Proceeding of the IEEE
International Conference on Robotics and Biomimetics
Dali, China, December 2019

Object-Aware Hybrid Map for Indoor Robot Visual Semantic Navigation*

Li Wang, Ruifeng Li, Jingwen Sun, and Lijun Zhao
State Key Laboratory of Robotics and System
Harbin Institute of Technology
Harbin 150001, China
{15b908017, lrf100 & zhaolj}@hit.edu.cn,
18S008061@stu.hit.edu.cn

Hezi Shi, Hock Soon Seah, and Budianto Tandianus
School of Computer Science and Engineering
Nanyang Technological University
Nanyang Avenue 639798, Singapore
{ashsseah & btandianus}@ntu.edu.sg,
hezi001@e.ntu.edu.sg

Abstract – In order to achieve an intuitive interaction and visual semantic navigation for the indoor robot, we propose a novel object-aware hybrid map. The existing map is usually a metric map, lacking semantics for interaction. We combine objects in the indoor environment with the metric map to constitute a hybrid map. The map consists of a 3D object semantic map and a 2D occupancy grid map, which transfers human commands to the grid map through object semantics, thereby enabling autonomous navigation for the robot. We utilize ORB-SLAM2 for continuous pose estimation and 3D mapping. 2D object detection in keyframes is conducted based on YOLO v3. The object point clouds in multiple perspectives are merged and a 3D bounding box of the object is estimated. These objects construct a 3D semantic map. Furthermore, we project a 3D point cloud map into a 2D plane in order to get an occupancy grid map. Finally, these two maps are combined forming an object-aware hybrid map. We conduct experiments in real environments in order to verify the feasibility and robustness of the hybrid map for robot semantic navigation.

Index Terms – Hybrid map, 3D object detection, robot semantic navigation

I. INTRODUCTION

In a typical office or home environment, some of possible tasks for autonomous service robots are delivering files to a place, bringing a cup to a table, and so on. When people interact with a robot, the most natural way is to give the robot a specific task directly through a voice command and the robot will perform the instructed command. This process mainly includes speech recognition, navigation, and task operations. The robot analyzes the voice command and converts it to a navigation goal. Then, the specific operation is performed at a designated location. However, there is still a gap between voice command and robot operation. The main reason is that it is difficult to combine semantic information from voice command with a metric map that is required for robot navigation.

With the rapid development of visual technology and deep learning, many visual-based 3D mapping, object recognition, and scene recognition methods have appeared. In the 3D mapping area, methods based on sparse feature points for continuous pose estimation are proposed, such as PTAM [1], ORB-SLAM2[2], SVO [3], and so on. These algorithms focus on the accuracy of pose estimation and they can realize mapping on a large scale. However, these methods generate 3D metric maps that do not have semantic information in the environment and they cannot be directly used for robot semantic navigation. In recent years, remarkable progress has been made in object recognition methods based on deep learning. Many algorithms with high detection accuracy and efficiency have appeared, such as Faster R-CNN [4], SSD [5], YOLO [6] and so on. These algorithms can identify the category of an object and utilize a 2D bounding box to determine the position of an object in the image. Some work combines object recognition with 3D mapping that results in semantic information of objects embedded in a 3D point cloud. But the work usually does not give an indication how these maps can be used for semantic robot navigation.

As indoor objects are principal components in an indoor environment, they constitute the main semantic content, such as tables, chairs, sofas, and cups. People often express semantic commands using high level description of objects, for example, “bring a cup on the table”. In order to accomplish the task, it is necessary to combine the semantic information with the metric map. To this end, we propose an object-based hybrid map for robot semantic navigation. The hybrid map consists of an object map and a 2D grid map.

We combine a 3D geometric map with an object recognition method. In this part, keyframes generated by ORB-SLAM2 are detected and their object point clouds are extracted. Then, the point clouds of the same object from different perspectives are automatically fused to obtain a more complete 3D object and its 3D bounding box is estimated based on the Manhattan frame. Finally, the object-aware map that contains the 3D sizes and locations in the environment is built.

* This work is partially supported by the National Key Research and Development Program “Intelligent Robot” Key Special Project [2018YFB1307100]; National Natural Science Foundation of China [61673136]; Self-Planned Task of State Key Laboratory of Robotics and System (HIT) [NO. SKLRS201906B, NO. SKLRS201715A]; the Foundation for Innovative Research Groups of the National Natural Science Foundation of China [No.51521003]; Centre for Augmented and Virtual Reality of Nanyang Technological University, and the China Scholarship Council [201706120137].

978-1-7281-6321-5/19//$31.00 ©2019 IEEE

Since the point cloud cannot be directly used for navigation, the 3D point cloud map is converted into a 3D grid map. Then, it is projected onto the 2D plane in order to obtain a 2D occupancy grid map for robot navigation. Finally, a hybrid map containing the 3D object-aware map and a 2D occupancy grid map is built. By constructing the hybrid map, semantic commands are mapped to geometric locations in the map, thereby enabling the robot navigation.

