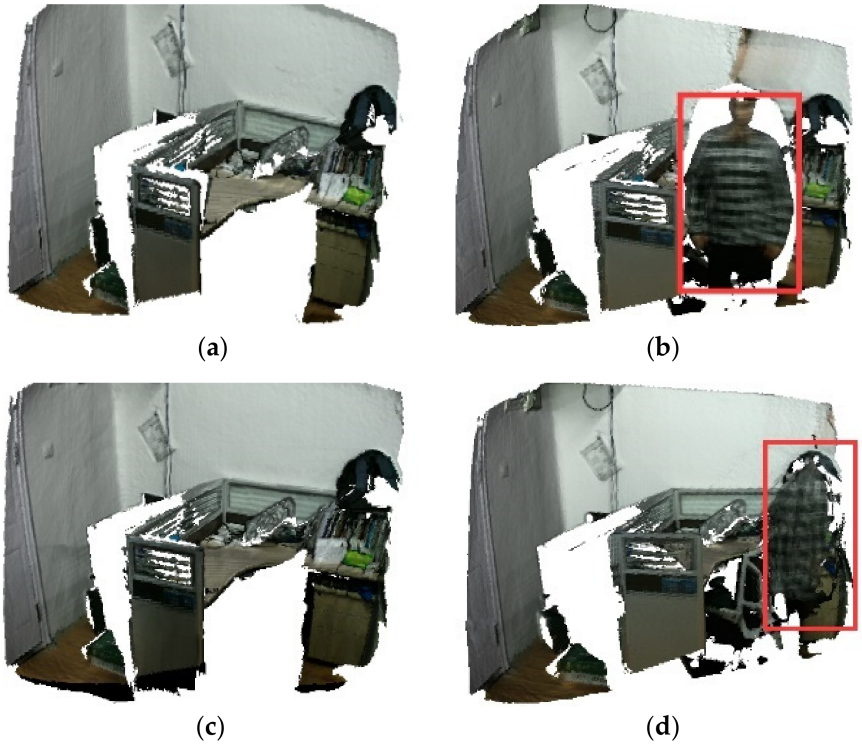


Figure 21. Dense reconstruction results. The left column shows the results of our method, and the right column shows the results of Refusion. (**a**) Dense reconstruction using our method in lowly dynamic real scenes. (**b**) Dense reconstruction using Refusion in lowly dynamic real scenes. (**c**) Dense reconstruction using our method in highly dynamic real scenes. (**d**) Dense reconstruction using Refusion in highly dynamic real scenes.

The results of the two sequences show that our method has higher camera tracking accuracy in real scenes and can reconstruct static backgrounds well.

5. Conclusions

In this paper, we proposed a robust and high-precision tracking and a clean static background reconstruction method. We divide our method into two stages. In the first stage, our method is based on ORB-SLAM2. To segment the dynamic object area accurately, the connected component segment method and a reference frame-based mask optimization method are used to refine the mask produced by Mask-RCNN. The dynamic features are filtered out by using the dynamic object mask and depth and projection check method, and feature points, lines, and planes in the nondynamic area are used to build optimization models to improve the robustness and accuracy of tracking. Sparse maps often fail to meet actual needs. Therefore, in the second stage, we construct a dense model of the static background by using TSDF fusion. First, the pose and mask information of the first stage are used for dense tracking and to obtain a depth map of the static area. Second, we designed algorithms to deal with divisible areas and indivisible areas. Static areas are extracted and fused into the map by using the TSDF method, the mesh of the static background is obtained by using the marching cube algorithm, and the reconstruction mesh is further optimized in the postprocessing. The experimental results on the public TUM datasets and real scenes prove that our method cannot only improve the tracking robustness and tracking accuracy but also reconstruct a clean and dense static background model in a dynamic environment.

Although our method has good performance, it also has some limitations in pose evaluation and background reconstruction. Mask-RCNN and connected component analysis methods are time-consuming, thus leaving our method unable to operate in real-time and limiting its practical use. Furthermore, we removed dynamic objects from the scene

without judging their specific motion state. When dynamic objects are static and occupy a large foreground, removing the dynamic objects will affect the stability of the system. The above-mentioned limitations will be addressed in our further research work.

