A Semantic OctoMap Mapping Method Based on CBAM-PSPNet

Xiaogang Ruan[1,2], Peiyuan Guo[1,2] and Jing Huang[1,2,*]

[1]*Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*Beijing Key Laboratory of Computational Intelligence and Intelligence System, Beijing 100124, China*
E-mail: huangjing@bjut.edu.cn
[*]*Corresponding Author*

Received 05 August 2021; Accepted 16 December 2021;
Publication 08 March 2022

Abstract

With the rapid development of computer vision and deep learning, researchers have begun to focus on the semantic characteristics of traditional Simultaneous Localization And Mapping in three-Dimensional scenes. The point cloud map generated by the traditional simultaneous localization and mapping method takes up considerable storage space and cannot extract semantic information from the scene, which cannot meet the requirements of intelligent robot navigation and high-level semantic understanding. To solve this problem, this paper proposes a semantic information fusion OctoMap method. First, the color and depth images obtained from RGB-D by ORB-SLAM2 are used to locate the camera. Second, the Convolutional Block Attention Module-Pyramid Scene Parsing Network is introduced to segment the input RGB image semantically to improve the segmentation accuracy and obtain high-level semantic information in the environment. Then, a semantic fusion algorithm based on Bayesian fusion is introduced to fuse multiview semantic information. Finally, the generated semantic point cloud is inserted into OctoMap, and its octree data structure is used to compress the storage

Journal of Web Engineering, Vol. 21_3, 879–910.
doi: 10.13052/jwe1540-9589.21315

space. Experimental results based on the ADE20K dataset show that, compared with Pyramid Scene Parsing Network, Convolutional Block Attention Module-Pyramid Scene Parsing Network improves Mean Pixel Accuracy by 2.55%, and Mean Intersection over Union by 1.88%. Experimental results based on the TUM dataset show that the proposed method greatly reduces storage space and achieves the effect of voxels by voxel dense semantic mapping compared with point clouds and a traditional OctoMap.

Keywords: SLAM, semantic mapping, OctoMap, semantic segmentation.

1 Introduction

The perception and modeling of the mobile robot environment is the basis of navigation-oriented synchronous positioning and map building systems [1]. It is widely used in various fields, such as indoor robots, AR, UAVs, and unmanned driving. For mobile robots to navigate in an unknown environment, they need to be able to access a world model that transmits both geometric and semantic information. To complete more complex tasks, robots need to know not only the location information of rooms and objects in the environment but also the high-level semantic information of the environment. For example, as shown in Figure 1, if an indoor robot tries to enter a room with the door closed, it not only needs to know where it is to reach the door and where the obstacles are. More importantly, the robot should know where the door is, i.e. it should distinguish the door from other objects by semantics, so it should be able to recognize it. After the robot enters the door, the common tables, chairs, bookcases, sofas, and walls in the room are also needed to be distinguished by semantics. The simple case indicates that not only the position but also the semantics of objects are important for robot navigation. However, in recent years, research in the field of SLAM has mainly focused on geometric mapping or the mapping of very few semantic classes. Our work will implement the mapping of dense semantic classes through semantic segmentation technology.

Semantic segmentation is an important technology that utilizes advanced semantic features of images to predict and classify image pixels. As a basic link in the frontier fields of computer vision research such as image understanding and image generation, semantic segmentation has been widely used in autonomous driving, medical imaging, and other industries, which has important research significance and application value. The traditional semantic segmentation method uses the underlying features of the image to

Figure 1 Diagram of indoor scene simulation.

divide the image into regions, but the segmentation accuracy is not ideal. Fully convolutional networks (FCN) [3] were proposed to replace the fully connected layer with the convolutional layer, which successfully applied the convolutional neural network in the semantic segmentation task and achieved a good effect. The unity networking (UNet) [4] model, based on an encoder and decoder structure, avoids the direct supervision and loss calculation in a high-level feature map but combines the features in a low-level feature map to achieve feature fusion at different scales. The pyramid scene parsing network (PSPNet) [5] model integrates more context information by introducing a pyramid pooling module. Most of these models are based on the improvement of the FCN model, which solves the problem of feature selection in traditional semantic segmentation methods to a certain extent. However, they only decode the high-level feature map with a smaller resolution and ignore the influence of the low-level spatial details on image segmentation. Therefore, there are still many challenges in complex scene applications, such as the segmentation of small targets, strip parts and fuzzy edge contour, and the differentiation of similar parts of different targets. A convolutional block attention module [6] (CBAM) calibrates feature weights from both channel and spatial dimensions and improves the network recognition effect based on multidirectional feature enhancement. It integrates seamlessly into any CNN architecture, with minimal overhead, and can be trained end-to-end with a CNN. Taking PSPNet as the main framework and integrating the CBAM module, this paper proposes a network model based on CBAM-PSPNet. The model can effectively highlight the key areas while discovering the subtle features, which can give full play to the common advantages of CBAM and PSPNet.

There are many types of map representations in SLAM systems, such as feature point maps, grid maps, topological maps, point cloud maps, and occupancy maps. In the research and development of visual SLAM, a 3D spatial map is usually constructed by using a camera to obtain environmental information, and then through the process of feature extraction, descriptor

matching, and RANSAC, the extracted image feature information is converted to global coordinates, and a 3D point cloud map is presented. When the intelligent robot uses the point cloud map to complete the navigation, there will be obvious bottlenecks. On the one hand, the point cloud map has a large scale and stores many unnecessary details, such as folds on the blanket and shadows at the corner of the table. These detail points usually do not affect the location or pose optimization, so they take up considerable storage space and have low storage efficiency. An intelligent robot with limited processing speed and storage space will lead to serious time consumption and poor processing performance. On the other hand, as the point cloud map only models the surface of the existing object, it is difficult to distinguish between the free areas and the unknown areas, leading to the inability of the intelligent robot to effectively use the 3D point cloud map for autonomous navigation. OctoMap [2], based on an octree, has the advantages of small memory, high storage efficiency, and real-time update. Each voxel of OctoMap updates the occupancy rate from different measurements in a probabilistic way, but OctoMap lacks high-level semantic information, and there is no semantic color and semantic confidence information stored in the voxels.

To solve these problems, a semantic OctoMap mapping method based on CBAM-PSPNet is proposed. First, our method changes the map representation of point clouds used in most SLAM systems and uses a flexible and compressible OctoMap to model 3D space. Second, a traditional OctoMap only contains space occupation information but lacks semantic information, and the proposed semantic OctoMap can obtain high-level semantic information in the environment through the CBAM-PSPNet model. By integrating the CBAM module based on PSPNet, semantic information accuracy is improved. Finally, a semantic fusion algorithm based on Bayesian fusion is used to fuse the semantic information from multiple perspectives. By storing the classes with high probability, the problem that the number of classes is too large to be updated is solved. The TUM datasets are used to demonstrate and evaluate the capabilities of our semantic OctoMap system in indoor environments of different scales and compared with 3D dense point cloud maps and traditional OctoMap maps for mapping the effectiveness and storage footprint. In an experiment, we demonstrate and evaluate the function of our semantic OctoMap system in indoor environments of different scales by using TUM datasets and compare the semantic OctoMap with a 3D dense point cloud map and traditional OctoMap in terms of mapping effects and storage space. The segmentation performance of CBAM-PSPNet, PSPNet, and FCN is compared using the ADE20K dataset, and the modeling quality

of the PSPNet model and the CBAM – PSPNet model applied in the semantic OctoMap is compared. We also compare the mapping effects of the semantic OctoMap at different resolution sizes.

