


RESEARCH ARTICLE

Long-term object search using incremental scene graph updating

Fangbo Zhou^{1,†} , Huaping Liu^{2,*}, Huailin Zhao¹ and Lanjun Liang¹

¹School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai, China and ²Department of Computer Science and Technology, Tsinghua University, Beijing, China.

*Corresponding author. E-mail: hpliu@tsinghua.edu.cn.

Received: 11 March 2022; **Revised:** 11 June 2022; **Accepted:** 20 July 2022; **First published online:** 22 August 2022

Keywords: household robot, long-term object search, incremental scene graph

Abstract

Effective searching for target objects in indoor scenes is essential for household robots to perform daily tasks. With the establishment of a precise map, the robot can navigate to a fixed static target. However, it is difficult for mobile robots to find movable objects like cups. To address this problem, we establish an object search framework that combines navigation map, semantic map, and scene graph. The robot updates the scene graph to achieve a long-term target search. Considering the different start positions of the robots, we weigh the distance the robot walks and the probability of finding objects to achieve global path planning. The robot can continuously update the scene graph in a dynamic environment to memorize the position relation of objects in the scene. This method has been realized in both simulation and real-world environments. The experimental results show the feasibility and effectiveness of this method.

1. Introduction

Indoor mobile robots have developed rapidly in recent years thanks to the achievements of environment perception, mapping, and path planning. Especially in well-structured environments, such as hotels, exhibition halls, etc., the robots have achieved remarkable success because of the pre-established accurate maps by SLAM [1, 2, 3, 4] in such scenarios.

In these scenarios, the robot can accurately reach a fixed target location on the map, during which it can successfully avoid dynamic obstacles [5, 6, 7, 8, 9] such as people. A typical example is that a hotel guest can get a toothbrush by just informing the service center that will employ a robot to send the toothbrush. However, the robots are limited in these tasks in which the fixed position of the targets is stored in advance.

We can still follow the above methods if the robot search almost fixed targets such as refrigerators and beds in indoor scenes. In fact, the techniques of SLAM, sensor-based obstacle detection, and path planning [10] have solved the tasks in such known scenarios. However, when looking for objects such as apples and remote controls, the positions of these objects are easy to be changed. For example, an apple can be on the table or the coffee table. Even if the robot has good location and obstacle avoidance capabilities, the robot still does not know where to find the target. Searching for “semi-dynamic objects” is still a non-trivial task.

The effective way to solve this problem is to introduce prior knowledge. Although the robot does not know the location of the apple, it can infer possible candidate targets through objects that often coexist with the cup (such as a table, a coffee table, etc.). We can transform the problem of searching for

[†]This work was completed while Fangbo Zhou was visiting Tsinghua University, Beijing, China.

