Hindawi
Discrete Dynamics in Nature and Society
Volume 2020, Article ID 5916205, 14 pages
https://doi.org/10.1155/2020/5916205

Research Article

3D Semantic VSLAM of Indoor Environment Based on Mask Scoring RCNN

Chongben Tao,[1,2] **Yufeng Jin**,[1] **Feng Cao**,[3] **Zufeng Zhang**,[4,5] **Chunguang Li**,[6] **and Hanwen Gao**[1]

[1]*School of Electronic and Information Engineering, Suzhou University of Science and Technology, Suzhou 215009, Jiangsu, China*
[2]*Suzhou Automobile Research Institute, Tsinghua University, Suzhou 215134, Jiangsu, China*
[3]*School of Computer and Information Technology, Shanxi University, Taiyuan 030006, Shanxi, China*
[4]*Department of Automation, Tsinghua University, Beijing 100084, China*
[5]*Wuhan Electronic Information Institute, Wuhan 430019, Hubei, China*
[6]*School of Computer Information and Engineering, Changzhou Institute of Technology, Changzhou 213002, Jiangsu, China*

Correspondence should be addressed to Zufeng Zhang; yafflestudio@126.com

Received 30 June 2020; Revised 12 August 2020; Accepted 9 September 2020; Published 20 October 2020

Academic Editor: Jaime Zabalza

In view of existing Visual SLAM (VSLAM) algorithms when constructing semantic map of indoor environment, there are problems with low accuracy and low label classification accuracy when feature points are sparse. This paper proposed a 3D semantic VSLAM algorithm called BMASK-RCNN based on Mask Scoring RCNN. Firstly, feature points of images are extracted by Binary Robust Invariant Scalable Keypoints (BRISK) algorithm. Secondly, map points of reference key frame are projected to current frame for feature matching and pose estimation, and an inverse depth filter is used to estimate scene depth of created key frame to obtain camera pose changes. In order to achieve object detection and semantic segmentation for both static objects and dynamic objects in indoor environments and then construct dense 3D semantic map with VSLAM algorithm, a Mask Scoring RCNN is used to adjust its structure partially, where a TUM RGB-D SLAM dataset for transfer learning is employed. Semantic information of independent targets in scenes provides semantic information including categories, which not only provides high accuracy of localization but also realizes the probability update of semantic estimation by marking movable objects, thereby reducing the impact of moving objects on real-time mapping. Through simulation and actual experimental comparison with other three algorithms, results show the proposed algorithm has better robustness, and semantic information used in 3D semantic mapping can be accurately obtained.

1. Introduction

Simultaneous Localization and Mapping (SLAM) is a technology which enables robots or UAVs to realize autonomous positioning in an unknown environment and autonomous mapping. The robot can get rich information through sensors, which brings more conveniences to solve the problem of localization and mapping. Therefore, SLAM technology is undoubtedly a priority for robot autonomy. Compared with traditional SLAM based on laser sensor, SLAM based on camera vision can make full use of rich texture information on pictures taken by the camera, which provides a huge advantage in relocation and classification of scene semantic information. In recent years, intelligent robots have been widely used in various industries, especially for rapid development of Visual SLAM (VSLAM). Image feature extraction methods represented by deep learning technology have appeared in VSLAM. Meanwhile, deep learning also associates images with semantics and combines with VSLAM methods to build a semantic map and semantic knowledge base of environment. Salehi et al. [1] focused on the real-time fusion of monocular Vision SLAM and GPS data, where a hybrid method of constrained BA/position map is put forward to obtain the

attitude estimation and reconstruction of city scale and geographical parameters. Liu et al. [2] proposed a SSD algorithm based on YOLO and Faster RCNN, adding multiple convolution layers of different scales to maintain the accuracy of Faster RCNN, while a faster speed than YOLO is obtained. Zhang et al. [3] used collinear relationship of points to optimize the existing VSLAM algorithm based on points, and a practical line matching algorithm was given, where compensating computation assisted by straight beam was utilized and the perspective of n-point algorithm was improved. The proposed method is evaluated on indoor sequences of different ranges in the dataset of TUM and also compared with point-based and line-based methods. The results show that the designed algorithm has faster computing speed in comparison with VSLAM system based on point line. Gao et al. [4] proposed an improved method of augmented reality registration based on VSLAM to solve the problem of unstable registration and low registration accuracy of unmarked augmented reality of standard homographic matrix. The VSLAM algorithm generates a 3D scene map in the process of dynamic camera tracking, and then AR based on VSLAM uses 3D map of scene reconstruction to calculate the position of virtual object, which enables and enhances the stability and accuracy of AR registration.

Recently, robustness and availability of VSLAM technology have been strengthened, which tends to be mature [5]. However, sparse image features can provide limited environmental semantic information in dealing with dynamic target motion, lack of texture, or single texture environment. For these problems, hierarchical image feature extraction methods represented by deep learning have appeared in the field of VSLAM in recent years, providing ideas for solving such problems. By modeling bounding box of the most representative first-level detector YOLOv3 in accordance with Gaussian parameters and redesigning loss function, Choi et al. [6] proposed a method to improve detection accuracy and support real-time operation. Li et al. [7] put forward a multitarget detection framework integrating RCNN and DPM, which can precisely detect each single object among all objects in the image. Especially better performance was shown when objects are close to each other. Cai and Vasconcelos [8] developed a multilevel target detection structure, namely, Cascade RCNN, which includes a series of detectors trained by increasing IOU threshold, and higher selectivity for approaching misinformation is obtained. Ren et al. [9] proposed an improved anchoring scheme, where high resolution characterized mapping of small targets for improvement of its performance was used. Eggert et al. [10] introduced an improved scheme for generating anchor proposals and proposed a modification to Faster RCNN which leverages higher resolution feature maps for small objects. A novel multiscale location perception kernel representation (MLKP) method was presented by Wang et al. [11] to obtain the high-order statistics of depth features, and it combined discriminated high-order statistics into representation of object proposals for effective detection for objects. Note that this method can be applied to target detection flexibly. Li et al. [12] put forward a SOR Faster RCNN algorithm, which was used to search same target in different scenes with less training samples. A new robust Faster RCNN method was developed by Zhou et al. [13] to detect targets in multitag images. Unlike Fast RCNN, this design method has stronger robustness. Tao et al. [14] proposed a method of 3D environment semantic mapping based on Mask RCNN algorithm. The input image sequence was filtered by ORB-SLAM for key frame and then image semantic segmentation was combined with SLAM technology to build a 3D semantic map of the environment. Schorghuber et al. [15] fused a robust static weighting strategy based on corresponding distance of depth edge into intensity assisted ICP and thus proposed a real-time RGB-D visual range measurement method. Laidlo and Leutenegger [16] proposed a 3D reconstruction system called DeepFusion which leverages the output of a convolutional neural network (CNN) in DeepLab-v2 [17] to produce fully dense depth maps for key frames that include metric scale. DeepFusion fuses the output of a semidense multiview stereo algorithm with the depth and gradient predictions of a CNN in a probabilistic fashion, using learned uncertainties produced by the network. McCormac et al. [18] proposed an improved Elastic Fusion SLAM [19] method based on convolution neural network to build a dense 3D semantic map, which relies on Elastic Fusion SLAM algorithm to provide estimation for interframe pose of indoor RGB-D video, uses convolution neural network to predict classes and labels of pixel-level object, and finally combines Bayesian upgrading strategy and conditional random field model to realize probability upgradation of predicted CNN value from different perspectives so as to generate a dense 3D semantic map. Mur-Artal et al. [20, 21] proposed a ORB-SLAM2 method, which uses depth information to synthesize a three-dimensional coordinate, and the information of an image can be accurately extracted. The backend uses BA algorithm to build a global sparse map reconstruction. Therefore, this method is more lightweight and can be used in semantic mapping [22]. However, these aforementioned methods have some drawbacks in correctness of classification in the case of sparse feature points.