The main contributions of this paper are as follows:

1. An OctoMap mapping method containing semantic information is proposed to represent the semantic classes contained in the environment. A semantic segmentation network is used to obtain semantic classes in the environment and generate a semantic point cloud. After transforming the point cloud into a semantic OctoMap, each voxel can be assigned a semantic class to obtain the dense classification of the voxel level. The model can be applied to the SLAM system, which makes it possible for the robot to use high-level semantic information for navigation.
2. The proposed CBAM-PSPNet model can effectively highlight the key areas while discovering the subtle features, which can give full play to the common advantages of CBAM and PSPNet. Because the CBAM-PSPNet model retains the shallow information of the small target at the edge of the object in the encoding and feature fusion process, the segmentation accuracy is further improved, and the semantic information applied in a semantic OctoMap is more accurate.
3. A large number of experiments are conducted to verify the effectiveness of the semantic OctoMap in indoor environments of different scales. We also compare the mapping effect and storage space of the semantic OctoMap with a 3D dense point cloud map and a traditional OctoMap and show the mapping effects of the semantic OctoMap at different resolutions.

2 Related Work

2.1 SLAM

In recent years, with the continuous development of SLAM algorithms, they can be divided into laser based SLAM algorithms and camera based SLAM algorithms according to the types of sensors. The feature point method in camera based visual SLAM is stable and insensitive to light and dynamic objects. Kinect Fusion [7] was the first project using the real-time generation of dense point clouds based on RGB-D cameras and used ICP to calculate the pose between frames. PTAM [8] proposed the parallelization of tracking and mapping, introduced KeyFrames, and used nonlinear optimization for the first time to optimize the camera pose, so it was fast and stable. However, it did not

consider loop closure detection, tracking was easy to lose, and it could only generate sparse point clouds. Mur-Artal proposed ORB-SLAM [9], which is representative of the feature point-based method. After the two-threaded structure of PTAM, ORB-SLAM has a three-threaded structure, including tracking, mapping, and loop closure detection. It can calculate the trajectory of the camera in real-time and generate sparse 3D reconstruction results of the scene. Based on ORB-SLAM, ORB-SLAM2 [10] also supports stereo and RGB-D cameras. ORB-SLAM2 can work well on a standard CPU, regardless of whether it is a handheld device in a small environment, or a UAV and the autonomous driving vehicle in a large environment. The latest ORB-SLAM3 [11] is the first system that can perform visual, visual-inertial, and multimap SLAM with monocular, stereo, and RGB-D cameras using pinhole or fisheye lens models. It is 2 to 5 times better in accuracy than ORB-SLAM2.

2.2 Semantic Segmentation

Traditional semantic mapping methods mainly use SVM, CRF, and other machine learning methods for object detection and segmentation. These methods must establish a local database, which limits the system from detecting objects that are not in the database [12]. With the continuous development of deep learning, object detection and semantic segmentation algorithms based on deep learning provide a new idea for semantic mapping, which can accurately detect more objects.

However, the result of object detection contains both the detected target and the background, so it cannot be segmented accurately. Semantic segmentation can achieve the pixel-level classification of images, that is, each pixel is assigned to its classes. Long et al. proposed using fully convolutional networks [3] for pixel-level prediction to achieve semantic segmentation. UNet [4] adopts an encoder-decoder structure. In the process of feature extraction, the encoder is downsampled continuously. In the decoder stage, by fusing shallow features and deep features, it gradually upsamples, and finally obtains a high-resolution prediction map. In recent years, various methods have been proposed to explore context dependence to obtain more accurate segmentation results. PSPNet [5] improves the FCN and uses a pyramid pooling module to aggregate global context information. When the segmentation layer has more global information, it reduces the probability of false segmentation. However, they capture the context relationships of the same class and ignore the context of a different class. Attention-based methods can also obtain context information, such as channel attention and

spatial attention, and can selectively aggregate context information between different classes. DANet [14] adds two types of attention modules based on an FCN, which simulates semantic interdependence in the spatial dimension and channel dimension and can adaptively integrate local and global features. CCNet [15] proposed a novel criss-cross attention module, which can be used to capture contextual information from long-distance dependencies more efficiently. SANet [16] introduced an attention convolutional channel to realize the attention of pixel groups on the conventional convolution, thus effectively considering the interdependence of the spatial channel. The convolutional block attention module [6] performs dual feature weight calibration from the channel and spatial dimensions, which can be seamlessly integrated into any CNN architecture with negligible extra overhead and can be trained end-to-end together with a CNN. After integrating CBAM into different models, the performance of the models has been improved consistently, showing its wide applicability.

2.3 Semantic Mapping

With the rapid development of SLAM and deep learning, it is possible to build a 3D semantic map for the localization and navigation of intelligent mobile robots. The early method of semantic mapping segmented the reconstructed map into semantic concepts. Pham et al. [17] first reconstructed a dense 3D model using Kinect Fusion [7] and then used a layered conditional random field (CRF) model to assign each 3D point its semantic label. In contrast, our method combines the tracking camera, category detection, and 3D reconstruction. Sünderhauf et al. [18] proposed a CNN-based method for robot location classification and semantic mapping. Scene classification is realized by fusing 2D LIDAR and camera data. It can obtain the category attributes of point clouds, but the space occupied by point clouds is large. SLAM++ [19] focuses on building maps of indoor scenes at the semantic level of objects. However, their method is limited to matching with the targets in the predefined database before establishment. It also does not provide the dense semantic annotation of the whole scene that we are trying to provide in this work. GPSM [20] used Gaussian Processes multi-class classification for map inference and learned the structural and semantic correlation from measurements to infer category labels, to reduce the misclassified labels transmitted to the map. Reference [21] constructed a semantic map composed of point clouds and the background of objects. However, due to the separation of 3D segmentation and object detection, the whole system is complicated

and has many missed detection errors. Semantic Fusion [22] is a dense 3D semantic mapping method using convolutional neural networks. It uses Elastic Fusion [23] as the backend of SLAM to provide pose estimation and combines Bayesian updating and CRF to realize the probability multiplication of predicted values from multiple perspectives. Finally, semantic information on each surfel is fused to generate a dense semantic map based on the surfel. However, similar to point cloud maps, map-based surfels do not contain space occupation information and cannot represent unknown areas, so they cannot be used for navigation.

2.4 OctoMap

Researchers have proposed several methods to build 3D environments, such as point clouds, elevation maps [24], and multilevel surface maps [25]. Unfortunately, point cloud maps have a large scale and store many unnecessary details, so they take up considerable storage space and have low storage efficiency. In addition, they cannot easily distinguish between occupied areas and free areas. Elevation maps and multilevel surface maps cannot represent unknown areas. More importantly, these methods cannot represent an arbitrary 3D environment. OctoMap [2], based on an octree structure, has the advantages of small memory, high storage efficiency, and real-time update. Each voxel node of OctoMap updates voxel occupancy rates from different measurements in a probabilistic manner. Zhang et al. [26] used a handheld RGB-D camera to build voxel based 3D semantic map in real time. Different methods are proposed to integrate semantic information from different perspectives to build a consistent mapping. RS-SLAM [27] used PSPNet for semantic segmentation and the max confidence fusion method to update semantic information in OctoMap. Zhang et al. [28] proposed a sparse outlier elimination algorithm based on K-nearest neighbor and Gaussian, which is used to eliminate sparse outliers of 3D point cloud maps, and then construct a more accurate and compact OctoMap according to the filtered point cloud map. It is essentially an improvement of point clouds. Recurrent-OctoMap [29] proposed a novel semantic mapping method for learning from long-term 3D LIDAR data, which focuses on the 3D refinement of semantic mapping rather than the fusion of 3D semantic mapping based on prediction. To improve the computational efficiency of the framework, Zhang [30] et al. constructed an improved OctoMap based on a fast line rasterization algorithm. In addition, the object detection module and the location module are integrated to obtain the semantic map of the environment. However,