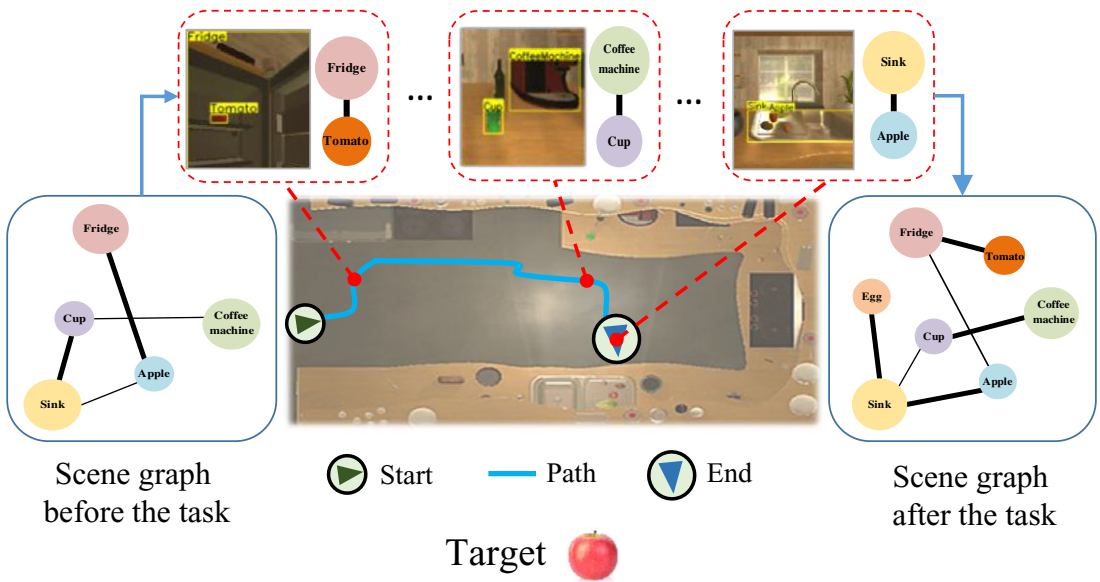


Figure 1. The robot performs the task of searching for an apple. In the process of searching, the robot find a tomato (semi-dynamic object) in the fridge (static object). The robot can update the scene graph according to the location relationship of the object. The updated scene graph is helpful for the next search task.

semi-dynamic objects into searching for related static objects. This idea has recently been widely used in visual semantic navigation [11, 12, 13].

However, using the pre-established fixed knowledge graph to help search for unseen objects is inefficient since the relation between the target and other objects is just learned by public datasets without considering the specific scenery. In our concern, the navigation target location is not fixed, or unseen objects that do not appear in the knowledge graph. This motivates us to develop a long-term object search method. As shown in Fig. 1, the core of this problem is how to develop a framework in which the robots can update the scene graph to describe the scenery during the search task.

Compared with the existing work [12, 13, 14, 15], our work can directly use the physical platform of the mobile robot. And comprehensively use the perceptual learning ability of the embodied intelligence field and the SLAM, path planning, and other technologies in the robotics field to build a target search platform with long-term learning ability. We use a scene graph to represent the concurrent relationship between the target object and other static objects. The robot can first explore around known static objects and gradually approach semi-dynamic objects. Because of the different initial positions of the robot, global path planning is realized by weighing the length of the path and the probability of object discovery. To memorize the scene quickly in the process of performing the task, we propose an incremental scene graph updating method in the semi-dynamic environment to dynamically update the scene graph. The updated scene graph will be helpful in planning a path for the next object search task. The main contributions of this study are summarized as follows:

- For object search in a known environment, we propose a framework of object search combined with established navigation, semantic, and semantic relation maps. Global path planning is realized by weighing the length of the path and the probability of object discovery.
- The robot can continuously update the semantic relation graph in a dynamic environment to memorize the position relation of objects in the scene.
- Real-world experiments have shown that our proposed method can be applied perfectly to physical robots.

The rest of this paper is organized as follows. Section 2 demonstrates the related works of Map-based navigation approaches, Learning-based approaches, and Scene Graph. In Section 3, the problem definition is introduced. In Section 4, we introduce the framework we propose and its components in detail. In Section 5, we introduce the semi-dynamic object search and incremental scene graph updating. We introduce the experiment details and analysis in Section 6 and conclude our method in Section 7.

2. Related work

Map-based navigation approaches: Classical navigation methods have divided the problem into two parts, mapping and path planning. A geometric map is usually built using SLAM [4, 16] technology. Once the environment map has been constructed, the robot can use a path planning algorithm, such as A* [17] or RRT* [18], to generate a collision-free trajectory to reach the target location, even if there are obstacles [19, 20]. With the development of deep learning, some work [21, 22, 23] built detailed semantic maps from images for complex indoor navigation learning in simulator [24, 25, 26]. However, the above method of building a semantic map is based on the premise that the robot can obtain accurate camera coordinates. However, accurate indoor positioning of robots in unknown physical environments is still a problem. On the one hand, we provide accurate location information to the robot through SLAM. On the other hand, we combine semantic map and navigation map to assist the object search task.