Motivated by the aforementioned existing problems, this paper proposed an ingenious semantic VSLAM algorithm combining BRISK feature [23] with a VSLAM algorithm based on Mask Scoring RCNN [24]. Semantic information of independent targets in scenes provides semantic information including categories. Meanwhile, the impact of moving objects during semantic mapping is reduced by the probability update of semantic estimation by marking movable objects.

2. Three-Dimensional Map Generation

2.1. System Overview. The overall architecture of the algorithm has two parts including front-end processing and back-end processing. A BRISK algorithm is used in front-end processing to extract features as well as key points. A Mask Scoring RCNN method is used in back-end processing including segmentation, semantic association, and semantic mapping as shown in Figure 1.

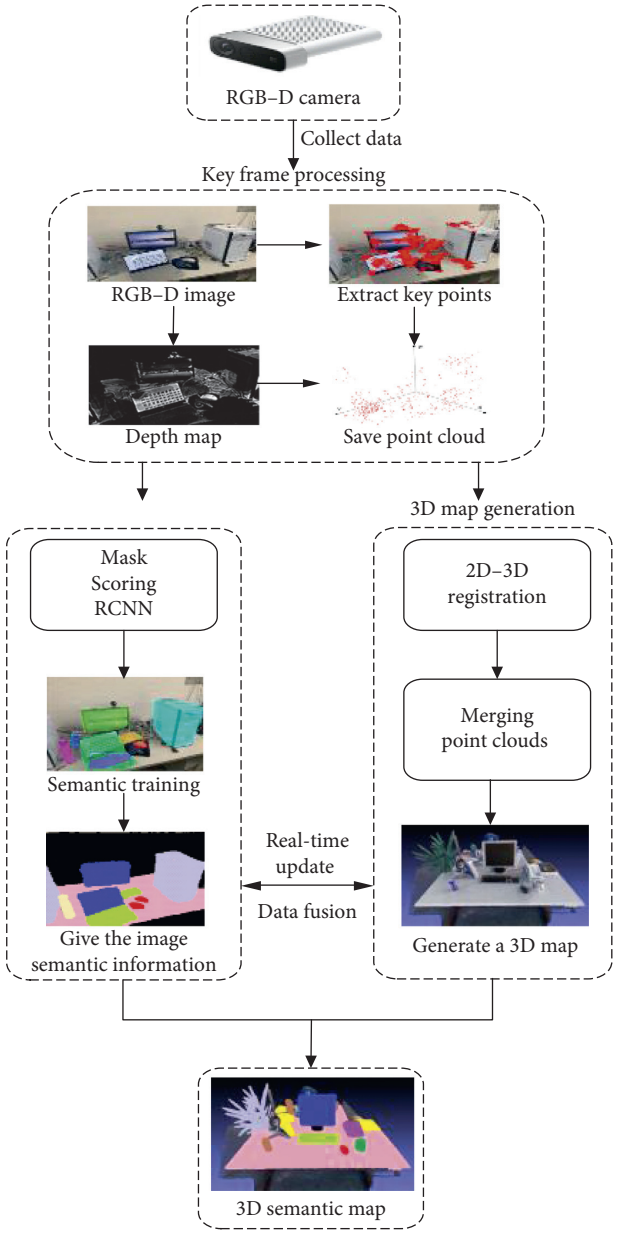

Figure 1: The entire framework of the proposed algorithm.

2.2. Dense SLAM Algorithm Based on BRISK Feature Extraction. Binary Robust Invariant Scalable Keypoints (BRISK) algorithm is similar to SIFT (scale-invariant feature transform), SURF (speeded-up robust feature), and ORB (oriented FAST and rotated BRIEF) [23], which is a feature point matching algorithm, but calculation speed is faster than other two algorithms. BRISK algorithm constructs image pyramid for multiscale expression, so it has good rotation invariance, scale invariance, good robustness, and so on. In particular, BRISK algorithm performs the best for image registration with large blurs. BRSIK algorithm consists of two parts: detection of key points and description of key points.

The detection for key points of BRISK is based on scale space composed of image pyramid. FAST is used to detect candidate key points in all layers of pyramid image, and candidate key points suppressed by nonmaximum are taken as final key points. After all the key points in image are obtained, key points need to be described. Different from descriptors constructed by SURF, SIFT, and other algorithms, BRISK algorithm applies binary string to describe key points so as to use Hamming distance to calculate matching degree and enable it to have a calculation speed faster than Euclidean distance. BRISK describes features in the mode of neighborhood sampling. The algorithm constructs multiple Bresenham concentric circles with key points as a center and takes N points evenly distributed to calculate feature direction and binary descriptors, respectively, in accordance with its long distance sampling points and short distance sampling points. Finally, Hamming distance is used to match above binary feature description so as to obtain global motion estimation of image.

In order to avoid the problem of sparse point cloud map caused by strict screening strategy to avoid gross error, this paper proposed a 3D mapping method of inverse depth filtering based on Visual SLAM. The task of inverse depth filter is used to estimate scene depth of created key frame and only build matching cost within depth range, which greatly reduces stereo matching time [25]. Based on the principle of depth similarity between adjacent pixels, after initial depth map is obtained, smoothing of intraframe and elimination of outer point are carried out, which increased density of depth map and eliminated possible isolated matching points. And Gaussian fusion is carried out for each candidate inverse depth hypothesis through an inverse depth fusion method of multikey frame to optimize current depth value of key frame. The specific algorithm steps are as follows:

Step 1: measuring for scene depth. Each map point observed by key frame at any time is projected into key frame image to calculate the depth value of the map point in the key frame coordinate system. Maximum depth and minimum depth are selected to set inverse-depth search range of scene.

$$p_i = (x_i, y_i, z_i)^T, \tag{1}$$

$$p_i^k = T_{k,w} p_i = \left(x_i^k, y_i^k, z_i^k\right)^T, \tag{2}$$

$$\begin{aligned} p_{\min} &= \min\left(z_i^{-k}\right), \\ p_{\max} &= \max\left(z_i^{-k}\right), \\ i &\in (0, n), \end{aligned} \tag{3}$$

where p_i is the homogeneous representation of 3D coordinates of map points in the world coordinate system; $T_{k,w} p_i$ is the pose transformation between the camera coordinate system and world coordinate system at time k; $p_i^k = T_{k,w} p_i$ is the homogeneous representation of 3D coordinates of map points in the camera coordinate system at time k; and N is the number of map points that can be observed in the key frame at time k.

Step 2: stereo matching. Pixel depth is calculated by using aggregate stereo matching algorithm of variable weight cost [26]. Based on layers of cost volume in the

limited stereo matching of scene depth value calculated in Step 1, it is only searched in the range of parallax opposite to inverse depth $(p_{\min}, p_{\max})$ so as to reduce the amount of calculation. Post-processing step of parallax deletion in stereo matching is eliminated at the same time, only retaining inverse depth of pixels with the same parallax in the left and right consistency matching.