Author Contributions: F.Z. and S.Z. conceived the idea and designed the methods; F.Z. and X.W. performed the experiments; F.Z. and X.H. analyzed the data; F.Z., S.Z., X.W. and X.H. wrote this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the National Natural Science Foundation of China (project number: 42101446).

Data Availability Statement: The TUM datasets we used are public datasets, which can be found here: http://vision.in.tum.de/data/datasets/rgbd-dataset (accessed on 12 July 2022).

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

References

1. Saputra, M.R.U.; Markham, A.; Trigoni, N. Visual SLAM and Structure from Motion in Dynamic Environments: A Survey. *ACM Comput. Surv.* **2018**, *51*, 37. [CrossRef]
2. Chang, J.; Dong, N.; Li, D. A Real-Time Dynamic Object Segmentation Framework for SLAM System in Dynamic Scenes. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 2513709. [CrossRef]
3. Sun, Y.X.; Liu, M.; Meng, M.Q.H. Improving RGB-D SLAM in dynamic environments: A motion removal approach. *Robot. Auton. Syst.* **2017**, *89*, 110–122. [CrossRef]
4. 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]
5. He, K.M.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
6. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *ITPAM* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
7. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
8. Fan, Y.C.; Zhang, Q.C.; Liu, S.F.; Tang, Y.L.; Jing, X.; Yao, J.T.; Han, H. Semantic SLAM With More Accurate Point Cloud Map in Dynamic Environments. *IEEE Access* **2020**, *8*, 112237–112252. [CrossRef]
9. Dai, W.; Zhang, Y.; Li, P.; Fang, Z.; Scherer, S. RGB-D SLAM in dynamic environments using point correlations. *ITPAM* **2020**, *44*, 373–389. [CrossRef]
10. Wang, Y.B.; Huang, S.D. Motion Segmentation based Robust RGB-D SLAM. In Proceedings of the World Congress on Intelligent Control and Automation (WCICA), Shenyang, China, 27–30 June 2014; pp. 3122–3127.
11. Liu, Y.; Miura, J. RDMO-SLAM: Real-time visual SLAM for dynamic environments using semantic label prediction with optical flow. *IEEE Access* **2021**, *9*, 106981–106997. [CrossRef]
12. Cheng, J.Y.; Sun, Y.X.; Meng, M.Q.H. Improving monocular visual SLAM in dynamic environments: An optical-flow-based approach. *Adv. Robot.* **2019**, *33*, 576–589. [CrossRef]
13. Brasch, N.; Bozic, A.; Lallemand, J.; Tombari, F. Semantic Monocular SLAM for Highly Dynamic Environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 393–400.
14. Wang, R.Z.; Wan, W.H.; Wang, Y.K.; Di, K.C. A New RGB-D SLAM Method with Moving Object Detection for Dynamic Indoor Scenes. *Remote Sens.* **2019**, *11*, 1143. [CrossRef]
15. Liu, G.H.; Zeng, W.L.; Feng, B.; Xu, F. DMS-SLAM: A General Visual SLAM System for Dynamic Scenes with Multiple Sensors. *Sensors* **2019**, *19*, 3714. [CrossRef] [PubMed]
16. Bescos, B.; Facil, J.M.; Civera, J.; Neira, J. DynaSLAM: Tracking, Mapping, and Inpainting in Dynamic Scenes. *IEEE Robot. Autom. Lett.* **2018**, *3*, 4076–4083. [CrossRef]
17. Zhang, C.Y.; Huang, T.; Zhang, R.C.; Yi, X.F. PLD-SLAM: A New RGB-D SLAM Method with Point and Line Features for Indoor Dynamic Scene. *Isprs Int. J. Geo-Inf.* **2021**, *10*, 163. [CrossRef]
18. MacQueen, J. Classification and analysis of multivariate observations. In Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, California, CA, USA, 27 December 1965–7 January 1966; pp. 281–297.
19. Yang, S.Q.; Fan, G.H.; Bai, L.L.; Zhao, C.; Li, D.X. SGC-VSLAM: A Semantic and Geometric Constraints VSLAM for Dynamic Indoor Environments. *Sensors* **2020**, *20*, 2432. [CrossRef]
20. Han, S.; Xi, Z. Dynamic scene semantics SLAM based on semantic segmentation. *IEEE Access* **2020**, *8*, 43563–43570. [CrossRef]
21. Cui, L.Y.; Ma, C.W. SOF-SLAM: A Semantic Visual SLAM for Dynamic Environments. *IEEE Access* **2019**, *7*, 166528–166539. [CrossRef]
22. Cui, L.Y.; Ma, C.W. SDF-SLAM: Semantic Depth Filter SLAM for Dynamic Environments. *IEEE Access* **2020**, *8*, 95301–95311. [CrossRef]