Learning-based approaches: To tackle the problem of object search in unknown scenes [27, 28, 29, 30], Zhu et al. [31] first proposed a target-driven navigation task. They used a pair of twin networks with shared parameters to extract the features of the currently observed image and the target image information. Then, they used the A3C reinforcement learning algorithm as the decision-making part of the robot. After that, many navigation methods [14, 15, 32] use deep reinforcement learning and imitation learning to train navigation strategies. Yang et al. [14] used graph convolutional neural networks to combine the prior information of the scene into the deep reinforcement model, and the categories of target objects were also encoded as word vectors. Wortsman et al. [15] also proposed a meta-reinforcement learning strategy that encourages agents to continue learning in a test environment using self-supervised interaction losses. Then, Mousavian et al. [32] used the semantic mask obtained using the current state-of-the-art computer vision algorithm as the result of the current observation image and used the deep network to learn the navigation strategy. They are made great progress in simulation environments. However, the end-to-end learning methods require a series of abilities, such as object detection [33], semantic prior [34], and obstacle avoidance [35]. The diversity of light, object materials, colors, and layouts in real environments have prevented the transfer of this progress in simulation platforms to real-world scenarios. The robot is still unable to adapt to the complex and changeable physical environment effectively. In the physical environment, the robot is likely to bump into people or other objects because of the defect of obstacle avoidance ability, and the cost of its wrong decision is very expensive.

Scene Graph: Many researchers [11, 12, 36] have noticed a concurrence between objects, for example, the remote control often appears next to the television. This concurrence between objects has been studied for tasks such as image retrieval [34] using scene graphs, visual relation detection [37], and visual question-answering [38]. By learning this relationship, the robot can narrow the scope of the search and improve the efficiency of object search. Qiu et al. [13] proposed a hierarchical object relation learning method to solve the problem of object-driven navigation. Although this method of using object relationships can help the robot determine where to go, in the face of a real-world complex, the robot still does not know how to go because of the lack of robot navigation ability. Zeng et al. [39] used the spatial relationship between room landmarks and target objects to introduce a semantic link map model to predict the next best view posture to search for the target objects. However, no connection exists between the tasks performed by the robot. The robot performing the last task will be helpful for the next task in our method, even if the target object is different.

3. Problem formulation

We set a semi-dynamic environment E_t that changes with time, and all objects in the room $O = \{o_1, \dots, o_n\}$, which contains static object o_s and semi-dynamic object o_d (explained in detail in Section 4.2).

When the robot executes the search task at a certain time t , given the robot's initial scene graph $SG_t = (O, E)$, where each node $o \in O$ represents the category of an object. Each edge $e \in E$ represents the value of the relationship between a static object and a semi-dynamic object $Rel(o_d, o_s) \in [0, 1]$.

The robot is placed at the initial position p_i in the environment E_t and gives the robot a semi-dynamic object target O_{target} , such as a cup. The robot passes through the scene graph to establish the association between o_{target} and the static object o_s . The robot plans a most efficient path to find the semi-dynamic object and gets close to the object.

The position of semi-dynamic objects will change over time, such as a mobile phone is sometimes on the table and sometimes on the sofa. Therefore, in the search process, the robot needs to continuously obtain the experience of correlation between objects in E_t , it can be denoted Exp_t . At the end of the task, these associated experiences are used to update the scene graph:

$$SG_{t+1} = \text{UPDATE}(SG_t; Exp_t) \quad (1)$$

The robot can use the updated scene graph SG_{t+1} for path planning of the next task.

The operating environment of our robot is a semi-dynamic scene, which allows the robot to build auxiliary maps M in advance, including navigation M_{nav} and semantic maps M_{sem} , before performing the object search task. The navigation map consists of several navigable viewpoints N_{nav} , and the distance between the navigable viewpoints and adjacent navigable viewpoints is 0.25 m. On the basis of this, the robot can not only obtain the accurate pose $x_t \in \mathbb{R}^3$ but also realize the path planning from one viewpoint to another.