Table 1 Comparison of semantic mapping/OctoMap solutions

	Type	Map	Semantic
Pham et al. [17]	Mapping	Point cloud	CRF model
Sünderhauf et al. [18]	Mapping	Grid-based map	Places205 network
SLAM++ [19]	SLAM	Dense surface reconstruction	Database
GPSM [20]	Mapping	Point cloud	SegNet
Sünderhauf et al. [21]	Mapping	Point cloud	SSD
Semantic Fusion [22]	SLAM	Surfel	Deconvolutional semantic segmentation network
Zhang et al. [26]	SLAM	OctoMap	PSPNet
RS-SLAM [27]	SLAM	OctoMap	PSPNet
Zhang et al. [28]	SLAM	OctoMap	×
Recurrent-OctoMap [29]	Mapping	OctoMap	Model-free objectness detection and a fully connected network
Zhang et al. [30]	SLAM	OctoMap	YOLO
DS-SLAM [31]	SLAM	OctoMap	SegNet
Yue et al. [32]	Mapping	OctoMap	Deeplab

OctoMap's semantic information is at the object level, and OctoMap is still colored for height information. The OctoMap constructed by DS-SLAM [31] and reference [33] was colored according to the semantic information of object category, but the semantic information in the map is sparse, and only the individual interesting objects are given semantic color information. Unlike our work, our semantic OctoMap assigns each voxel the semantic color of its class, and the semantic information is dense in the map. A comparison of semantic mapping and OctoMap solutions is shown in Table 1.

3 Method

To provide an environment model with small storage space and semantic information for a SLAM system, we propose a semantic OctoMap mapping method based on CBAM-PSPNet. Our method changes the way that most SLAM systems use point clouds to represent maps but uses a flexible and compressible OctoMap to model 3D space. Second, we use a semantic segmentation network to obtain high-level semantic information

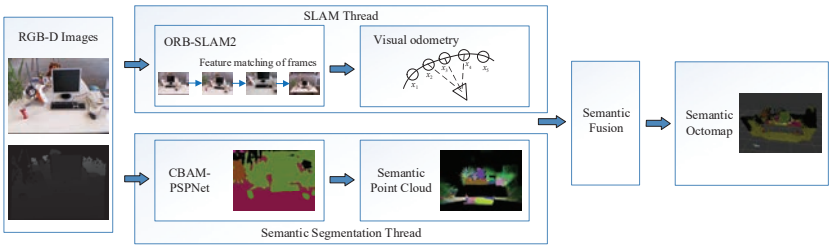


Figure 2 System overview.

in the environment and propose a semantic segmentation network based on CBAM-PSPNet to improve the accuracy of semantic segmentation to give the semantic information to the OctoMap more accurately. Finally, we use a semantic fusion algorithm based on Bayesian fusion to fuse multiview semantic information, and store categories with high probability to solve the problem that it is difficult to update due to a large number of categories. The input of our system is the RGB-D images, and the output is a 3D semantic OctoMap. The structure of the system is shown in Figure 2. First, the RGB-D images acquired by the RGB-D camera are used as input data. Second, RGB-D images are sent to two different threads. ORB-SLAM2 is used as the background thread to obtain the camera pose according to the feature points extracted in each frame. The other thread is responsible for generating a semantic point cloud, the input RGB image is semantically segmented by using CBAM-PSPNet, and the point cloud is generated according to the input depth image and the camera's internal reference matrix. Then, a semantic fusion algorithm based on Bayesian fusion is used to fuse the semantic information from multiple perspectives. Finally, the generated semantic point cloud is inserted into OctoMap.

3.1 SLAM Module

Our semantic mapping system needs to obtain the corresponding knowledge of the global coordinate system from the 2D image, which can be provided by the SLAM system. The SLAM system runs as a background thread and provides the pose of the current camera according to the feature points extracted from each frame. In this paper, we use ORB-SLAM2 [10] as our SLAM system. ORB-SLAM2 is a classic feature-based SLAM that extracts specific feature points from the image to estimate the pose. It has strong anti-interference to illumination and violent movement and is the mainstream method at present. ORB-SLAM2 consists of three parallel threads: tracking,

local mapping, and loop closure detection. The tracking thread is responsible for extracting orb features from the image, estimating the pose according to the previous frame, and deciding when to insert the KeyFrames. The local map thread is responsible for local map construction, including inserting KeyFrames, verifying and filtering newly generated map points, generating new map points and removing redundant KeyFrames and low-quality map points. The loop closure detection thread is responsible for aligning the two sides of the detected loop and optimizing the pose map under the similarity constraint to achieve global consistency and eliminate error accumulation. Although ORB-SLAM2 is a very practical algorithm, it still faces the problem of how to provide semantic information for maps.

The proposed system is a semantic mapping system based on ORB-SLAM2, which can provide semantic maps for robots to perform advanced tasks. As mentioned in Reference [21], our method does not belong to "semantic SLAM". Semantic SLAM requires two directions of information flow: SLAM helps semantics, and semantics helps SLAM. In our work, the SLAM system only provides the camera pose and performs semantic mapping, which means that information only flows in a single direction, so we call our work "semantic mapping".

3.2 Semantic Segmentation Using CBAM-PSPNet

To construct the environment model with semantic information, our method uses a semantic segmentation network to obtain high-level semantic information in the environment. Semantic segmentation can assign a semantic category to each pixel in the input image to obtain pixelated dense classification and assign semantic information to each point in the point cloud. After inserting into OctoMap, each voxel can be assigned a semantic category to obtain a voxelized dense classification.

As a classic semantic segmentation network, PSPNet not only considers real-time performance but also has a better precision of multiclass segmentation. However, the segmentation accuracy of PSPNet is not ideal in the face of small targets, strip parts, and fuzzy edge contours, which will directly lead to poor semantic mapping accuracy of the 3D semantic map. To solve this problem, this paper proposes a semantic segmentation model CBAM-PSPNet, which integrates the convolutional block attention module (CBAM) attention mechanism. CBAM-PSPNet can not only discover subtle features but also effectively highlight key areas, which can give full play to the common advantages of CBAM and PSPNet.

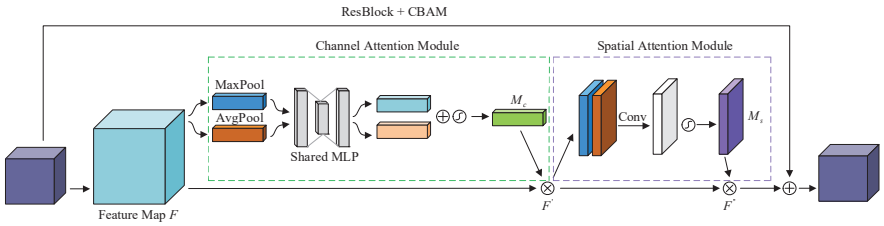


Figure 3 CBAM embedded in ResNet.