Step 3: elimination of isolated outer point. It is assumed that parallax obtained by stereo matching follows the Gaussian distribution of variance 1, i.e., d: $N(d_0, 1)$:

$$p = z^{-1} = d(fb)^{-1}, \tag{4}$$

where d_0 is the parallax value calculated by stereo matching, f is the focal length of the camera, b is the baseline, z is the depth value of the pixel, and p is the inverse depth. The inverse depth distribution after transformation is as follows:

$$p\text{: } N\left(\frac{d_0}{fb}, \frac{1}{fb}\right). \tag{5}$$

The inverse depth map obtained in stereo matching stage is filled and isolated outliers are eliminated. The specific steps are as follows:

(1) For each pixel with inverse depth distribution, the number of pixels whose inverse depth distribution meets χ distribution of less than 5.99 is calculated. As shown in formula (6), inverse depth is eliminated in case of number less than 2. When the number is greater than 2, formula (7) is used to fuse the inverse depth that meets the requirements of χ distribution. After fusion, inverse depth of the pixel is p_p, while variance $\sigma_{p_p}^2$ is the minimum variance of inverse depth before fusion.

$$\frac{(p_x - p_y)^2}{\sigma_x^2} + \frac{(p_x - p_y)^2}{\sigma_y^2} < 5.9, \tag{6}$$

$$p_p = \frac{\sum_n \left(1/\sigma_{p_j}^2\right)}{\sum_n \left(1/\sigma_{p_j}^2\right)}, \tag{7}$$

$$\begin{aligned} p_p &= \frac{\sum_n \left(1/\sigma_{p_j}^2\right) p_j}{\sum_n \left(1/\sigma_{p_j}^2\right)}, \\ \sigma_p^2 &= \frac{1}{\sum_n \left(1/\sigma_{p_j}^2\right)}, \end{aligned} \tag{8}$$

where x and y are eight surrounding pixels around current pixel and n is a number which satisfies χ distribution.

(2) For each pixel that does not have an inverse depth distribution, check whether the inverse depth distribution between the eight surrounding pixels meets the chi-square distribution. When the number which satisfies χ distribution is greater than 2, formula (2) is used for inverse depth fusion, and homomorphic variance is the minimum variance of inverse depth before fusion.

Step 4: fusion of inverse depth. After position and pose of key frame are calculated by tracking thread, current depth information of key frame is optimized through following six inverse depth maps of the key frame. The specific steps are as follows:

(1) Project map point corresponding to inverse depth map of current key frame to adjacent key frame and read the inverse depth p_0 of projection point and inverse depth variance of σ_0^2.

(2) Map points whose inverse depth is $p_0 + \sigma_0$, p_0 and $p_0 - \sigma_0$ in the adjacent frame to current frame, and the reverse depth of p_1, p_2, and p_3 is retained.

(3) Construct candidate inverse depth of fusion, assuming ρ: $N(p_2, [\max(|p_1 - p_2|, |p_3 - p_2)]^2)$.

(4) Cycle above steps to obtain 6 candidate hypotheses of fusion inverse depth and select inverse depth hypothesis to be fused by using χ distribution less than 5.99. After fusion, inverse depth p_p and variance σ_p^2 are

where p represents pixels of current frame and n represents numbers of inverse depth to be fused.

Step 5: re-elimination of isolated outer point. Based on assumption that depth of adjacent areas in scene is similar, inverse depth map obtained by inverse depth fusion is smoothed in frame and removed from outer points so as to improve accuracy of output map points and increase density of point cloud. The specific steps are reverse depth filling and removal in Step 2.

Step 6: get cloud point map. All points in the processed depth graph are transformed to the global coordinate system, and point cloud map is obtained to construct current environment map. However, if point cloud data of each frame are integrated into map, a lot of computing resources will be occupied, thus reducing real-time performance of the system. Therefore, this paper uses point cloud map based on key frame to build dense environment map by dividing an entire map into several submaps with specific key frames to reduce memory consumption. The extracted key frame is optimized and saved to global map, and the dense global map is finally output, as shown in Figure 2.

For a key frame, the RGB-D camera provides color image and depth image. The formula of 3D point cloud in accordance with camera internal parameters is as follows:

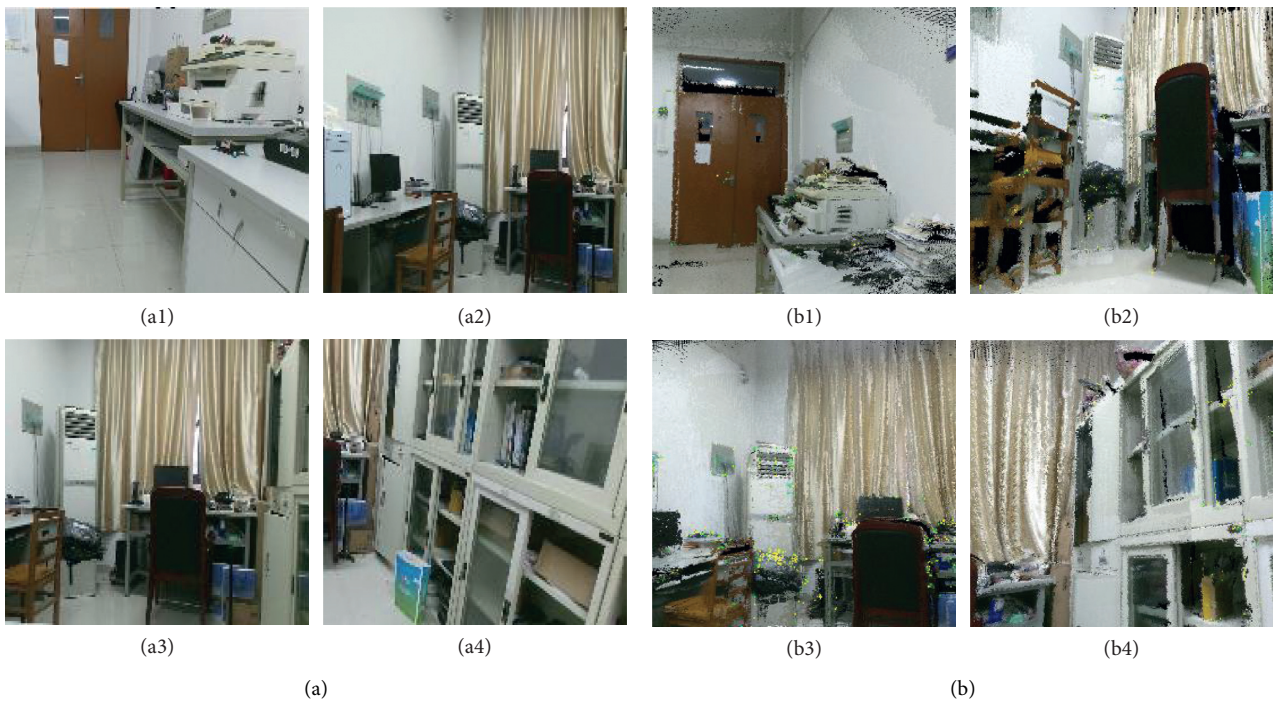

Figure 2: Key frame reconstruction in submaps. (a) Local image of four segments of experiments. (b) Local image of four segments of experiments after being constructed.

$$\begin{cases} z = \dfrac{d}{s}, \\ x = (u - c_x) \cdot \left(\dfrac{z}{f_x}\right), \\ y = (v - c_y) \cdot \left(\dfrac{z}{f_y}\right), \end{cases} \tag{9}$$

where f_x, f_y, c_x, c_y are internal parameters of the camera; (u, v) is the image coordinate; (x, y, z) is the image coordinate system; d is the distance of pixel point measured by the depth camera, with unit of mm; and s is the scale coefficient of actual distance and measured distance d. In this method, one of the advantages of point cloud is that it can be generated directly from RGB-D image efficiently without additional processing, with very intuitive operation of filtering (Algorithm 1).