4. Framework

4.1. Overview

The search target of the robot belongs to a semi-dynamic object, and the robot tends to find the target object near the static object. When the target is found, the nearest navigable viewpoint is reached. As shown in Fig. 2, the robot combines the semantic map and the scene graph to plan a path that is most likely to find the target object within the shortest distance. The path consists of the starting point of the robot, the nearest navigable viewpoint of the static object with a non-zero relation value to the target object, and the endpoint. When the robot finds the target object, the robot carries out local semantic reconstruction of the target object. It reaches the nearest navigable viewpoint of the target object, which is the destination.

In the process of searching the target object along the path, the robot will store the relationship pairs between semi-dynamic objects and static objects in the experience pool, such as the bowl with the table and the cup with the sink. Then, the scene graph is updated through such relationship pairs to help find the target faster next time. Specifically, our method updates the object relationship value between semi-dynamic objects and static objects and then adjusts the path of the next search target object.

4.2. Scene graph

Following the classification of objects by Meyer-Delius et al. [40], we have slightly modified the classification as follows:

- **Static objects:** objects are large and not easily moved in a room are called static objects. For example, a refrigerator is a static object in a kitchen scene, and a bed is a static object in a bedroom. Such objects will help the robot to find semi-dynamic objects.

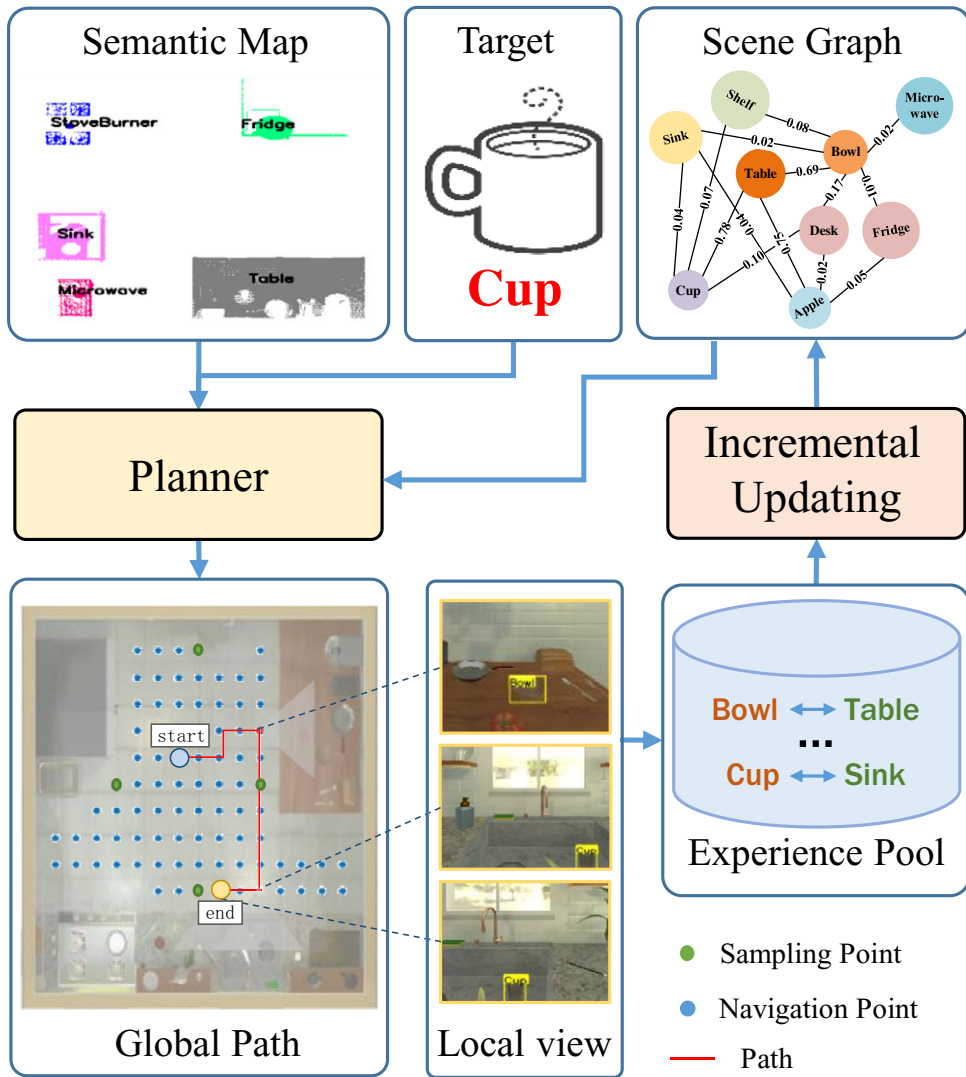