The main contributions of this paper are:

(1) A proposed object-aware hybrid map that fuses objects with metric maps. The map can be directly used for semantic navigation of robots;

(2) A proposed 3D object detection method fusing multiple view angles, which can effectively detect objects and estimate a 3D bounding box of the object.

The paper is organized as follows. Section II introduces related work on object detection and robot semantic navigation. Section III describes the method for building an object-aware hybrid map. Section IV shows our experimental details. Conclusions and some suggestions for future work are given in Section V.

II. Related Work

Autonomous robot operation in indoor environments is an important requirement nowadays. The usual solution is that the robot establishes an environmental map and navigates using it to the given goal coordinate. However, it is less convenient compared to being able to command the robot by using voice command. Therefore, a lot of effort is devoted to improving semantic information of maps.

Computer vision and deep learning techniques encourage the development of indoor semantic perception for robots. Some work focuses on dense semantic mapping for adding labels to the map constructed by visual SLAM. Hermans *et al.* [7] develop a method that generates a 3D semantic mapping from RGB-D images. The 2D semantic segmentation based on randomized decision forests is used to create a consistent 3D semantic mapping of indoor scenes. McCormac *et al.* [8] propose the SemanticFusion method for dense 3D semantic mapping by using convolutional neural networks (CNNs). They combine the CNNs with ElasticFusion [9] which is one of the state-of-the-art dense visual SLAM methods for generating a point cloud with semantic labels. Qi *et al.* [10] propose a PointNet network for 3D classification and segmentation on point clouds using a neural network. The network uses point clouds as direct inputs and outputs and the unordered problem of point clouds is well solved in the network. However, dense point clouds with semantic labels consume a large amount of storage space. Moreover, the aforementioned work does not consider how to use a semantic map for robot navigation.

Some other work concerns more on object representation as indoor semantic information. For indoor scenes, objects are usually the main semantic components and they can be used to express the environment. Therefore, by using objects, we effectively describe an indoor scene at a higher level than point clouds. A SLAM system named SLAM++ is developed by Salas-Moreno *et al.* [11] to express indoor scenes at the semantic object level. However, the system is limited to recognize objects due to using a pre-defined database. In the work by Grinvald et al. [12], the use of higher-level entities, such as object instances, improves navigation and planning interactions in a real environment for robots. A volumetric object-centric map using an RGB-D camera is proposed. However, there is no description of how to use it for robots. Recently, 3D object detection based on deep learning has made great progress. Many new neural networks, such as DSS [13], Amodal 3D [14], 2D-Driven method [15], Frustum PointNet++ [16], are developed to regress 3D bounding box (including size, position and orientation) and object label. These methods are usually trained on the datasets which include a large number of well-labeled data, such as NYU V2 [17], SUN RGB-D [18], SceneNN [19]. Although these methods can conduct 3D object detection directly, the disadvantage is that the operation requirement is quite high and time-consuming. Therefore, it is currently difficult to deploy on robots.

Most works on semantic mapping and object detection pay more attention on segmentation accuracy of object point clouds, not the application for robots. What format should be used for a robot map containing object semantics? How can an object-based map be utilized by the robot for semantic navigation? Many problems in this area still exist, and little work provides solutions for robot semantic navigation based on objects. Therefore, we explore the issue in this paper and propose a new kind of map, namely the object-aware hybrid map, for semantic robot navigation.

III. Method

For the purpose of indoor visual semantic robot navigation, we propose an object-aware hybrid map built from semantic order of users and the metric map that is utilized by a robot for navigation. The hybrid map consists of a 3D object-aware map and a 2D occupancy grid metric map. We use an RGB-D camera to generate the hybrid map automatically. The whole framework of our algorithm is mainly composed of three modules, namely 3D object-aware semantic map module, 2D occupancy grid map module, and object-aware hybrid map module, as shown in Fig. 1.

A. 3D object-aware semantic map based on multi-view fusion

As objects are regarded as vital semantic information for describing a common indoor environment, we use objects to establish a semantic map. Several object properties are used, such as category, 3D size, and central point. Moreover, we utilize a 3D bounding box to represent the spatial position of the object. Therefore, the main problem is to conduct 3D object detection in the environment.