In the CBAM-PSPNet model, ResNet50 (Residual Network 50) [34] was used as the basic feature extraction network, and the problems of gradient disappearance, explosion, and network degradation caused by the increase in network layers were solved as much as possible, which could better learn high-level semantic features. Without destroying the original network structure of ResNet, the CBAM attention mechanism is embedded outside each convolution block of ResNet to suppress the influence of useless features on the model. Attention refers to the important spatial information and channel information in the feature channel. It is generally believed that the feature channel pooled by the convolution network has the same importance, but in fact, the importance of the features of each channel is not the same. Therefore, the attention mechanism can identify the key features in the input data through a new layer of weight assignment so that the neural network can learn the feature areas that need to be considered in the input data.

As shown in Figure 3, the network embedded in CBAM first compresses the input feature map F by max-pooling and average-pooling through the channel attention module. Then, the results are input to the multilayer perceptron (MLP) for dimensionality reduction and dimensionality elevation, and the sum of the two output vectors is calculated. Finally, the channel attention weight M_c is generated through the sigmoid function. The weight coefficient is multiplied by the feature map F to obtain the feature map F' after channel weight adjustment. The channel attention mapping process is shown in formula (1).

$$\begin{aligned} M_c(F) &= \sigma(MLP(AvgPool(F)) + MLP(MaxPool(F))) \\ &= \sigma(W_1(W_0(F_{avg}^C)) + W_1(W_0(F_{\max}^C))) \end{aligned} \tag{1}$$

where MLP denotes the multilayer perceptron, σ denotes the sigmoid function, W_0 and W_1 denote the weight matrix in the multilayer perceptron, and F_{avg}^{C} and $F_{\max}^{C}$ denote the average pooling feature and maximum pooling feature in the channel attention module.

Then, the spatial attention module first makes max-pooling and average-pooling for the weighted feature map F' and connects them serially. Then, the convolution is used to reduce its dimension into a single channel feature map. Finally, the sigmoid function is used to generate the spatial attention weight M_s. The weight M_s and the feature map F' are multiplied to obtain the final attention feature map F''. Finally, by adding it to the output of the previous ResBlock, the input of the next ResBlock can be obtained. The process of spatial attention mapping is shown in formula (2).

$$\begin{aligned} M_s(F) &= \sigma(f^{7\times7}[AvgPool(F); MaxPool(F)]) \\ &= \sigma(f^{7\times7}([\mathrm{F}_{avg}^{S}; \mathrm{F}_{\max}^{S}])) \end{aligned} \tag{2}$$

where $f^{7\times7}$ represents a convolution operation with a filter size of 7×7 and F_{avg}^{S} and $F_{\max}^{S}$ denote the average-pooled feature and max-pooled features, respectively, in the spatial attention module.

3.3 Semantic Fusion

Bayesian fusion is widely used in semantic fusion from multiple perspectives. Its principle is to assume that the fusion process is a Markov process. Bayesian fusion is realized by multiplying the semantic label likelihood of a single frame at a pixel and normalizing the product to obtain an effective probability distribution [35]. Given a series of measured and predicted values, Bayesian fusion is often used to aggregate the semantic segmentation of individual views. According to Bayesian rule:

$$p(y|z^i) = \frac{p(z_i|y, z^{i-1})p(y|z^{i-1})}{p(z_i|z^{i-1})} \tag{3}$$

$$= \eta_i p(z_i|y, z^{i-1})p(y|z^{i-1}) \tag{4}$$

where y denotes the semantic labeling of a pixel, z_i denotes its measurement in frame i, and z^i denotes a set of measurements before frame i. If the

measurements satisfy the i.i.d. condition, i.e., $p(z_i|y, z^{i-1}) = p(z_i|y)$ and equal a priori probability for each class, then Equation (3) simplifies to

$$p(y|z^i) = \eta_i p(z_i|y) p(y|z^{i-1}) = \prod_i \eta_i p(z_i|y). \tag{5}$$

In our system, the main disadvantage of Bayesian fusion is that the number of semantic classes cannot be effectively scaled in terms of memory storage efficiency. For example, if Bayesian fusion is performed on the models trained on the ADE20K [36] dataset, then 150 semantic colors and their confidence need to be stored in each voxel of OctoMap. When all nodes are stored, the map occupies much storage space, and the storage efficiency is very low.

To solve this problem, we propose a semantic fusion algorithm based on Bayesian fusion. The main idea of our method is to store only a few categories with high probability and to summarize the remaining categories into one category to address the situation in which the number of classes is too many to update. Specifically, only three semantic colors with the highest confidence are stored in each voxel of OctoMap and named as a semantic set together with their semantic confidence. We classify all the other remaining classes into one class and noted as c_others. The confidence is equal to 1 minus the sum of the three highest semantic confidence, noted as cof_others.

The specific steps of fusing the two semantic sets are as follows: On the one hand, if two semantic sets have the same elements, i.e., the first three semantic colors are the same, then we perform Bayesian fusion for these three classes and c_others directly. On the other hand, if the semantic colors of the two sets are different, new semantic colors are first added to each semantic set so that the semantic colors contained in the two semantic sets are the same. This needs to add new semantic color by splitting the confidence of the class c_others. We set the confidence of each new semantic color to be λ times that of class c_others, and the confidence of class c_others is reduced correspondingly by $1 - \lambda$ times. We set λ to be close to 1 because if a semantic color is not in one set but in another set, it may account for a large proportion of the probability of class c_others. Finally, we only keep the top three semantic colors with the highest confidence in the fusion semantic set. The pseudocode is shown in Algorithm 1.

Algorithm 1 Semantic fusion

Input: Global OctoMap G, current local OctoMap C, node n, node label lab, label confidence cof, punishment value λ, semantic set L

Output: Updated global OctoMap U

for n in C **do**
  if n exists in G **then**
    function SEMANTIC FUSION($L(G_n), L(C_n)$)
    $cof\_others1 = 1 -$ *sum of confidences in* $L(G_n)$
    $cof\_others2 = 1 -$ *sum of confidences in* $L(C_n)$
    if $L(G_n) = L(C_n)$ **then**
      Do nothing
    else
      for lab in $L(G_n)$ not in $L(C_n)$, ordered by $cof_1(lab)$ **do**
        Add lab into $L(C_n)$
        $cof_2(lab) = \lambda * cof\_others2$
        $cof\_others2 = (1 - \lambda) * cof\_others2$
      end for
      for lab in $L(C_n)$ not in $L(G_n)$, ordered by $cof_2(lab)$ **do**
        Add lab into $L(G_n)$
        $cof_1(lab) = \lambda * cof\_others1$
        $cof\_others1 = (1 - \lambda) * cof\_others1$
      end for
    end if
  end if
end for
Update G **with** C **to get the updated global OctoMap** U

3.4 Map Presentation

In our system, we use OctoMap as a 3D map representation. Before inserting OctoMap, the point cloud can be generated according to the input depth image and the camera's internal reference matrix. However, point cloud maps are difficult to apply to intelligent robot navigation or capturing point selection and other advanced complex tasks. This is because the point cloud does not use any data structure to store each point, occupying large storage space and low storage efficiency, which is not conducive to a fast search. An intelligent robot with limited processing speed and memory space will lead to serious time consumption and poor processing performance. Moreover, each point in the point cloud has no volume information, so it is impossible to distinguish between the free areas and the unknown areas, leading to the failure of the intelligent robot in the collision detection task. OctoMap represents space as an octree storage point cloud of voxels, which can distinguish

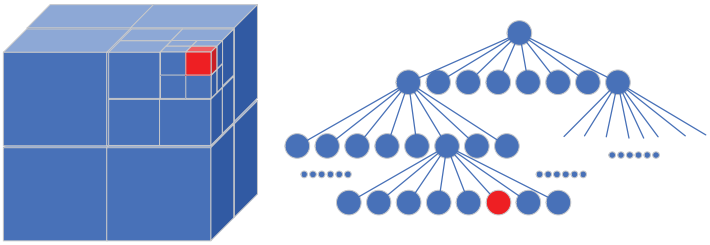


Figure 4 Octree used by OctoMap.

between unknown areas and free areas. Each node in the octree represents a voxel of a specific size, depending on its level in the tree. Each parent node of the octree is subdivided into eight subnodes until the most refined resolution is achieved. The structure of the octree is shown in Figure 4.