Figure 2. The architecture overview of our propose navigation method equipped with the scene graph and semantic map.

- **Semi-dynamic object:** it is static during the search process, but its position can be easily changed, such as an apple or cup. This type of object is what the robot’s search target.
- **Dynamic objects:** The position of an object changes easily, even during the search, such as moving people and moving pet dogs. Such objects are not the search target of the robot.

Following Yang et al. [14], we also extract the relationship between semi-dynamic and static objects from the image captions in the Visual Genome dataset (VG) [41]. But the difference is that our relationship graph has definite values for the strength of the relationship between objects. For a semi-dynamic object o_d and a static object o_s , the object relationship can be expressed as $Rel(o_d, o_s) \in [0, 1]$, the larger the value, the closer the relationship between them. $Rel(o_d, o_s)$ can be calculated as follows:

$$Rel(o_d, o_s) = \frac{C(o_d, o_s)}{C(o_d)} \tag{2}$$

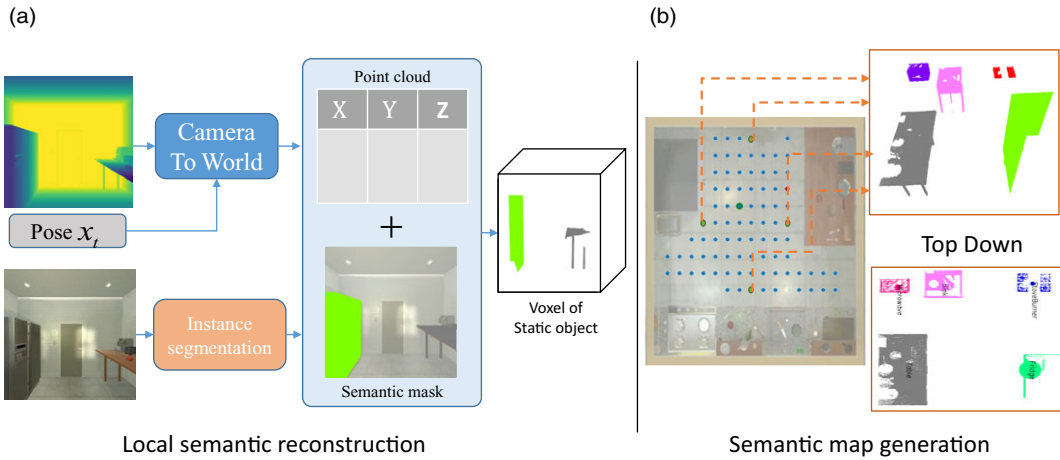


Figure 3. (a) *Local semantic reconstruction:* The robot obtains the semantic point cloud of each static object from the RGB and depth image observed on the pose $x_t \in \mathbb{R}^3$. (b) *Semantic map generation:* it consists of two parts: first, the global semantic reconstruction is carried out, and then the semantic map is projected from top to bottom.

where $C(d, s)$ represents the number of times that the semi-dynamic object d and a static object s appear together in the image captions, and $C(d)$ represents the total number of the semi-dynamic object appears in the image caption. In addition, we combine aliases of objects, such as “Cellphone” and “Phone”.