In this paper, we use only one RGB-D camera mounted on a robot to perform 3D object detection. It always encounters incomplete observation of objects due to the limited perspective of using merely a single camera. As the ro-

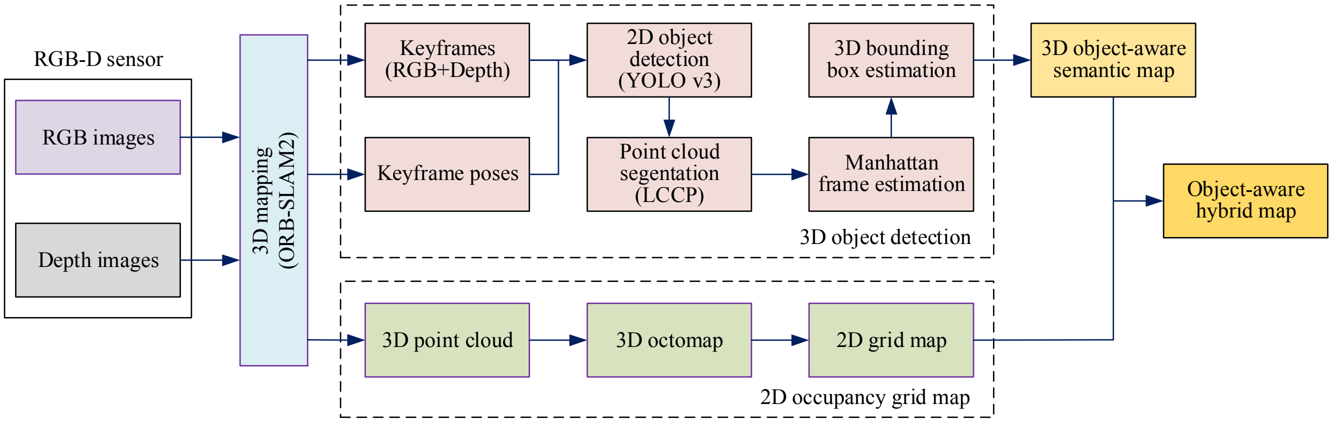


Fig. 1 The framework of the object-aware hybrid map.

bot moves freely, the multi-view observation can be obtained. We employ visual SLAM to estimate keyframe poses and fuse them in order to conduct 3D object detection.

(1) Object point cloud extraction in each keyframe

The visual SLAM algorithm ORB-SLAM2 is used as the pose estimation algorithm. To improve processing efficiency, object detection is performed only in keyframes. We employ YOLO v3 [20] to conduct object detection, which is an end-to-end and real-time method based on deep learning. It can achieve both accurate and real-time performance (35fps on YOLO v3-416, and with the accuracy of 55.3 mAP on the COCO dataset [21], with both were done on Geforce GTX Titan X GPU).

For the *n*-th keyframe I_n, 2D bounding boxes of objects can be obtained when conducting object detection. Then, combining with a depth image, object point clouds $P_{nk}(k \in N)$ can be obtained. Let world coordinate system {O$_W$}, and pose transformation matrix of the *n*-th keyframe T_n, the object point cloud P_{nk} can be transformed to {O$_W$}, described as P^W_{nk} :

$$P^W_{nk} = T_n P_{nk} \tag{1}$$

(2) Object point cloud segmentation

As the 2D bounding boxes in an image inevitably include background, the obtained object point clouds would also contain background points. This situation will adversely affect the accuracy of 3D object bounding box. Therefore, it is necessary to remove the object background point cloud first.

First of all, point cloud filtering including a voxel filter and a statistical filter is conducted in each object point cloud in order to remove noise points and to reduce computational complexity. Then, the LCCP algorithm (Locally Convex Connected Patches) which is an unsupervised learning method is used to remove the extra background point cloud. Firstly, the whole point cloud is subdivided into several small voxels by using super segmentation. Subsequently, an adjacency graph is computed to connect nearby supervoxels. Finally, supervoxel clustering is conducted by considering the convexity and concaveness of supervoxels. We use a region growth algorithm that grows a region across the convex side is used to cluster smaller supervoxels into larger ones. The point cloud is subdivided into several parts by using this method. Since we assume the object to be usually at the center of a 2D bounding box and occupies most area in the box, the object point cloud can be selected out from background point cloud according to the category of center area and number of points.

(3) Manhattan frame estimation of object point cloud

In order to estimate the smallest bounding box of the object, the main direction of the object has to be calculated. We utilize the Manhattan world assumption to solve this problem. The assumption states that each plane that is perpendicular to an axis in a coordinate system (called Manhattan Frame (MF)) characterizes the main direction of object distribution.

The Manhattan frame uses a rotation matrix R to express the coordinate system. The matrix can be solved by maximizing the number of inliers over the rotation search space:

$$\underset{R\in SO(3)}{\arg\max} \sum_{i=1}^{N} \sum_{j=1}^{6} [\![\angle\left(\vec{n}_i, R\,\mathrm{e}_j\right) \le \tau]\!], \tag{2}$$

Where $\angle()$ is the vector angle, τ is the threshold to determine the inner point, and $[\![\cdot]\!]$ is an indicator function.