3. Semantic Information Acquisition

The task of target detection includes classification and positioning, which not only gives the category information of an object to be detected but also determines position and size of the object in an image and surrounds it with a smallest rectangular frame. The main steps of target detection include preprocessing of input image and filtering of candidate areas of the image by a sliding window. Then, one kind of feature extraction algorithm is used including SIFT, HOG, or DPM to extract features for candidate areas, and finally a classification algorithm is used to classify extracted features. However, some defects such as unstable matching, weak antinoise ability, slow detection speed, and poor extraction effect for fuzzy and smooth edges coexist in the traditional object detection model. Compared with the traditional object detection model, the object detection model based on deep learning has more powerful feature expression ability, strong generalization ability, and good robustness.

A BMASK-RCNN network is designed in this paper which refers to Mask Scoring RCNN based on deep neural networks. Mask Scoring RCNN evolves from Mask RCNN, whose network framework is shown in Figure 3. The traditional Mask RCNN consists of two stages. The first stage is realized by convolution of RPN. Regardless of the object category, bounding box of a candidate object will be proposed. The second stage is called RCNN stage, which uses RoIAlign to extract features for each candidate where a bilinear interpolation is used to complete pixel-level alignment and finally generate candidate classification, bounding box regression, and mask prediction.

The loss function of Mask RCNN consists of three parts, namely, classification error, detection error, and segmentation error. The expression is as follows:

$$L = L_{\text{cls}} + L_{\text{box}} + L_{\text{mask}}, \tag{10}$$

where L_{cls} and L_{box} are the same with Faster RCNN; mask branch has dimensions of km^2 for each ROI, which indicates the solution is k binary masks with the solution of $m \times m$; K represents numbers of category, conducting sigmoid for each pixel; and L_{mask} is defined as average entropy loss of binary cross.

(1) **Input:** map point data x
(2) **Output:** point cloud map y
(3) The search range of scene depth measurement $(p_{\min}, p_{\max})$ is defined as $p_i = (x_i, y_i, z_i)^T$
(4) The scene depth value limits the number of layers of matching cost in stereo matching
(5) Isolated outlier culling while $p = z^{-1} = d(fb)^{-1}d$: $N(d_0, 1)$ then The inverse depth distribution after transformation is p: $N((d_0/fb), (1/fb))$
(6) Inverse deep fusion
(7) Isolated outlier secondary culling
(8) Get point cloud map

Algorithm 1: 3D mapping method for inverse depth filtering.

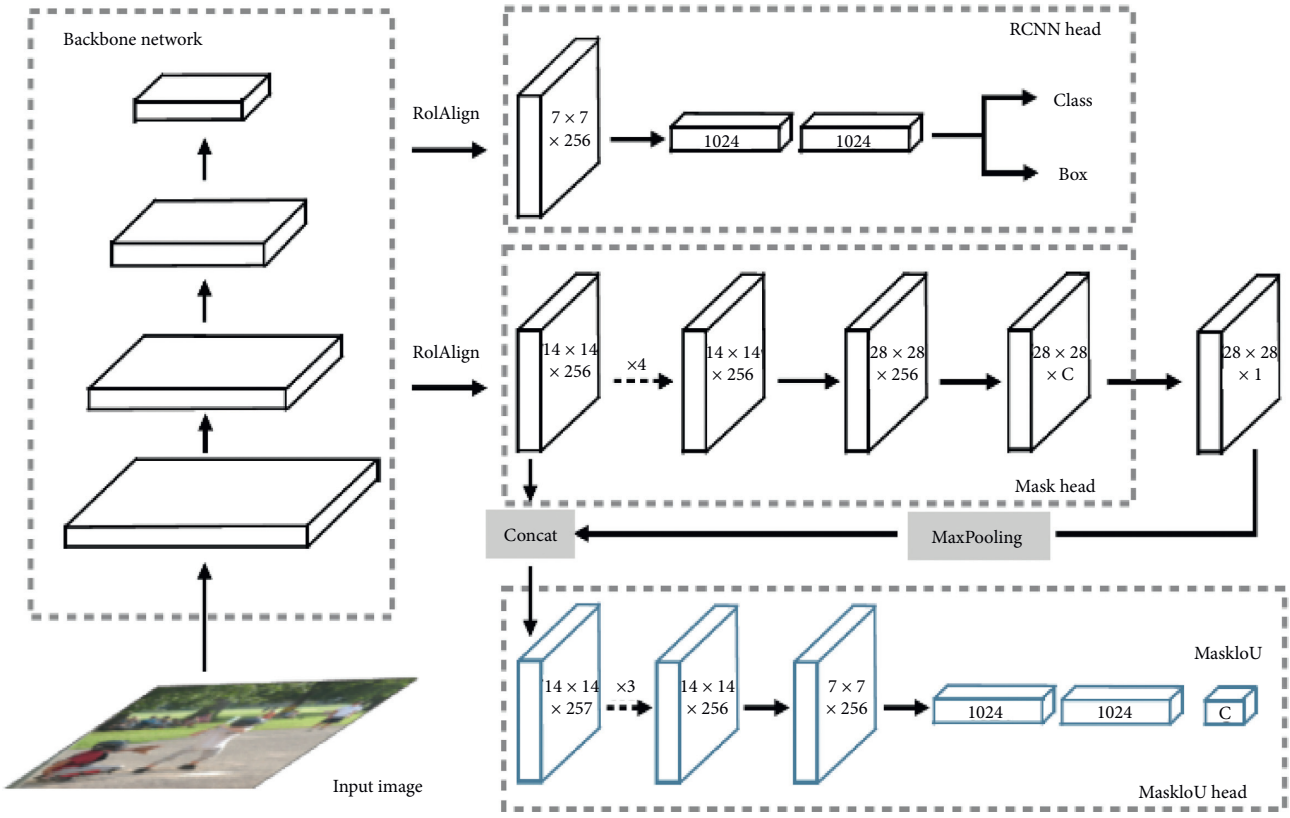

Figure 3: The network framework of Mask Scoring RCNN [24].

$$L_{\text{mask}} = -\frac{1}{m^2}\sum_{1\le i,j\le m}\left[y_{ij}\log\hat{y}_{ij}^{k} + \left(1-y_{ij}\right)\log\left(1-\hat{y}_{ij}^{k}\right)\right], \tag{11}$$

where y_{ij} is the label of cell (i, j) in the real mask within region of $m \times m$ and $\hat{y}_{ij}^{k}$ is the predicted value of the same cell in the k learning mask of ground truth value class.

However, the score for detecting (instance segmentation) hypotheses is determined by the largest element in its classification score in the current Mask RCNN framework. Due to clutter background, occlusion, and other problems, the score for classification may be high but mask quality is low. To overcome this problem, on the premise of generality of Mask Scoring RCNN, MaskIoU head module is added to enable the improved Mask RCNN for obtaining higher mask scores.

MaskIoU head is used to regress the IoU between predicted mask and its true label mask. For this purpose, feature concatenation and predicted mask of RoIAlign layer are used as input of MaskIoU head. A maximum pooling layer with a kernel size of 2 and a step size of 2 is used to make the predicted mask have the same space size as the RoI feature, and only MaskIoU is chosen to return to real label class. The MaskIoU head consists of four roll up layers and three fully connected layers. The four roll up layers follow mask head and set the kernel size and filter number of all the convolution layers to 3 and 256, respectively. Three fully connected (FC) layers follow RCNN head and set output of

first two FC layers to 1024 and the final FC output to the number of classes.

Table 1 is a comparison of results of MS RCNN algorithm and other algorithms on COCO test set, which shows that the MS RCNN algorithm has obvious advantages over other algorithms.

4. 3D Semantic Mapping Method

In the process of semantic mapping, VSLAM not only obtains geometric information in the environment but also recognizes independent individuals and obtains semantic information such as their position, posture, and functional attributes. The key of semantic VSLAM is to accurately recognize objects in the environment. Extracted features from target frame correspond to stored target object and map data, respectively, and then mapping relationship between the image data and the target object is established. The core idea of this paper can be expressed as follows: semantic information is extracted from key frames during the process of 3D mapping, and then the semantic information is fused into the constructed 3D map to create a new 3D semantic map. The flowchart of 3D semantic mapping is shown in Figure 4.