4.3. Semantic map

As shown in Fig. 2, on the boundary of all navigable viewpoints, the navigable viewpoint of the median coordinate of all navigable viewpoints on the boundary is taken as the sampling point. The robot captures information about the room by collecting RGB and depth images of the view every 45 degrees at each sampling point. As shown in Fig. 3, Similar to ref. [22], we first perform local semantic reconstruction for a single view of a single sampling point. We can obtain the robot pose $x_t \in \mathbb{R}^3$ through SLAM, which represents the coordinates and orientation of the robot on the navigation map. In the real world, we can predict the semantic mask of the current observation by the existing target detector Mask-RCNN [42] and project it into the point cloud.

The robot performs local semantic reconstruction at each perspective of each sampling point. In order to overcome the shortcoming of insufficient environmental information obtained from a single perspective, we integrate semantic point clouds generated by multiple sampling points and multiple perspectives to complete local to global semantic reconstruction. The voxel of static objects contains semantic information in the whole room. By projecting from top to bottom, we can get a semantic map with information about the spatial distribution of static objects.

5. Simultaneous search and scene graph updating

5.1. Semi-dynamic object search

In order to facilitate the robot to reach the nearest navigable viewpoint of the static object o_s , we perform the following calculations on the objects on the semantic map: (a) calculate the center of mass of an object semantic point cloud as the object’s position absolute position loc_s in the semantic map. (b) the nearest viewpoint: a navigable viewpoint $N_s \subset N_{nav}$ that is closest to the absolute position of the static object. (c) the appropriate observation angle θ_s : according to the position of the object and the nearest

navigable viewpoint, the robot can calculate the angle at which the robot can observe the static object after moving to the navigable viewpoint.

For semi-dynamic objects, the search strategy is that the robot first looks for the area around the static objects related to the target object and then gradually approaches the target object. Specifically, the robot will first search in the room where the starting point is located. If it does not find it, the remaining unexplored rooms are sorted according to the size of $Rel(o_{target}, R)$, which can be defined as the sum of all $Rel(o_d, o_s)$ in room R . According to the sorted results, the robot selects the next room to explore until the robot finds the target object.

Because of the difference in initial positions, the robot needs to weigh the path's length against the probability of finding the object. We propose a path length weighting (WPL) method to evaluate all possible paths. If there are n static objects associated with o_{target} in the room, then the number of all possible search paths is $n!$. The WPL can be expressed as

$$WPL = \sum_{i=0}^n \frac{L_i}{(1 + \alpha * Rel(o_{target}, o_s)) * 2^i} \quad (3)$$

Among them, L_i represents the distance from the previous position to the nearest navigable viewpoint of the static object. $Rel(o_{target}, o_s)$ represents the relationship value between the static object and the target object. The hyperparameter α can control the influence of the object relationship value on the path. Choose the path with the smallest WPL value as the path searched by the robot.

We divide static objects into two types: container type, such as refrigerator and microwave oven, and non-container type, such as table and bed. For container-like static objects, the robot needs to open the container and find the target. When the robot reaches the vicinity of a non-container static object, the robot will rotate 45 degrees left and right to find the target object. If the robot finds an object on the planned path, it will perform a local 3D semantic reconstruction of the target object and find the navigable viewpoint and orientation closest to the target object to approach the target object. If the target is not found, the task is failed.

5.2. Incremental scene graph updating

Because the object relationship is extracted from the visual genome, and it is not necessarily suitable for all scenes, and the position of the semi-dynamic object may change with time, the robot needs to update the value of the relationship between the semi-dynamic object and the static object. When the robot is performing a search task, if a semi-dynamic object is found on the navigable viewpoint N_s closest to a static object o_s , the robot considers the relationship between the static object and the dynamic object as relationship pair. The robot can store this relationship pair in the experience pool. Only when the area of the detected object mask reaches a certain threshold, the object is considered to be seen. Therefore, to a certain extent, when the distance between the semi-dynamic object and the static object is relatively close, the robot can store this relationship pair in the experience pool. At the end of the task, the robot updates the relationship graph based on the relationship pairs in the experience pool. For each semi-dynamic object o_d , the strategy for updating the object relationship between it and the static object o_s is as follows:

$$Rel(o_d, o_s)' = \begin{cases} Rel(o_d, o_s), & \text{if } o_d \text{ is not visible} \\ Rel(o_d, o_s) / 2, & \text{if } o_d \text{ is visible but not at } N_s \\ Rel(o_d, o_s) / 2 + 0.5, & \text{if } o_d \text{ is visible at } N_s \end{cases} \quad (4)$$

Among them, $Rel(o_d, o_s)$ and $Rel(o_d, o_s)'$ are the relationship values between o_d and o_s before and after the update, respectively.