We calculate the surface normal vectors of object point cloud and use them as inputs to solve the rotation matrix R of MF. As the MF of most objects in an indoor environment is rotated around ground normal vector, we apply the constraint to the rotation matrix and obtain the converted matrix R'. We define the coordinate system as {O$_{MF}$} and the origin is located to the center of the object point cloud. Then, the transformation matrix ${}^{W}_{MF}T$ between {O$_{MF}$} and {O$_W$} can be solved.

(4) 3D bounding box estimation based on Manhattan frame

In order to calculate the size of 3D bounding box, the object point cloud should be converted to the Manhattan frame coordinate system {O$_{MF}$}. Afterwards, P^W_{nk} can be transformed to P_{nk}^{MF} in {O$_{\mathrm{MF}}$} by using transformation matrix ${}_{MF}^{W}T$.

$$P_{nk}^{MF} = {}_{MF}^{W}T^{-1} \cdot P_{nk}^{W} \tag{3}$$

Subsequently, the maximum and minimum values of the object point cloud in three axes are calculated respectively in the coordinate system {O$_{\mathrm{MF}}$}. The size of the 3D bounding box is obtained.

(5) Object point cloud fusion using multi-keyframes

We adopt the above method to estimate the 3D objects. However, the object may not be observed completely because of the limited viewing angle and this will reduce the accuracy. When the robot moves, multiple keyframes can be generated by using ORB-SLAM2 algorithm. Therefore, we can fuse multi-keyframes to achieve more accurate estimation of 3D objects.

For each keyframe, 3D objects are detected. We need to determine which object to fuse together. Since different objects have different sizes, we utilize the center point distance of 3D bounding boxes to decide whether objects with the same category need to be fused. Finally, the fused object point cloud is used to estimate the 3D bounding box.

(6) 3D object-aware semantic map generation

After the above steps, the robot can acquire objects in the indoor environment. As the 3D bounding box has the properties of 3D size, central point, and object coordinate system, we can use it to express the object in the 3D space. Therefore, all the object categories and 3D bounding boxes constitute a 3D map, named 3D object-aware semantic map M_{obj}^{3D} . This map includes object semantics and spatial distribution information.

B. 2D occupancy grid map generation

In order to achieve autonomous navigation in indoor environment, the robot needs a map to conduct global planning and navigation. The 3D point cloud map $M_p^{3D} = \{p_i\}(i \in N)$ of the environment can be established by using ORB-SLAM2 (p_i is the 3D point in the {O$_{\mathrm{MF}}$}). However, the map M_p^{3D} uses a lot of storage space and it cannot be leveraged by a robot with limited storage to navigate. Therefore, the map M_p^{3D} needs to be converted to a suitable map representation.

The 2D occupancy grid map is usually adopted for robot navigation when using a laser-based method. The metric map is quantized to a grid map according to the set resolution. Each grid in the map has three states: occupied, free or unknown. Therefore, we can also use this type of map for the RGB-D camera situation. The octomap method is utilized to transfer the point cloud map M_p^{3D} to the 3D occupancy grid map M_g^{3D} . As the robot performs planning and navigation in 2D space (i.e. floor), unlike the drone in 3D space, we project the 3D grid map M_g^{3D} to a 2D plane. Generally, we project to the floor plane and obtain the 2D occupancy grid map M_g^{2D} .

C. Object-aware hybrid map generation

The map M_p^{3D} generated in section A includes object information which can be used to relate to human semantic commands. For example, if the user command is “bring me a cup”, the robot can recognize the keyword “cup” by speech recognition and search it in the map in order to obtain the pose of the “cup”. However, this map is not easy to be leveraged to conduct path planning and navigation for the robot. The map M_g^{2D} generated in section B is a 2D grid map to make up for this deficiency. Therefore, in order to combine the advantages of both maps, we design a new object-aware hybrid map including object semantics and metric information. As both maps are in the world coordinate system {O$_{\mathrm{W}}$}, the map M_{obj}^{3D} can be fused with the 2D map M_g^{2D} directly. Finally, the new hybrid map M_{hybrid}^{3D} can be constructed, expressed as:

$$M_{hybrid}^{3D}{=}M_{obj}^{3D}{+}M_g^{2D} \tag{4}$$

D. 2D object pose update for semantic navigation

The hybrid map is built and can be used to conduct semantic navigation for the robot. People give semantic command, such as “please find a cup”. The keyword “cup” is detected by speech recognition. And then the object is searched in the hybrid map. Finally, the central point of object can be obtained as the navigation goal. However, the robot may not be able to arrive at this goal as it is surrounded by occupied grid cells in 2D. We need to update the navigation goal to a grid cell with the free status.