Before inserting OctoMap, structural information is stored in the form of a point cloud for information transmission. A point cloud is a group of unordered points, each of which contains its coordinates in a specific reference system. First, the depth image is registered to the reference coordinate system of the RGB image. This can be done with a camera. Then, according to the position and depth of each pixel in the image and the internal reference of the camera, the real-world coordinates of each pixel are calculated to generate the point cloud. In the pinhole camera model, given the pixel provided by the RGB image and its coordinate (u, v) and depth d, its coordinate in the camera coordinate system is $P = (X, Y, Z)$. The coordinates of the obstacle points in the camera coordinate system are the point cloud data. The relation formula of coordinates of points in camera coordinate system P and pixel coordinate system P_{uv} is as follows:

$$ZP_{uv} = Z\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = KP \tag{6}$$

By consolidating the above formula, we obtain

$$\begin{cases} X = Z(u - c_x)/f_x \\ Y = Z(v - c_y)/f_y \\ Z = d \end{cases} \tag{7}$$

where f_x and f_y denote the camera focal length, (c_x, c_y) denotes the center of the camera in the pixel coordinates of the image, and K denotes the camera internal reference matrix.

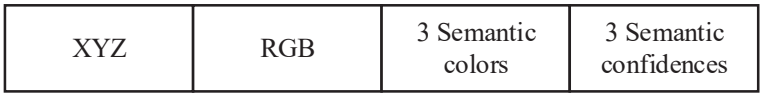


Figure 5 Illustration of simplified point structure.

In addition to location information and RGB information, semantic information obtained from the semantic segmentation network of CBAM-PSPNet is also stored in the point cloud, which contains three semantic colors and their confidence. The simplified structure of points in the point cloud is shown in Figure 5.

OctoMap is a storage representation of the environment that divides the environment into 8-dimensional subspaces. The 3D measurement unit in the environment is represented by voxels, which include occupied, free and unknown states. In the actual SLAM system implementation, camera motion and ranging error will generate considerable noise in the map. To reduce this effect, OctoMap uses a probabilistic model of marker voxels to solve this problem. Each leaf node in OctoMap stores the probability that it is occupied or free. Given the sensor measurement $z_{1:t}$, the occupancy probability $P(n|z_{1:t})$ of the leaf node n is calculated according to the following formula:

$$P(n|z_{1:t}) = \left[1 + \frac{1 - P(n|z_t)}{P(n|z_t)} \frac{1 - P(n|z_{1:t-1})}{P(n|z_{1:t-1})} \frac{P(n)}{1 - P(n)}\right]^{-1} \tag{8}$$

where n denotes the leaf node, z_t denotes the current measurement, $P(n)$ denotes the prior probability, $P(n|z_{1:t-1})$ denotes the previous estimates, and $P(n|z_t)$ denotes the occupied probability of voxel n for a given measurement z_t.

In general, a priori probability $P(n) = 0.5$ is assumed to represent the undefined state. Before the camera moves, all nodes will be marked as unknown. As the camera starts to perceive the environment, the confidence changes. The probabilistic octree uses the following formula to update log odds value $L(n|z_{1:t})$ in time sequence 1,2…,t:

$$L(n|z_{1:t}) = L(n|z_{1:t-1}) + L(n|z_t) \tag{9}$$

with

$$L(n) = \log\left[\frac{P(n)}{1 - P(n)}\right] \tag{10}$$

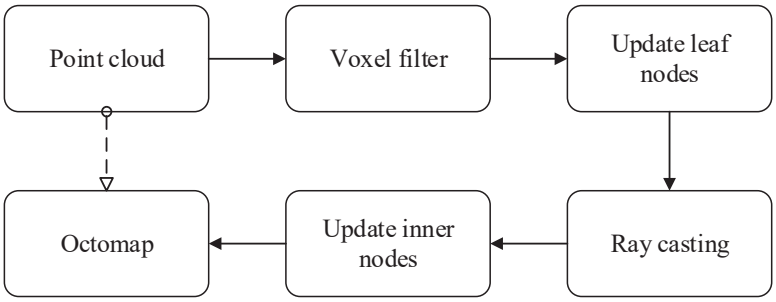


Figure 6 Flowchart of inserting point cloud into OctoMap.

Although the traditional OctoMap can update the occupancy rate of voxels by a probability model, it cannot represent and update the semantic color and confidence of voxels. Our method adds the semantic information obtained by semantic segmentation to the point cloud and inserts the semantic point cloud into OctoMap so that OctoMap can obtain higher-level semantic information.

The point cloud is inserted into OctoMap, and the flow chart is shown in Figure 6. First, the voxel filter is used to subsample the points, and only one point is reserved in the given voxel space. We only need one point to update one octree node. Second, the highest resolution leaf node voxel is updated. Their occupancy, RGB color, semantic color, and confidence are updated. The occupancy rate of voxels is updated by the probability model. RGB color is the average of the previous color. The semantic color and confidence are updated according to the improved semantic fusion method based on Bayesian fusion. Then, ray casting is performed to clear the free areas along the line between the origin and the endpoint. Finally, the internal nodes of OctoMap are updated to obtain information on voxels with lower resolution. We set the occupancy rate of the parent node to the maximum of its eight children nodes, the color of the parent node to the average of its children nodes, and the semantic information of the parent node is the semantic fusion of the child nodes.

In a traditional OctoMap, if the occupancy rate of all the children nodes is the same, the children nodes will be pruned. In contrast, the proposed semantic OctoMap leaf node retains semantic information, so only when the occupancy, semantic color and semantic confidence of all the children nodes are the same, will its children nodes be pruned, which will lead to an increase in map storage space.

4 Experiments

4.1 Semantic Segmentation

To verify the improved effect of the CBAM-PSPNet model, comparative experiments are designed to analyze the segmentation performance of CBAM-PSPNet, PSPNet, and FCN. The segmentation accuracy of the CBAM-PSPNet model is verified using the ADE20K [36] dataset. The ADE20K dataset contains 20210 training sets and 2000 verification sets. The advantage of the ADE20K dataset is that it has rich scenes, including indoor, outdoor, and natural scenes. At the same time, it has a large number of semantic classes, a total of 150, which means that our semantic mapping system can more accurately identify objects in the scene and even distinguish chairs and swivel chairs.

This experiment is based on an Intel Xeon E5-2603 V3, NVIDIA GeForce GTX 1080 11 GB, and the deep learning framework PyTorch 1.0. The basic network used in the model training is ResNet50, and the initial learning rate of the network is set as 0.0001. Influenced by GPU memory, the sample needs to be input into the network in batches during network training. The batch size is set to 4, which determines the number of training images. The epoch value is set to 50, which is the number of iterations of the entire training set. The loss function used in the training test is the cross-entropy function.