As shown in Fig. 4, the robot according to the relationship pairs store in the experience pool in the process of finding the cup in Fig. 2. The scene graph is updated based on the relationship of the

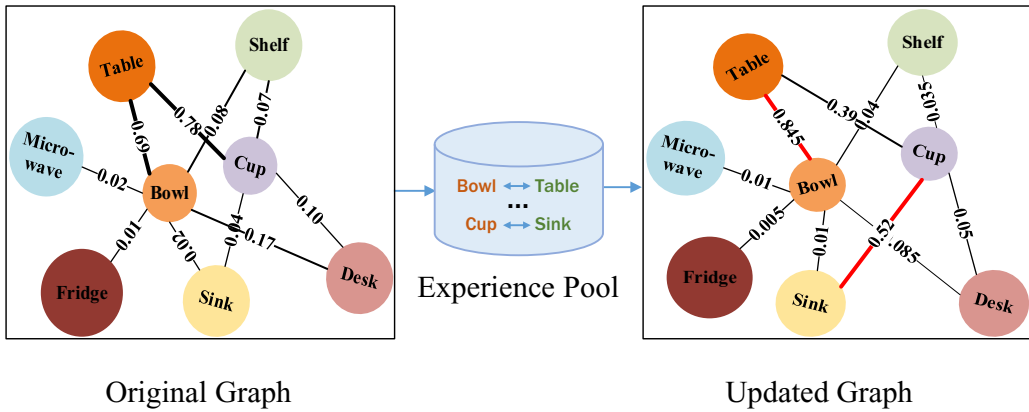


Figure 4. Update the original relationship graph according to the relationship pair corresponding to the bowl and cup in the experience pool. Please note that the semantic relationship values in the updated scene graph on the right have changed compared to the left.

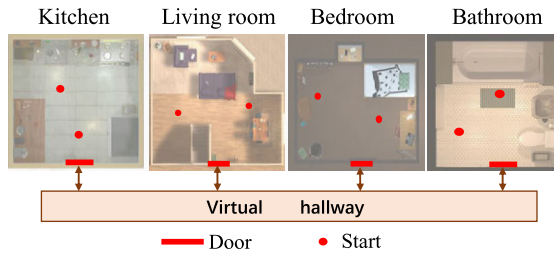


Figure 5. The top view of the combined room, and the location of the starting point.

experience pool and has no direct relationship with the task’s success. If the task is successful, but no other semi-dynamic objects are found during the search, there is only one relationship pair of the target object and the static object in the experience pool. If the search task fails and no other semi-dynamic objects are found during the search process, the experience pool is empty, and the relationship graphs of all semi-dynamic objects are not updated.

6. Experiment

We conduct experiments in the simulator and physical environment, respectively. To verify the effectiveness of our proposed method, we use path length (PL) as an evaluation indicator; it can be defined as the total mileage of the robot from the starting point to the discovery of the object.

6.1. Simulation experiment

We first conduct our experiments in Ai2thor [43] simulation environment. The simulator includes four different room categories: kitchen, living room, bedroom, and bathroom. As shown in the Fig. 5, similar to ref. [44], we take a room in each room category to form a home scene together. To verify the adaptability of our method, the robot distributes two starting points in each room. When the robot arrives at the door of a room, it can be directly teleported to any other room. The list of our static objects is as follows: Sink, Table, Microwave, Fridge, Shelf, Sofa, Bed, Toilet, Bathtub, Stoveburner, Desk. In the simulation environment, the detection results from the robot’s perspective, such as the object category and mask, are provided by the simulator. The robot can detect even objects that are far away. In addition,

Table I. The robot uses different methods to search for the results of the path length.