The diagram of navigation goal calculation is shown in Fig 2. In the 2D hybrid map $M_{g\_obj}^{2D}$, we take the object 1 as an example to calculate the navigation goal. We need to find a suitable goal instead of the central point as the coordinate system of object 1 has been obtained based on Manhattan frame. We can search in a set step along the four directions ($\vec{e}_x$, -$\vec{e}_x$, $\vec{e}_y$, -$\vec{e}_y$) of the object coordinate axes to find the grid cell with the free status. In order to ensure safety, the goal should be set to be within a safe distance ψ to the nearby occupied cell as the robot has a certain size (ψ is often set as half a diameter of the robot). After obtaining the goals meeting the requirements, the goal closest to the robot $\arg_i(\min(|\vec{d}_i|))(i \in N^+)$ is selected. Finally, the robot uses this new goal to implement semantic navigation.

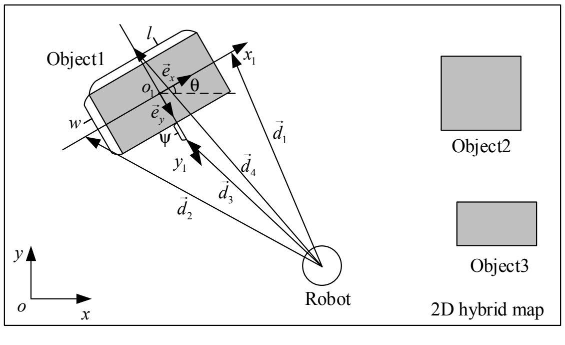


Fig. 2 The diagram of navigation goal calculation.

IV. Experiments

Experiments were implemented to demonstrate the object-aware hybrid map for the indoor visual semantic robot navigation. As the 3D object detection was a vital module in the construction of an object-aware hybrid map, we carried out quantitative experiments on datasets to verify the effectiveness of 3D object detection in our method. And then, the experiment on object-aware hybrid map creation and semantic navigation application for the robot in an actual environment was conducted.

To illustrate the details of the 3D object detection in our method clearly, an example including the intermediate results was shown in Fig. 3. The sofa was detected in the input keyframe from ORB-SLAM2 algorithm. Then, the object point cloud was extracted and we conducted filtering before segmentation. After supervoxel segmentation and clustering, the object point cloud without the background was obtained. The Manhattan frame estimation was conducted to calculate the frame of the object. Finally, the 3D bounding box was estimated.

In order to quantify the performance of the 3D object detection module in our method, the SceneNN [17] dataset was selected to evaluate the accuracy. The dataset provided raw continuous RGB and depth images captured by an Asus Xtion Pro sensor. 3D map representations and per-pixel annotations were given in the dataset. To compare with other work [22, 12], we used the same scenes in the dataset to conduct experiments. In the previous work [22], the NYU RGB-D V2 dataset was utilized to evaluate 40 classes of object segmentation accuracy. However, the YOLO v3 used in our method was trained on the Microsoft COCO object dataset with 80 classes. Therefore, in order to evaluate the accuracy, the common 9 classes were used to calculate the accuracy. Finally, the average accuracy was given in Table 1 in which 10 scenes were used to evaluate. The experimental result showed that our method could achieve better accuracy than the work of Pham et al. [22] in eight out of ten scenes and the work of Grinvald et al. [12] in six out of ten scenes. The final results on scenes 011, 016 and 030 were also given to observe intuitively, as shown in Fig.4.

The actual experiment was performed in an indoor room to construct the object-aware hybrid map and then we demonstrated how to use the map to perform semantic navigation for the robot. A Turtlebot robot with an Asus Xtion Pro RGB-D sensor was used in our experiment. The whole process was shown in Fig. 5. We controlled the robot to move in an indoor room to construct the 3D point cloud map using ORB-SLAM2 (Fig. 5(1)). Meanwhile, the 3D object detection based on YOLO v3 was conducted (Fig. 5(4)). After the 3D mapping, the 3D occupancy grid map was generated with a resolution of 5cm (Fig. 5(2) and (5), different colors mean height). Then, the 3D grid map was projected to the plane perpendicular to the ground normal vector. The 2D occupancy grid map was acquired (Fig. 5(3)). Finally, the object-aware