Firstly, Mask Scoring RCNN is used to train semantic database and then determine whether the current frame is a key frame. After key frame is determined, objects contained in semantic database in frame are detected and segmented, and then 2D image in the current key frame is semantically labeled. Finally, points containing semantic information in 2D image are mapped to 3D point cloud. It is regarded as the same object if there is the same semantic information.

For a system, system resources will be greatly consumed if all the image frames acquired by the camera are processed, so image key frame is usually selected for processing, and front-end tracking module of SLAM algorithm determines whether to select a current image frame as the key frame. Rules of key frame selection are as follows:

(1) There must be a sequence interval between the current frame and the previous key frame

(2) The thread of the local map is idle

(3) The current frame and previous key frame share a build area below a certain range

(4) The current frame has enough feature points to match, as shown in Algorithm 2

For each key frame, semantic information $X_t = \{X_k\}^N$ can be obtained through instance segmentation algorithm of Mask Scoring RCNN to obtain semantic information $X_t = \{X_k\}^N$, where $X_k = (x_k^a, x_k^b, x_k^c)$; x_k^a represents category of instance object; x_k^b represents outline of instance object; and x_k^c represents confidence level of instance object. The result of semantic acquisition for a key frame is shown in Figure 5.

The flowchart of semantic mapping is shown in Figure 4. After a key frame is selected, the key frame will be processed by two threads simultaneously: one is VSLAM algorithm, which runs according to original VSLAM system; the other is mainly the association and fusion process of object semantic. The obtained semantic information is processed in two aspects: on one aspect, feature points with dynamic category are marked as unavailable to reduce the impact of object movement on the mapping. On the other aspect, 2D image with semantic annotation information in the key frame is mapped to 3D point cloud so as to find mapping relationship between map points through finding feature points of object frame and semantic information. Algorithm 3 is used for data fusion.

5. Experiments and Analysis

5.1. Introduction to Experiment Platform. This experiment uses a self-built experimental platform, as shown in Figure 6, which is equipped with Microsoft Kinect 3.0 depth camera. The main body is composed of a main control unit, bracket, driving wheel, and chassis. The operating system adopts ROS (Robot Operating System) [30]. ROS is a robot-oriented open source operating system, which provides services including hardware abstraction, low-level device control, implementation of commonly functions, interprocess messaging, and package management. Operating frame is a processing architecture where ROS communication module is used to realize network connection of loose coupling between modules. It performs various types of communication, including service-based synchronous RPC (Remote Procedure Call) communication, topic-based communication of data flow, and data storage on parameter server. The mobile robot independently designed in this paper is a comprehensive experimental platform integrating environment perception, dynamic decision making and planning, behavior control, and execution. Deep learning and training are carried out in Ubuntu 18.04 system environment, with processor model of Intel i9-9900k and memory of 64 GB. In order to get higher training and testing speed, this paper uses GTX 2080Ti graphics card to accelerate training.

5.2. Verification Experiments. In order to prevent irrelevant semantic information from interfering with map construction, the network structure of Mask Scoring RCNN is adjusted. This experiment uses a TUM RGB-D SLAM dataset, where 24 types of objects are selected as shown in Table 2.

Since the onboard computer of the robot is not equipped with a GPU processor, the target detection algorithm of this paper is completed by a graphics workstation which uses TensorFlow as the framework. ROS is used for communication between the workstation and the robot. The graphics workstation is equipped with a GTX2080Ti graphics card for computing acceleration. After the key frame is detected, the semantic information of the target point cloud can be obtained according to the coordinate correspondence. The image of target detection and recognition effect and semantic map of dense point cloud are shown in Figure 7.

Comparisons of loss iteration curves for four algorithms are shown in Figure 8. The red line in Figure 8 represents the loss value of the proposed BMASK-RCNN method, and its

Table 1: Comparative results of MS RCNN algorithm and other instance segment algorithms on COCO testing set.

Method	Backbone	AP	AP@0.5	AP@0.75	APS	APM	APL
MNC [27]	ResNet-101	23.2	43.2	25.1	4.5	24.8	44.3
FCIS [28]	ResNet-101	28.9	48.7	—	—	—	—
FCIS+++ [28]	ResNet-101	34.2	53.7	—	—	—	—
Mask RCNN [14]	ResNeXt-101 FPN	36.9	61.2	38.6	17.1	38.7	52.4
MaskLab+ [29]	ResNet-101(JET)	37.8	62.4	41.0	18.2	40.9	50.7
Mask RCNN	ResNet-101	33.3	55.0	36.6	13.2	36.4	52.3
MS RCNN		35.4	54.9	38.1	13.7	37.6	53.3
Mask RCNN	ResNet-101 FPN	37.0	59.2	39.5	17.1	39.3	52.9
MS RCNN		38.3	58.8	41.5	17.8	40.4	54.4
Mask RCNN	ResNet-101-DCN+FPN	38.4	61.2	41.2	18.0	40.5	55.2
MS RCNN		39.6	60.7	43.1	18.8	41.5	56.2

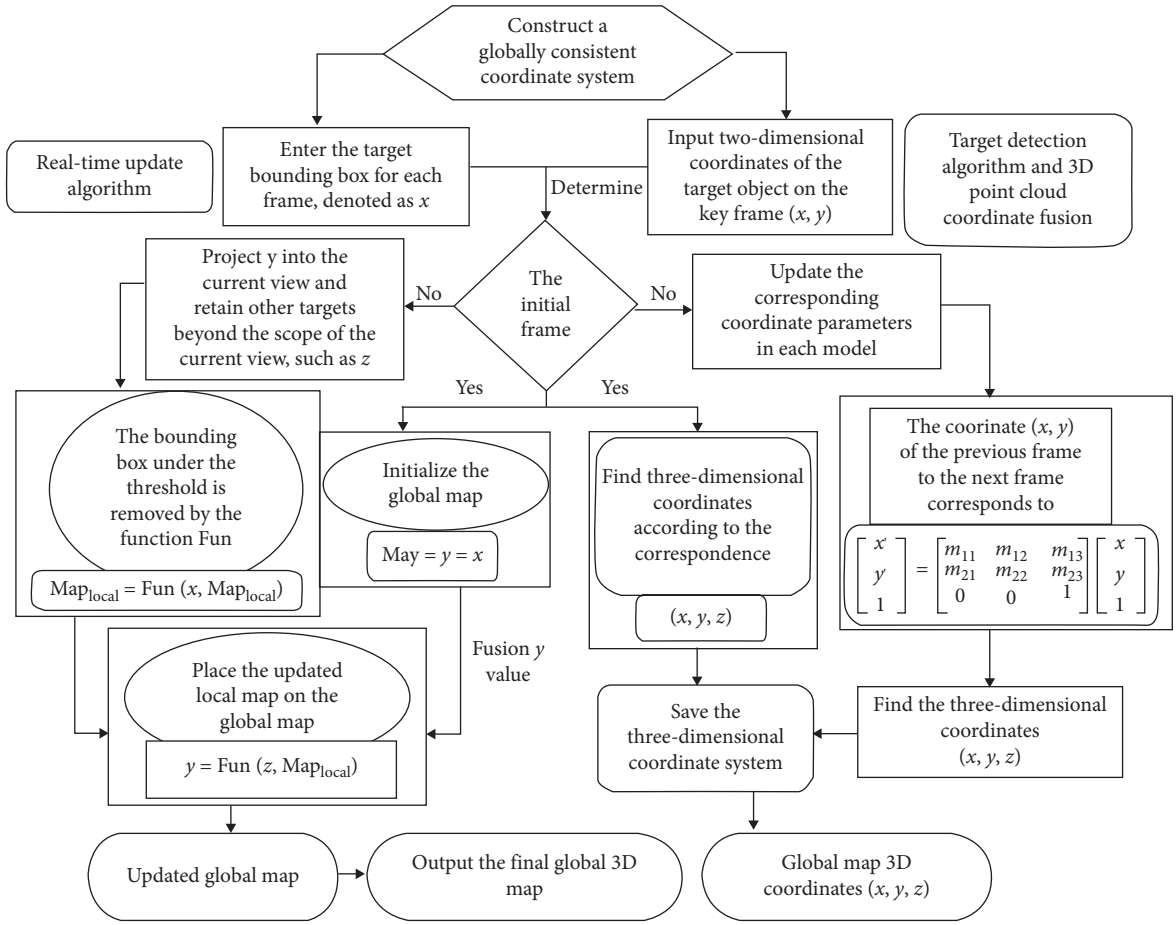

Figure 4: Flowchart for construction of semantic map.