23. Yu, C.; Liu, Z.X.; Liu, X.J.; Xie, F.G.; Yang, Y.; Wei, Q.; Fei, Q. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1168–1174.
24. Cheng, J.; Wang, Z.; Zhou, H.; Li, L.; Yao, J. DM-SLAM: A feature-based SLAM system for rigid dynamic scenes. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 202. [CrossRef]
25. Zhao, X.; Zuo, T.; Hu, X. OFM-SLAM: A Visual Semantic SLAM for Dynamic Indoor Environments. *Math. Probl. Eng.* **2021**, *9*, 202–219. [CrossRef]
26. Xiao, L.; Wang, J.; Qiu, X.; Rong, Z.; Zou, X. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robot. Auton. Syst.* **2019**, *117*, 1–16. [CrossRef]
27. Liu, Y.; Wu, Y.L.; Pan, W.Z. Dynamic RGB-D SLAM Based on Static Probability and Observation Number. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 8503411. [CrossRef]
28. Xie, W.F.; Liu, X.P.; Zheng, M.H. Moving Object Segmentation and Detection for Robust RGBD-SLAM in Dynamic Environments. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 5001008. [CrossRef]
29. Ran, T.; Yuan, L.; Zhang, J.B.; Tang, D.X.; He, L. RS-SLAM: A Robust Semantic SLAM in Dynamic Environments Based on RGB-D Sensor. *IEEE Sens. J.* **2021**, *21*, 20657–20664. [CrossRef]
30. Ai, Y.B.; Rui, T.; Lu, M.; Fu, L.; Liu, S.; Wang, S. DDL-SLAM: A Robust RGB-D SLAM in Dynamic Environments Combined With Deep Learning. *IEEE Access* **2020**, *8*, 162335–162342. [CrossRef]
31. Zhang, L.; Wei, L.Q.; Shen, P.Y.; Wei, W.; Zhu, G.M.; Song, J. Semantic SLAM Based on Object Detection and Improved Octomap. *IEEE Access* **2018**, *6*, 75545–75559. [CrossRef]
32. Runz, M.; Buffier, M.; Agapito, L. MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects. In Proceedings of the IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Munich, Germany, 16–20 October 2018; pp. 10–20.
33. Xu, B.B.; Li, W.B.; Tzoumanikas, D.; Bloesch, M.; Davison, A.; Leutenegger, S. MID-Fusion: Octree-based Object-Level Multi-Instance Dynamic SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 5231–5237.
34. Scona, R.; Jaimez, M.; Petillot, Y.R.; Fallon, M.; Cremers, D. StaticFusion: Background Reconstruction for Dense RGB-D SLAM in Dynamic Environments. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 3849–3856.
35. 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 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 4–8 November 2019; pp. 7855–7862.
36. Von Gioi, R.G.; Jakubowicz, J.; Morel, J.-M.; Randall, G. LSD: A line segment detector. *Image Process.* **2012**, *2*, 35–55. [CrossRef]
37. Feng, C.; Taguchi, Y.; Kamat, V.R. Fast Plane Extraction in Organized Point Clouds Using Agglomerative Hierarchical Clustering. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 6218–6225.