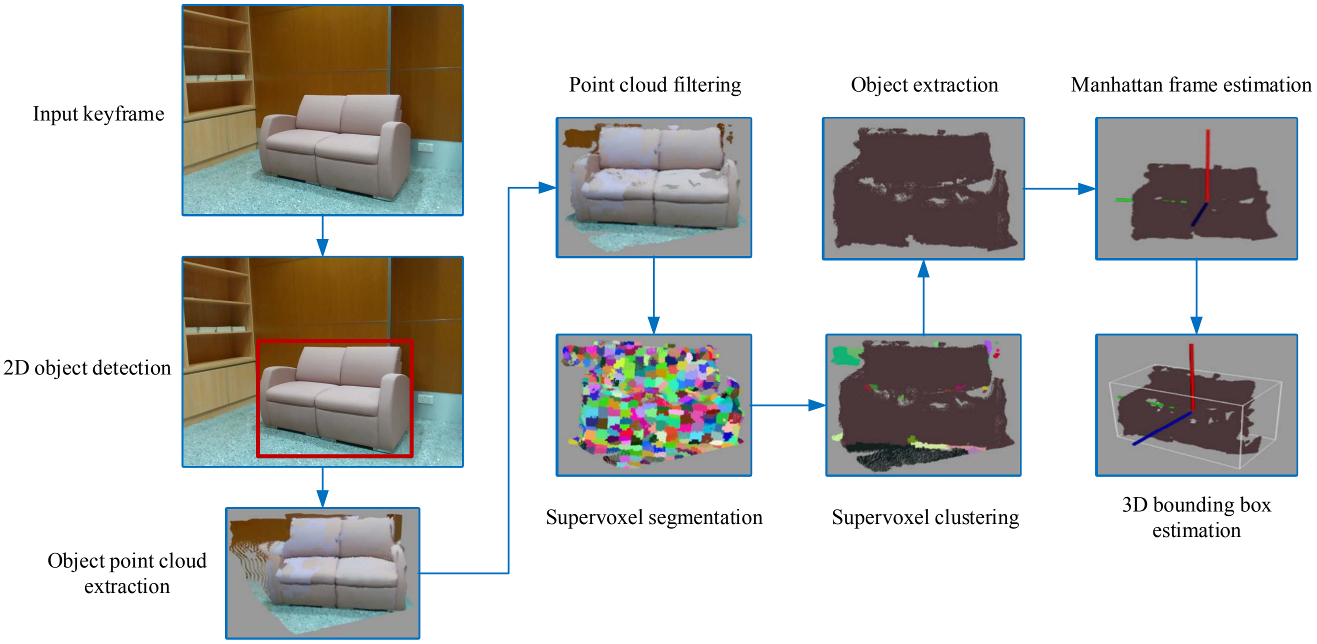
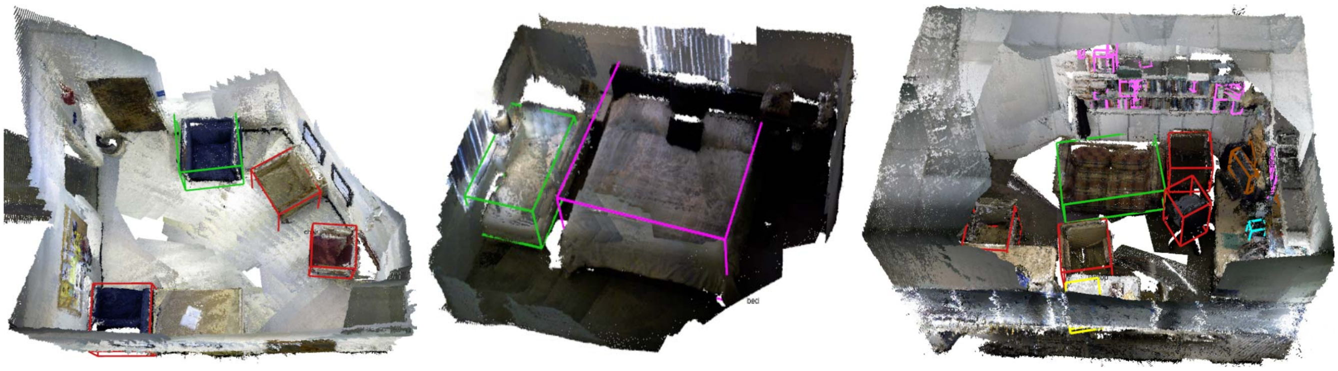


Fig. 3 The details of 3D object detection in our method.