loss value is always smaller than Fast RCNN and Faster RCNN. Although between 0.5×10^4 iterations and 1.5×10^4 iterations, the loss value of BMASK-RCNN is comparable to Mask RCNN, but after 15000 iterations, the curve of BMASK-RCNN stabilized below Mask RCNN. After 1.5×10^4 iterations, it can be seen that the proposed BMASK-RCNN method is more accurate than three methods.

Comparisons of precision-recall curves for four algorithms are shown in Figure 9, where the ordinate value represents detection accuracy of a measured target. The value of abscissa represents recall rate, namely, the total number of correctly detected targets divided by the total number of targets. Obviously, when the area under the curve is larger, the performance of the algorithm is better, and the detection effect is more accurate and complete. It can be seen from Figure 9 that the area under the precise recall curve of this algorithm is significantly larger than other three methods. Simulation results show that the proposed

Input: last key frame;
Output: new key frames;
(1) **if**
(2) (1) The interval between the current and previous key frame sequence is 30 frames;
(3) (2) Local map thread is idle;
(4) (3) The current frame and previous key frame share a build area threshold of less than 90%;
(5) (4) The number of matching point pairs is at least 100;
(6) Select as key frame;
(7) **else**
(8) discarded;
(9) **end if**

Algorithm 2: Key frame selection algorithm.

Figure 5: Acquisition of semantic information.

Input: feature points and semantic features on the current key frame ;
Output: 3D global semantic map coordinates;
(1) Coordinate system;
(2) Mark unusable points on map;
(3) Determine current frame;
(4) **if** initial frames then
(5) (1) Find map point coordinates corresponding to the target feature points;
(6) (2) Get semantic information about the target $p_k = (p_1, p_2, p_3)$, where p_1 is the category, p_2 is the confidence of the detection result, and p_3 is the target contour;
(7) (3) Semantic information is associated with geometric feature points through mapping relation so that feature points have both geometric and semantic information;
(8) (4) The relative motion of the camera is calculated according to feature matching, and the coordinates of the 3D map corresponding to the target feature points are found;
(9) **else**
(10) (5) The new parameters are substituted into the built model;
(11) (6) Insert a new key frame;
(12) (7) Repeat step (1), step (2), step (3), and step (4);
(13) (8) Save coordinate data;
(14) **end if**

Algorithm 3: Data association and fusion processes.

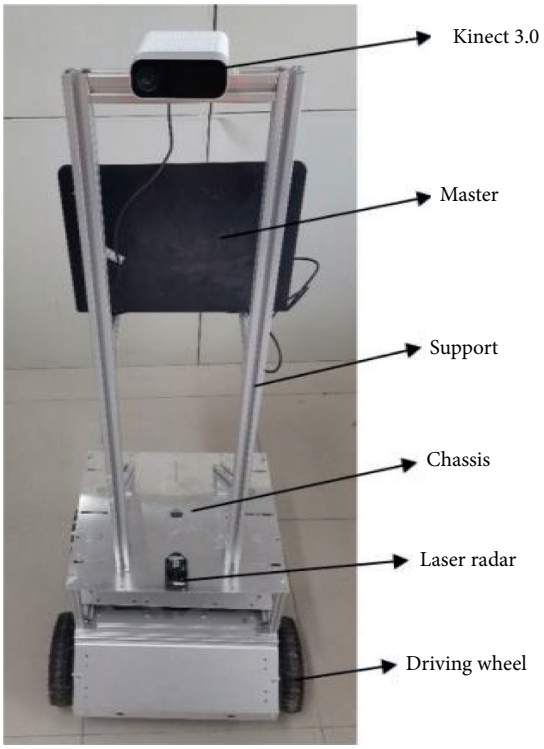

Figure 6: Experimental platform of robots.

Table 2: Selection for 24 types of objects.

Chair	Air conditioner	Screen	Robot	Desk	Bookcase
Door	Keyboard	Mouse	Drone	Cup	Person
Trophy	Switchbox	Bottle	Desk	Flower pot	Book
TV	Jackboard	Cell phone	Potted plant	Suitcase	Umbrella

(a) (b)

Figure 7: Effect image of target detection and recognition (a) and semantic map of dense point cloud (b).

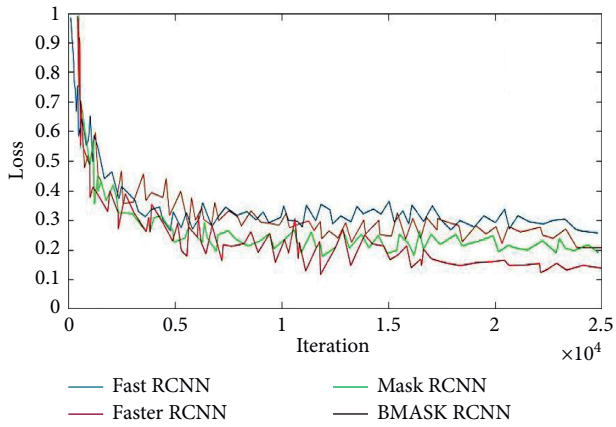

Figure 8: Curve comparative chart of loss iteration.

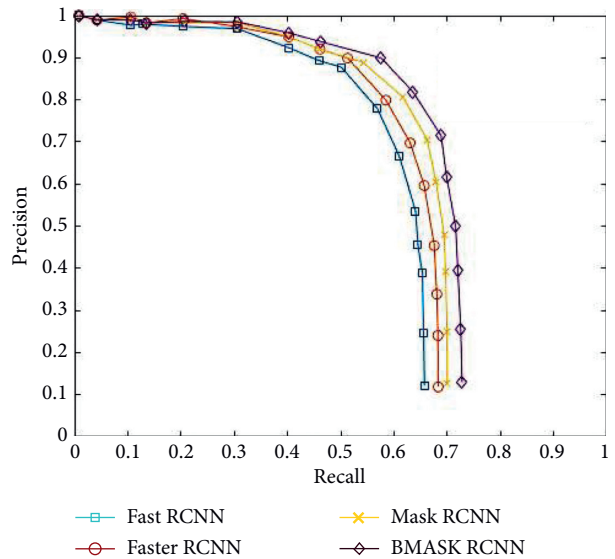

Figure 9: Curve comparative chart of precision and recall.

BMASK-RCNN method has higher accuracy than other three methods.