In this experiment, MPA and MIoU are used as evaluation indexes to evaluate the segmentation effect of the model on the ADE20K dataset. MPA calculates the proportion of the number of correctly classified pixels of each class and then accumulates the average value. MIoU is the ratio of the intersection and union between the predicted results of each category and the real value of the model, and the result of summation and averaging. They are calculated as follows:

$$MPA = \frac{1}{k+1} \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \tag{11}$$

$$MIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{12}$$

where $k+1$ represents the total number of categories; p_{ij} denotes that the actual category is class i, but the predicted result is the number of pixels of class j.

The comparison of segmentation accuracy based on the ADE20K dataset is shown in Table 2, and the comparison of segmentation effect is shown in

Table 2 Comparison of segmentation accuracy of ADE20K dataset

Model	MPA	MIoU
FCN	69.83	26.23
PSPNet	71.70	30.69
CBAM-PSPNet	**74.25**	**32.57**

(a) Image (b) Ground Truth (c) FCN (d) PSPNet (e) CBAM-PSPNet

Figure 7 Comparison of segmentation effect of ADE20K verification set.

Figure 7. Table 2 shows that the MPA and MIoU index values of the CBAM-PSPNet model proposed in this paper are higher than those of FCN and PSPNet on the ADE20K dataset. MPA increased by 4.42% and 2.55% respectively, and MIoU increased by 6.34% and 1.88% respectively. As shown in Figure 7, the segmentation effect of the CBAM-PSPNet model on small object edges is better than that of the FCN and PSPNet models, such as the segmentation of pillows on beds, edge contours of beds, and pendant lamps.

This is because the FCN model uses the convolutional network to perform many times continuous downsamplings in the encoding process, which makes the resolution of features continue to decline, and the image loses many shallow edge contours and other details. In the decoding process, transpose convolution is used to directly restore the feature map to the original image size, which makes the details of restoration too rough. The PSPNet model integrates global and local features by using a pyramid pooling module on

a high-level feature map, which overcomes the defect that pooling can only obtain the feature information of a fixed window to a certain extent. However, in the continuous downsampling process, too much shallow information is still lost, resulting in the segmentation of the target edge not being fine enough. Our CBAM-PSPNet model preserves the shallow information of the small target at the edge of the object in the encoding stage and feature fusion stage, which further improves the segmentation accuracy.

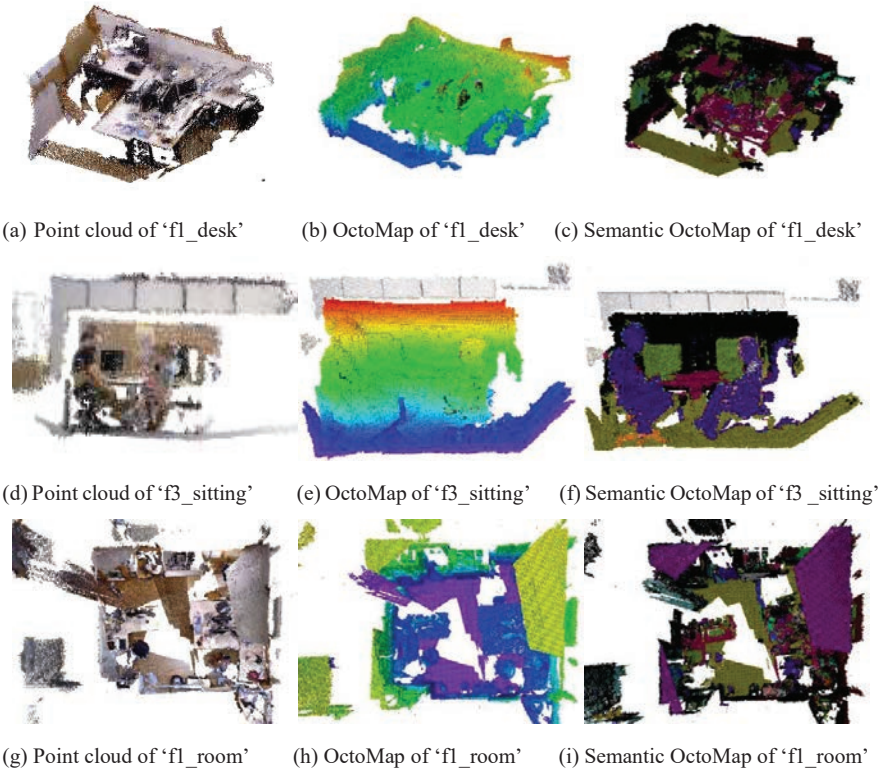
4.2 Semantic Map

We demonstrate and evaluate the functions of our semantic OctoMap in different indoor environments by using the TUM dataset. The experiments were validated in several sequences in the TUM dataset, from the office environment containing the desk, the environment where two people sat at the desk to a large and complete office room. We complete the mapping of each environment and compare our semantic OctoMap with a 3D dense point cloud map and a traditional OctoMap.

As shown in Figure 8, from left to right are the dense point cloud maps of 'freiburg1_desk', 'freiburg3_sitting_static', and 'freiburg1_room' sequences in the TUM dataset, the traditional OctoMap, and our proposed semantic OctoMap. Figures 8(a), (d), (g) are dense point cloud maps. According to the estimated pose of the camera, RGB-D data are transformed into point clouds and then stitched together to obtain a point cloud map composed of discrete points. When there is no requirement for the appearance or obstacle information of the map, the construction of a point cloud map is the simplest and most intuitive method. However, the point cloud map is only a group of discrete points, which only contains the basic position coordinates and RGB color information and cannot distinguish the occupied areas and free areas in the environment, so it cannot be used for robot navigation tasks.

Figures 8(b), (e), and (h) show traditional OctoMap images that can be converted directly from point cloud maps and visualized by octovis tools. The traditional OctoMap stores the probability information of nodes occupied or free in areas and gives the color information from deep to light according to the height from low to high. We can judge whether the mobile robot can pass or not according to the information of the occupancy probability of voxels to realize the navigation task. The altitude information can be used as a reference for the flying area of the UAV.

Figures 8(c), (f), and (i) show the semantic OctoMap proposed by us. During the operation of the system, the SLAM module estimates the pose of



(a) Point cloud of 'f1_desk' (b) OctoMap of 'f1_desk' (c) Semantic OctoMap of 'f1_desk'

(d) Point cloud of 'f3_sitting' (e) OctoMap of 'f3_sitting' (f) Semantic OctoMap of 'f3 _sitting'

(g) Point cloud of 'f1_room' (h) OctoMap of 'f1_room' (i) Semantic OctoMap of 'f1_room'

Figure 8 Comparison of mapping effects.

the camera in real time. At the same time, CBAM-PSPNet semantic segmentation network obtains the location and semantic category labels of objects in the environment. Then, a semantic fusion algorithm based on Bayesian fusion is used to fuse the semantic information from multiple perspectives. Finally, the semantic OctoMap is generated. Each voxel in our proposed semantic OctoMap is dyed according to its semantic information, which shows that the system can accurately identify indoor objects. For example, the floor, desk, computer monitor, people, ceiling, and wall are given semantic color information such as yellow, red, green, purple, pink, and black, respectively.

We also performed a comparative analysis of the storage space occupied by the above maps, as shown in Table 3. As seen in Table 3, the storage space occupied by the dense point cloud map is much larger than that of the traditional OctoMap and the semantic OctoMap because both the traditional OctoMap and the semantic OctoMap adopt the octree data structure.