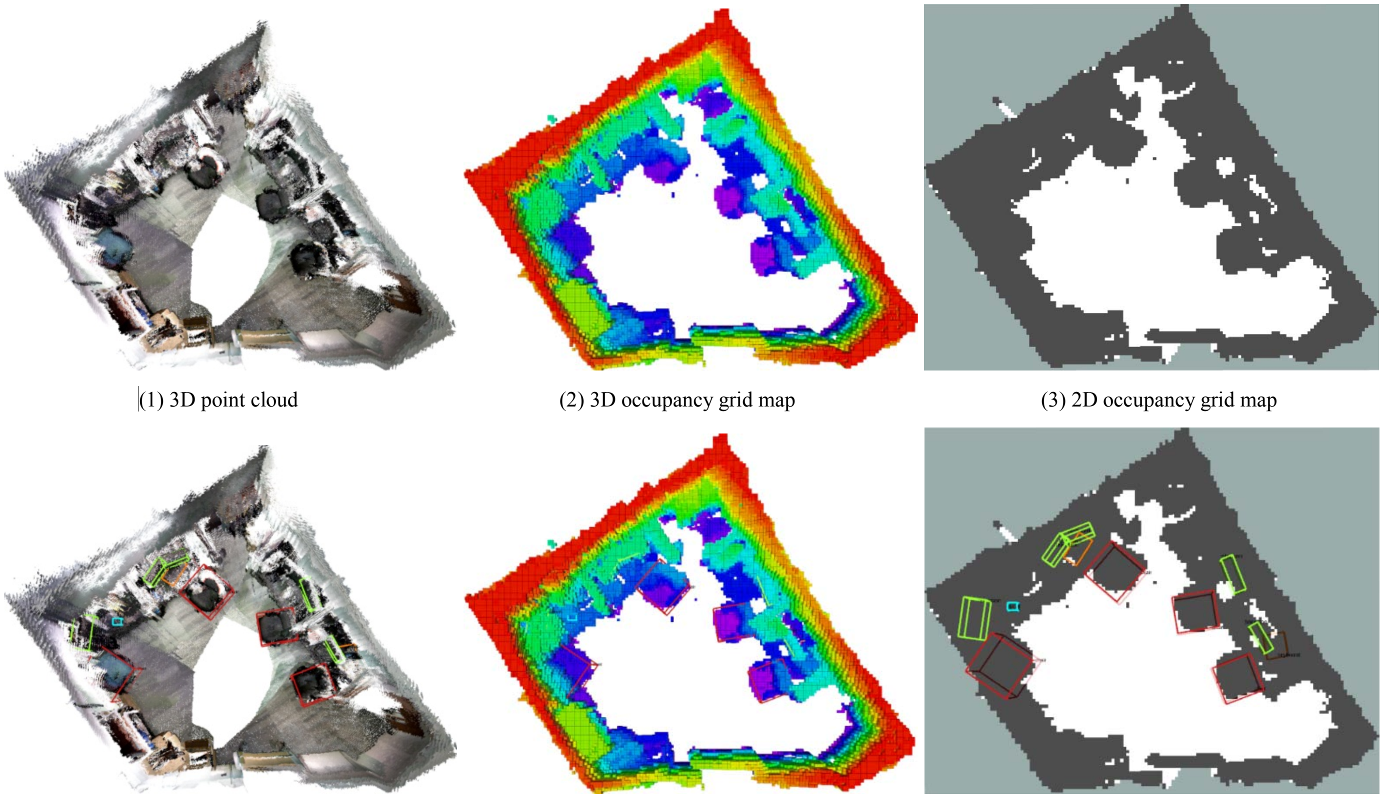
Table 1. 3D object detection experiments on 10 scenes of SceneNN dataset (units: %).

Scene ID Methods	011	016	030	061	078	086	096	206	223	255
[30]	52.1	34.2	56.8	59.1	34.9	35.0	26.5	41.7	40.9	48.6
[12]	**75.0**	33.3	56.1	**62.5**	45.2	20.0	29.2	**79.6**	43.8	**75.0**
Our method	62.9	**38.1**	**58.5**	41.2	**46.5**	**38.4**	**35.7**	40.2	**45.1**	52.5



(1) Scene 011 (2) Scene 016 (3) Scene 030

Fig. 4 3D object detection on scenes 011, 016, and 030 of SceneNN dataset.



(1) 3D point cloud (2) 3D occupancy grid map (3) 2D occupancy grid map

(4) 3D point cloud with 3D objects (5) 3D occupancy grid map with 3D objects (6) 2D occupancy grid map with 3D objects

(television: green, chair: red, keyboard: orange, bottle: blue)

Fig. 5 Object-aware hybrid map construction.

hybrid map with a 2D occupancy grid map and a 3D object-aware map was obtained (Fig. 5(6)).

The semantic navigation experiment in the robot was done by using the object-aware hybrid map. We utilized the speech command recognition module from Iflytek CO., LTD. to translate user speech to executable commands for the robot. In the map, there were five monitors (the label was television in the COCO dataset), four chairs, two keyboards, and one bottle. In order to avoid ambiguity, we selected the unique bottle as the navigation goal. We set user voice command as "Find a bottle". Then, the object "bottle" was searched in the hybrid map and the central point coordinate of the bottle in the world coordinate system was obtained. Afterwards, the coordinate was updated to a reachable coordinate according to the method described in the third part. Subsequently, the path planning and navigation were conducted in the Turtlebot robot by using available methods in ROS (Robot Operating System). The experimental result with the trajectory was shown in Fig. 6.