Figures 10 and 11 show error analysis graph generated using TUM dataset of freiburg2_large_with_Loop and freiburg1_XYZ to run the proposed VSLAM algorithm. freiburg1_xyz is a common small scenario dataset of TUM dataset, and freiburg2_large_with_loop is a large scene dataset from TUM. It can be seen from Figures 10 and 11 that overall effect of the proposed VSLAM algorithm is better than RGB-D SLAM. In the small environment, two systems have better stability; however, compared with RGB-D SLAM, the red lines representing errors in the absolute trajectory error diagram of the VSLAM algorithm are significantly reduced. In large scenarios with closed loops, under the influence of complex environment, RGB-D SLAM errors are relatively high and prone to drift. However, through the semantic information in the scene, the VSLAM algorithm can improve accuracy of mapping and localization, and thus the peak value of blue broken line in the relative pose error is small. In the same period of time, the peak value of broken lines is kept within 0.3 m, while the peak value of RGB-D SLAM lines reaches 0.8 m at most. The attitude error of the proposed VSLAM algorithm is closer to the same range, and the error is relatively low. In large scenarios, the performance of the proposed VSLAM algorithm is obviously better than RGB-D SLAM.

In order to obtain more accurate experimental results, the TUM RGB-D SLAM dataset is used which provides RGB-D images at a frame rate of 30 Hz, with a resolution of 640×480, as shown in Figure 12, for the operation effect.

The front-end part of the algorithm is the SLAM pose estimation and synchronous positioning module, which performs target detection tasks at the same time. Then, key frame pictures as well as all the useful data of key frame images including corresponding map points, semantic information, and position information are saved. Finally, data are transmitted to the server for data fusion calculation.

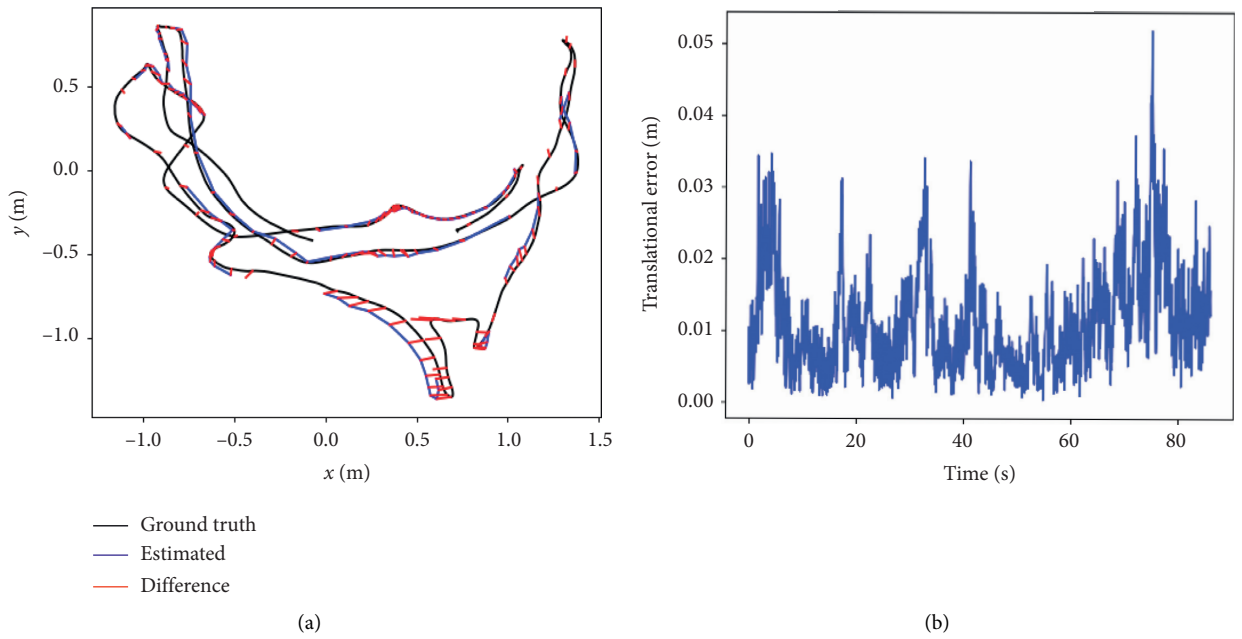

Figure 10: Error analysis of dataset freiburg1_xyz. (a) Absolute track error for construction of map. (b) Relative pose error for construction of map.

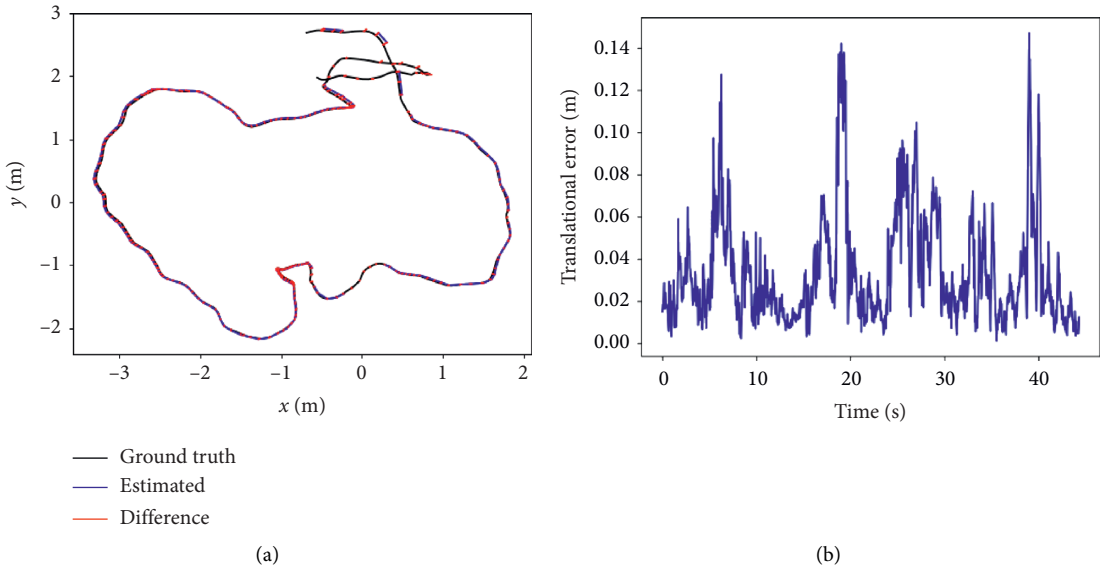

Figure 11: Error analysis of dataset freiburg2_large_with_loop. (a) Absolute track error for construction of map. (b) Relative pose error for construction of map.

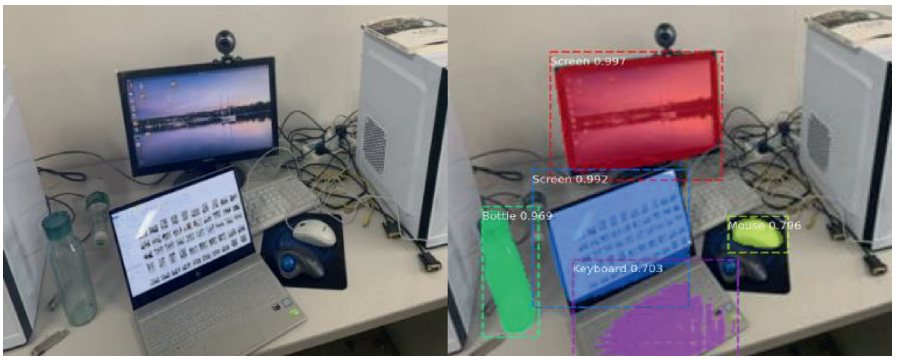

Figure 12: Operation images of the dataset.