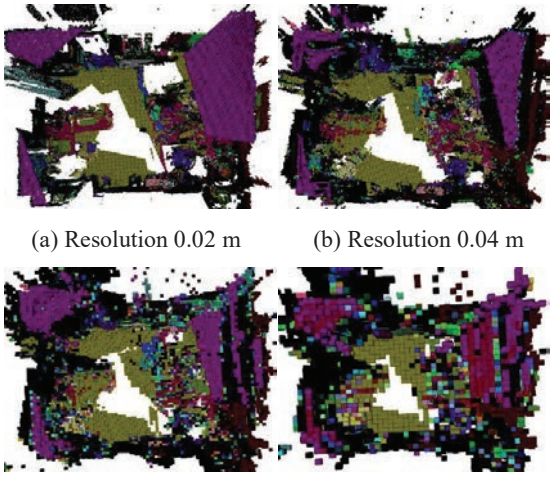
Table 3 Comparison of map storage space

Sequence	Dense Point Cloud Map	Traditional OctoMap	Semantic OctoMap
freiburg1_desk	3.3 MB	16.0 KB	240.0 KB
freiburg3_sitting_static	1.5 MB	32.0 KB	256.0 KB
freiburg1_room	14.3 MB	80.0 KB	1.1 MB

Although the point cloud has limited the one-to-one relationship between voxels and points by a voxel filter, the information of whether it is occupied is stored in voxels. When all the children nodes of a node are occupied or not occupied, it is unnecessary to expand the node. For example, when the map that the system just started to build is blank, it only needs a root node instead of a complete tree. When gradually adding information to the map, the actual objects in the real environment are often connected, and the blank places are often connected, so most octree nodes do not need to expand to the leaf level. The semantic OctoMap takes up slightly more storage space than the traditional OctoMap because semantic OctoMap not only contains occupancy information but also gives semantic color and semantic confidence to each leaf node. The pruning of the children nodes can only be done if all the children nodes have the same occupancy, semantic color, and semantic confidence. Although the storage space of the semantic OctoMap has been improved slightly, robots can navigate with high-level semantic information.

The proposed semantic OctoMap can use the internal nodes of OctoMap to change its resolution to adapt to different scenes. The principle is to update the occupancy rate, RGB color, semantic color, and semantic confidence of internal nodes of OctoMap to obtain the information of lower resolution voxels. As shown in Figure 9, (a)–(d) correspond to map resolutions of 0.02 m, 0.04 m, 0.08 m and 0.16 m respectively. We find that when the resolution is 0.02 m, the mapping effect is very fine, while when the resolution is 0.16 m, the mapping effect is slightly rough. We not only obtain global semantic information of a larger range of 3D scenes by increasing the resolution, but also obtain the semantic map of a smaller range of indoor scenes by reducing the resolution, with less noise.

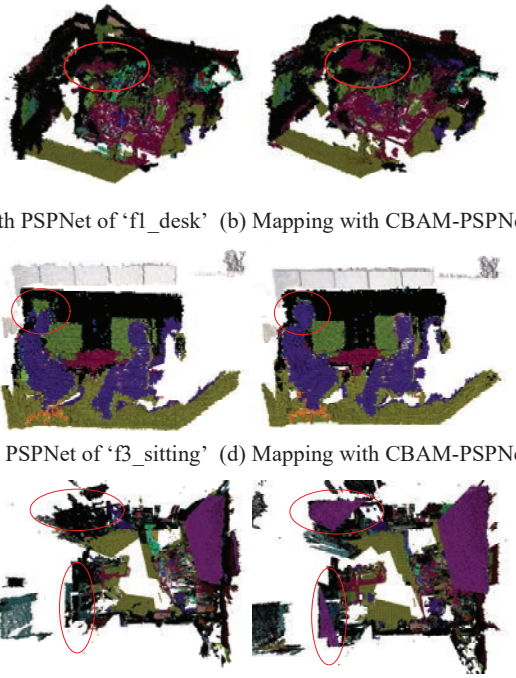
To verify that the accuracy of the semantic segmentation network will directly affect the accuracy of the semantic information of the map, we design a comparative experiment to qualitatively analyze the effects of PSPNet and CBAM-PSPNet models in the semantic OctoMap. As shown in Figure 10, (a), (c), (e) and (b), (d), (f) are the semantic OctoMap mapping effects of using PSPNet and CBAM-PSPNet in the sequence ‘freiburg1_desk’,



(a) Resolution 0.02 m (b) Resolution 0.04 m

(c) Resolution 0.08 m (d) Resolution 0.16 m

Figure 9 Comparison of the map resolution effect.



(a) Mapping with PSPNet of 'f1_desk' (b) Mapping with CBAM-PSPNet of 'f1_desk'

(c) Mapping with PSPNet of 'f3_sitting' (d) Mapping with CBAM-PSPNet of 'f3_sitting'

(e) Mapping with PSPNet of 'f1_room' (f) Mapping with CBAM-PSPNet of 'f1_room'

Figure 10 Comparison of mapping effects using PSPNet and CBAM-PSPNet models.

'freiburg3_sitting_static', and 'freiburg1_room' of the TUM dataset. Because the segmentation effect of the CBAM-PSPNet model is better than that of the PSPNet model in object edge small target segmentation, the CBAM-PSPNet model is more precise than the PSPNet model in semantic OctoMap mapping for table edges, edges of human heads and ceilings.

5 Discussion

As mentioned in the introduction, the robot's perception and modeling of the environment has been widely studied in the SLAM system, to provide a world model that transmits geometric information and semantic information simultaneously in the process of robot navigation. However, the point cloud generated by the SLAM system has some problems, such as large storage space, lack of semantic information, and cannot be used by some navigation tasks because of lacking the information of whether the area is occupied. In order to address the above problems, we choose OctoMap as an alternative in our research. In addition, we try to build a semantic segmentation model by introducing an attention mechanism to improve the accuracy of semantic information in semantic OctoMap.

In reference [30], object detection module YOLO [13] is used to convert detected objects into OctoMap. However, object detection can only recognize the object of interest in images, which can express only a small part of the semantic information. Our model uses semantic segmentation to classify images at the pixel level and predict the class to which each pixel belongs. Therefore, applying the semantic segmentation model to OctoMap can contain more semantic information. For example, the wall or ceiling in the semantic OctoMap can be represented by black and pink. In addition, their OctoMap is dyed according to the height information, while we dye according to the semantic information. Although the OctoMap constructed by DS-SLAM [31] and reference [33] is dyed according to the semantic information of object categories, the semantic information in the map is sparse because of the limits of the datasets used in training the semantic segmentation network. For instance, PASCAL VOC dataset [37] used in DS-SLAM [31] contains only 21 classes, and the MS COCO dataset [38] used in the reference [33] contains 80 classes. Different from the 2 datasets, the ADE20K dataset [36] that we used contains 150 classes, consequently, it can provide each voxel with the class, and make the semantic information dense in the map. References [26, 27] uses semantic segmentation network PSPNet to construct semantic OctoMap, but there is still some space for improving

the accuracy of semantic segmentation. Inspired by some studies on attention mechanisms, we introduce it to the semantic segmentation model and try to capture subtle features to improve the semantic understandings of maps.

Meanwhile, our research also has some limitations. For example, we do not consider the interference of dynamic objects in the environment while building semantic OctoMap as in references [27, 31]. In addition to the idea of improving the semantic understanding of a framework in semantic OctoMap, 3-D refinement of semantic maps (i.e. fusing semantic observations) [29] and semantic OctoMap integrating multi-robot systems [32] both provide new ideas on how to build semantic OctoMap.