V. Conclusions

In this paper, we propose a novel object-aware hybrid map for visual semantic navigation of the robot in indoor scenes. This hybrid map combines a metric map with object semantics. Compared with 3D dense point cloud map or other object-based semantic maps, our map makes it possible for an intuitive interaction between human, robot, and environment. Moreover, the map occupies tiny memory space and it needs minor calculation. In future work, we will focus on the ambiguous problem caused by multiple objects with the same label in the map.

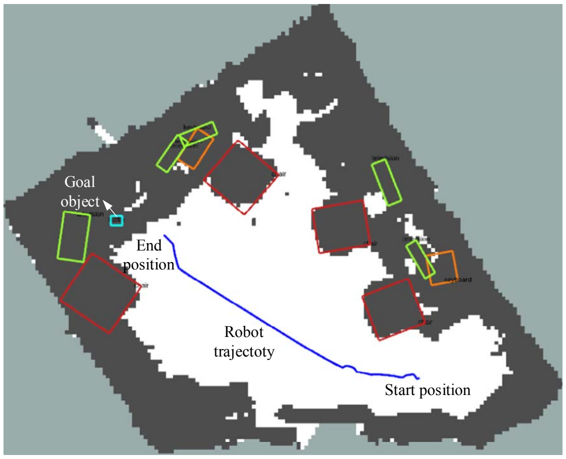


Fig. 6 Semantic navigation utilizing the object-aware hybrid map.

References

[1] G. Klein, and D. Murray, “Parallel tracking and mapping for small AR workspaces,” *IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1-10, 2007.

[2] R. Mur-Artal, and J. Tardós, “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.

[3] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast semi-direct monocular visual odometry,” *IEEE International Conference on Robotics and Automation*, pp. 15-22, 2014.

[4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, pp. 91-99, 2015.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg, “SSD: Single shot multibox detector,” *European Conference on Computer Vision*, pp. 21-37, 2016.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.

[7] A. Hermans, G. Floros, and B. Leibe, “Dense 3d semantic mapping of indoor scenes from rgb-d images,” *IEEE International Conference on Robotics and Automation*, pp. 2631-2638, 2014.

[8] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, “SemanticFusion: Dense 3D semantic mapping with convolutional neural networks,” *IEEE International Conference on Robotics and automation*, pp. 4628-4635, 2017.

[9] T. Whelan, S. Leutenegger, R. Salas-Moreno, and B. Glocker, “ElasticFusion: Dense SLAM without a pose graph,” *Robotics: Science and Systems*, 2015.

[10] C. Qi, H. Su, K. Mo, and L. Guibas, “Pointnet: Deep learning on point sets for 3D classification and segmentation,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 652-660, 2017.

[11] R. Salas-Moreno, R. Newcombe, H. Strasdat, P. Kelly, and A. David, “SLAM++: Simultaneous localisation and mapping at the level of objects,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1352-1359, 2013.

[12] M. Grinvald, F. Furrer, T. Novkovic, J. Chung, C. Cadena, R. Siegwart, and J. Nieto, “Volumetric instance-aware semantic mapping and 3D object discovery,” *IEEE Robotics and Automation Letters*, vol. 4, pp. 3037-3044, 2019.

[13] S. Song, and J. Xiao, “Deep sliding shapes for amodal 3D object detection in rgb-d images,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 808-816, 2016.

[14] Z. Deng, and L. Latecki, “Amodal detection of 3D objects: Inferring 3D bounding boxes from 2d ones in rgb-depth images,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5762-5770, 2017.

[15] J. Lahoud, and B. Ghanem, “2D-driven 3D object detection in RGB-D images,” *IEEE International Conference on Computer Vision*, pp. 4622-4630, 2017.

[16] C. Qi, W. Liu, C. Wu, H. Su, and L. Guibas, “Frustum pointnets for 3d object detection from rgb-d data,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918-927, 2018.

[17] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from RGBD images,” *European Conference on Computer Vision*, pp. 746-760, 2012.

[18] S. Song, S. Lichtenberg, and J. Xiao, “SUN RGB-D: A RGB-D scene understanding benchmark suite,” *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 567-576, 2015.

[19] B. Hua, Q. Pham, D. Nguyen, M. Tran, L. Yu, and S. Yeung, “SceneNN: A scene meshes dataset with annotations,” *International Conference on 3D Vision*, pp. 92-101, 2016.

[20] J. Redmon, and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint*, arXiv:1804.02767, 2018.

[21] T. Lin, M. Maire, S. Belongie, et al. “Microsoft coco: Common objects in context,” *European Conference on Computer Vision*, pp. 740-755, 2014.

[22] Q. Pham, B. Hua, D. Nguyen, and S. Yeung, “Real-time progressive 3D semantic segmentation for indoor scene,” *IEEE Winter Conference on Applications of Computer Vision*, pp. 1089-1098, January 2019.