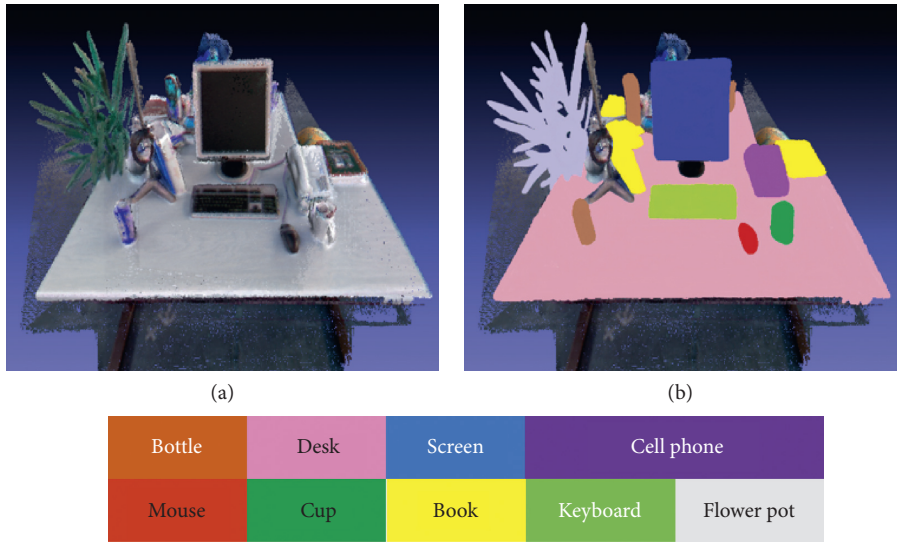

Figure 13: 3D map (a) and 3D semantic map (b).

Additionally, a semantic map is built in the robot in real time as shown in Figure 13.

6. Conclusion

This paper firstly uses a BRISK algorithm to extract feature points, then a Mask Scoring RCNN algorithm is used to detect targets and obtain semantic information of key targets in the environment, and the relative position relationship between target detection results is established. Then, targets are matched, and the similarity is calculated between key frames. Finally, the Mask Scoring RCNN algorithm is used to complete segmentation of targets, and a dense 3D semantic map surrounding the robot is constructed. The proposed method in this paper has achieved good results on the TUM RGB-D SLAM dataset and has verified the feasibility of the application of semantic information in Visual SLAM mapping. There is still room for improvement in this research. For example, the relationship between line and surface features in the target detection frame and the category of the corresponding object can be established to achieve stronger robustness and structure a semantic VSLAM system with better performance.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This study was supported in part by the National Natural Science Foundation of China (grant no. 61801323), the Science and Technology Project Fund of Suzhou (grant nos. SYG201708 and SS2019029), and the Construction System Science and Technology Fund of Jiangsu Province (grant no. 2017ZD066).

References

[1] A. Salehi, V. Gay-bellile, S. Bourgeois, and F. Chausse, "A hybrid bundle adjustment/pose-graph approach to VSLAM/GPS fusion for low-capacity platforms," in *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1728–1735, Los Angeles, CA, USA, June 2017.

[2] W. Liu, D. Anguelov, D. Erhan et al., "SSD: single shot multibox detector," in *Proceedings of the 2016 European Conference on Computer Vision*, Amsterdam, Netherlands, 2016.

[3] F. Zhang, T. Rui, C. Yang, and J. Shi, "Lap-slam: a line-assisted point-based monocular vslam," *Electronics*, vol. 8, no. 2, p. 243, 2019.

[4] Q. H. Gao, T. R. Wan, W. Tang, L. Chen, and K. B. Zhang, "An improved augmented reality registration method based on visual slam," *E-learning and Games*, vol. 10345, pp. 11–19, 2017.

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

[6] J. Choi, D. Chun, H. Kim et al., "Gaussian yolov3: an accurate and fast object detector using localization uncertainty for autonomous driving," in *Proceedings of the 2019 IEEE International Conference on Computer Vision*, pp. 502–511, Seoul, Republic of Korea, November 2019.

[7] J. Li, H.-C. Wong, S.-L. Lo, and Y. Xin, "Multiple object detection by a deformable part-based model and an R-CNN," *IEEE Signal Processing Letters*, vol. 25, no. 2, pp. 288–292, 2018.

[8] Z. Cai and N. Vasconcelos, "Cascade RCNN: delving into high quality object detection," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6154–6162, Salt Lake City, UT, USA, June 2018.

[9] S. Ren, K. He, R. Girshick et al., "Faster RCNN: towards real-time object detection with region proposal networks,"

Advances in Neural Information Processing Systems, vol. 39, pp. 91–99, 2015.

[10] C. Eggert, D. Zecha, S. Brehm et al., "Improving small object proposals for company logo detection," in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pp. 167–174, Bucharest, Romania, June 2017.

[11] H. Wang, Q. Wang, M. Gao et al., "Multi-scale location-aware kernel representation for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1248–1257, Salt Lake City, UT, USA, June 2018.

[12] H. Li, Y. Huang, and Z. Zhang, "An improved faster R-CNN for same object retrieval," *IEEE Access*, vol. 5, pp. 13665–13676, 2017.

[13] T. Zhou, Z. Li, and C. Zhang, "Enhance the recognition ability to occlusions and small objects with Robust faster R-CNN," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 11, pp. 3155–3166, 2019.

[14] C. Tao, Z. Gao, J. Yan, C. Li, and G. Cui, "Indoor 3D semantic robot VSLAM based on mask regional convolutional neural network," *IEEE Access*, vol. 8, pp. 52906–52916, 2020.

[15] M. Schorghuber, D. Steininger, Y. Cabon et al., "SLAMANTIC-leveraging semantics to improve VSLAM in dynamic environments," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 3759–3768, Seoul, Republic of Korea, October 2019.

[16] T. Laidlo and C. S. Leutenegger, "DeepFusion: real-time dense 3D reconstruction for monocular SLAM using single-view depth and gradient predictions," in *Proceedings of the 2019 International Conference on Robotics and Automation (ICRA)*, pp. 1231–1240, 2019.

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

[18] J. McCormac, A. Handa, A. Davison et al., "Semanticfusion: dense 3D semantic mapping with convolutional neural networks," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 2017.

[19] T. Whelan, R. F. Salas-Moreno, B. Glocker et al., "ElasticFusion: real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2019.

[20] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

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

[22] L. Li, Z. Liu, Ü. Özgüner et al., "Dense 3D semantic SLAM of traffic environment based on stereo vision," in *Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV)*, pp. 965–970, Dearborn, MI, USA, 2018.

[23] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: binary robust invariant scalable keypoints," in *Proceedings of the 2011 International Conference on Computer Vision*, pp. 2548–2555, Barcelona, Spain, November 2011.

[24] Z. Huang, L. Huang, Y. Gong et al., "Mask scoring RCNN," in *Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, 2019.

[25] C. Forster, Z. Zhang, M. Gassner et al., "SVO: semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, no. 2, pp. 249–265, 2016.

[26] P. Jianjian and B. Ruilin, "Variable weight cost aggregation algorithm for stereo matching based on horizontal tree structure," *Acta Optica Sinica*, vol. 38, no. 1, pp. 115–125, 2018.

[27] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3150–3158, Las Vegas, NV, USA, June 2016.

[28] Y. Li, H. Qi, J. Dai et al., "Fully convolutional instance-aware semantic segmentation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2359–2367, Honolulu, HI, USA, July 2017.

[29] L. C. Chen, A. Hermans, G. Papandreou et al., "Masklab: instance segmentation by refining object detection with semantic and direction features," in *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4013–4022, Salt Lake City, UT, USA, June 2018.

[30] M. Quigley, K. Conley, B. Gerkey et al., "ROS: an open-source robot operating system," *ICRA Workshop on Open Source Software*, vol. 3, no. 2, pp. 1–5, 2009.