6 Conclusion

This paper proposes a semantic OctoMap mapping method based on CBAM-PSPNet. Our method changes the map representation of point clouds in most SLAM systems and uses a flexible and compressible OctoMap to model 3D space. Our proposed semantic OctoMap uses ORB-SLAM2 to obtain camera pose. The CBAM-PSPNet model is used to obtain high-level semantic information in the environment, to improve the accuracy of 3D semantic annotation. A semantic fusion algorithm based on Bayesian fusion is used to fuse the semantic information from multiple perspectives. The efficient storage, compressibility, and semantic fusion effectiveness of our proposed semantic OctoMap are verified on the TUM dataset.

Acknowledgment

This work was supported by National Key R&D Program of China (No. 2020YFB1005900), National Natural Science Foundation of China (Nos. 61773027 and 62076014), and Industrial Internet Innovation and Development Project (No. 135060009002).

References

[1] Cadena C, Carlone L, Carrillo H, et al. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age[J]. IEEE Transactions on robotics, 2016, 32(6): 1309–1332.

[2] Hornung A, Wurm K M, Bennewitz M, et al. OctoMap: An efficient probabilistic 3D mapping framework based on octrees[J]. Autonomous robots, 2013, 34(3): 189–206.

[3] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431–3440.
[4] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015: 234–241.
[5] Zhao H, Shi J, Qi X, et al. Pyramid scene parsing network[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2881–2890.
[6] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3–19.
[7] Izadi S, Kim D, Hilliges O, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera[C]//Proceedings of the 24th annual ACM symposium on User interface software and technology. 2011: 559–568.
[8] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces[C]//2007 6th IEEE and ACM international symposium on mixed and augmented reality. IEEE, 2007: 225–234.
[9] Mur-Artal R, Montiel J M M, Tardos J D. ORB-SLAM: a versatile and accurate monocular SLAM system[J]. IEEE transactions on robotics, 2015, 31(5): 1147–1163.
[10] Mur-Artal R, Tardós J D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras[J]. IEEE transactions on robotics, 2017, 33(5): 1255–1262.
[11] Campos C, Elvira R, Rodríguez J J G, et al. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM[J]. IEEE Transactions on Robotics, 2021.
[12] Reddy N D, Singhal P, Krishna K M. Semantic motion segmentation using dense CRF formulation[C]//Proceedings of the 2014 Indian conference on computer vision graphics and image processing. 2014: 1–8.
[13] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779–788.
[14] Xue H, Liu C, Wan F, et al. Danet: Divergent activation for weakly supervised object localization[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 6589–6598.

[15] Huang Z, Wang X, Huang L, et al. Ccnet: Criss-cross attention for semantic segmentation[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 603–612.
[16] Fan H, Ling H. Sanet: Structure-aware network for visual tracking[C]//Proceedings of the IEEE conference on computer vision and pattern recognition workshops. 2017: 42–49.
[17] Pham T T, Reid I, Latif Y, et al. Hierarchical higher-order regression forest fields: An application to 3d indoor scene labelling[C]//Proceedings of the IEEE international conference on computer vision. 2015: 2246–2254.
[18] Sünderhauf N, Dayoub F, McMahon S, et al. Place categorization and semantic mapping on a mobile robot[C]//2016 IEEE international conference on robotics and automation (ICRA). IEEE, 2016: 5729–5736.
[19] Salas-Moreno R F, Newcombe R A, Strasdat H, et al. Slam++: Simultaneous localisation and mapping at the level of objects[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2013: 1352–1359.
[20] Jadidi M G, Gan L, Parkison S A, et al. Gaussian processes semantic map representation[J]. arXiv preprint arXiv:1707.01532, 2017.
[21] Sünderhauf N, Pham T T, Latif Y, et al. Meaningful maps with object-oriented semantic mapping[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 5079–5085.
[22] McCormac J, Handa A, Davison A, et al. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks[C]//2017 IEEE International Conference on Robotics and automation (ICRA). IEEE, 2017: 4628–4635.
[23] Whelan T, Salas-Moreno R F, Glocker B, et al. ElasticFusion: Real-time dense SLAM and light source estimation[J]. The International Journal of Robotics Research, 2016, 35(14): 1697–1716.
[24] Kweon I S, Hebert M, Krotkov E, et al. Terrain mapping for a roving planetary explorer[C]//IEEE International Conference on Robotics and Automation. IEEE, 1989: 997–1002.
[25] Triebel R, Pfaff P, Burgard W. Multi-level surface maps for outdoor terrain mapping and loop closing[C]//2006 IEEE/RSJ international conference on intelligent robots and systems. IEEE, 2006: 2276–2282.
[26] Xuan Z, David F. Real-Time Voxel Based 3D Semantic Mapping with a Hand Held RGB-D Camera. 2018[J]. GitHub, GitHub repository, https://github.com/floatlazer/semantic_slam.

[27] Ran T, Yuan L, Zhang J, et al. RS-SLAM: A Robust Semantic SLAM in Dynamic Environments Based on RGB-D Sensor[J]. IEEE Sensors Journal, 2021, 21(18): 20657–20664.

[28] Zhang J, Liu S, Gao B, et al. An improvement algorithm for OctoMap based on RGB-D SLAM[C]//2018 Chinese Control And Decision Conference (CCDC). IEEE, 2018: 5006–5011.

[29] Sun L, Yan Z, Zaganidis A, et al. Recurrent-octomap: Learning state-based map refinement for long-term semantic mapping with 3-d-lidar data[J]. IEEE Robotics and Automation Letters, 2018, 3(4): 3749–3756.

[30] Zhang L, Wei L, Shen P, et al. Semantic SLAM based on object detection and improved octomap[J]. IEEE Access, 2018, 6: 75545–75559.

[31] Yu C, Liu Z, Liu X J, et al. DS-SLAM: A semantic visual SLAM towards dynamic environments[C]//2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018: 1168–1174.

[32] Yue Y, Li R, Zhao C, et al. Probabilistic 3d semantic map fusion based on bayesian rule[C]//2019 IEEE International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM). IEEE, 2019: 542–547.

[33] Liu K, Fan Z, Liu M, et al. Object-aware Semantic Mapping of Indoor Scenes using Octomap[C]//2019 Chinese Control Conference (CCC). IEEE, 2019: 8671–8676.

[34] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770–778.

[35] Ma L, Stückler J, Kerl C, et al. Multi-view deep learning for consistent semantic mapping with rgb-d cameras[C]//2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017: 598–605.

[36] Zhou B, Zhao H, Puig X, et al. Semantic understanding of scenes through the ade20k dataset[J]. International Journal of Computer Vision, 2019, 127(3): 302–321.

[37] Everingham M, Van Gool L, Williams C K I, et al. The pascal visual object classes (voc) challenge[J]. International journal of computer vision, 2010, 88(2): 303–338.

[38] Lin T Y, Maire M, Belongie S, et al. Microsoft coco: Common objects in context[C]//European conference on computer vision. Springer, Cham, 2014: 740–755.

Biographies

Xiaogang Ruan received the Ph.D. degree in control science and engineering from Zhejiang University, China in 1992. Now he is a professor of Beijing University of Technology, and he is also as a director of Institute of Artificial Intelligent and Robots (IAIR). His research interests include automatic control, artificial intelligence, and intelligent robot.

Peiyuan Guo received the B.E. degree in building electricity and intelligence from Qingdao University of Technology, China in 2019. He is currently a master student in control science and engineering at Faculty of Information Technology of Beijing University of Technology, China. His research interest is SLAM.

Jing Huang received the Ph.D. degree in pattern recognition and intelligent system from Beijing University of Technology, China in 2016. Now she is an associate professor in Faculty of Information Technology, Beijing University of Technology, China. Her research interests include cognitive robotics, machine learning, and artificial Intelligence.