	Apple	Cup	Bowl	Phone	Laptop	Vase	Pillow	Toilet paper	avg
SDP	28.63	8.88	20.88	13.13	16.25	8.31	7.5	13.88	14.68
RVP	15.06	6.50	10.63	8.56	20.5	8.48	5.75	6.69	10.27
Ours	14.25	6.18	10.13	9.00	19.81	8.48	5.75	6.69	10.04

to simulate the real scene, we set a threshold for each object, and the object is considered visible if it masks above a certain threshold.

6.1.1. Comparison of different methods

In this experiment, we count the robot's success rate in finding different objects and the path length in the case of mission success. To ensure the unity of the task, we removed the bowl, laptop, and pillow in the living room, phone, and vase in the bedroom and kept only one Toilet paper in the bathroom. The positions of other objects remain unchanged. Our benchmark includes the following methods:

- The shortest distance priority (SDP): The robot arrives next to the static object related to the target object in the shortest path, regardless of the object relationship value, that is, $\alpha = 0$ in Eq. (4). The robot randomly selects the next room to search.
- The relationship value priority (RVP): The robot only considers the scene graph, sorts according to the value of the object relationship, and arrives next to the corresponding static objects in turn. The robot selects the next room to search according to the value of $Rel(o_{target}, R)$.
- Ours: The robot integrates the position of the static object in the scene graph and the semantic map, and weighs the length of the search path and the probability of finding the object. We set $\alpha = 10$ in Eq. (4).

In this experiment, because the semantic map containing the spatial distribution information of static objects is used to search for semi-dynamic objects, the robot can achieve a 100% success rate. From Table I, we can see that if the scene graph is not considered, the robot does not know that the target object can be found in that room, and can only blindly search for the static object. However, after adding the scene graph, our method can significantly reduce the search path length of most objects.

However, for laptops and vases, after adding the scene graph, the length of the search path is increased. Our analysis shows that the scene graph we extract from VG does not completely conform to the current spatial distribution of semi-dynamic and static objects. In addition, after the introduction of WPL, the robot weighs the distance and the probability of finding the target, once again shortening the length of the search path.

6.1.2. Long-term object search

In this experiment, our main purpose is to verify that we can realize the memory of the scene by updating the scene graph, which can significantly reduce the length of the search path in a dynamic environment. As shown in Table II, the bold font indicates that the position of the object has changed. Some semi-dynamic objects appear in different positions in a room. Every time the robot looks for an object, it will start from eight starting points.

It can be seen from Fig. 6 that whether it is to update only the relationship pair related to the target object in the experience pool or to update all the relationship pairs in the experience pool. Compared with not updating the scene graph, the average path length after updating the scene graph is shortened by 25.27%. In the task of searching for an object once performed by the robot, updating the semantic relationship value of the target object will help the robot to find the target object next time. We use the scene graph update method can effectively shorten the path length of the target search. During the search

Table II. The distribution of semi-dynamic objects in the scene.

Scene	1	2	3	4	5	6	7	8	9	10
Apple	Sink	Sink	Sink	Sink	Fridge	Fridge	Fridge	Fridge	Table	Table
Cup	Table	Table	Sink	Sink	Sink	Table	Table	Sink	Sink	Table
Bowl	Table	Sink	Sink	Table	Table	Table	Sink	Sink	Table	Table
Phone	Table	Table	Table	Desk	Desk	Desk	Desk	Table	Table	Table
Laptop	Sofa	Sofa	Bad	Bad	Bad	Table	Table	Table	Desk	Desk
Pillow	Bad	Bad	Bad	Bad	Sofa	Sofa	Bad	Bad	Bad	Bad
Vase	Shelf	Shelf	Shelf	Table	Table	Table	Shelf	Shelf	Shelf	Shelf
Toilet paper	Toilet	Toilet	Toilet	Toilet	Shelf	Shelf	Shelf	Toilet	Toilet	Toilet

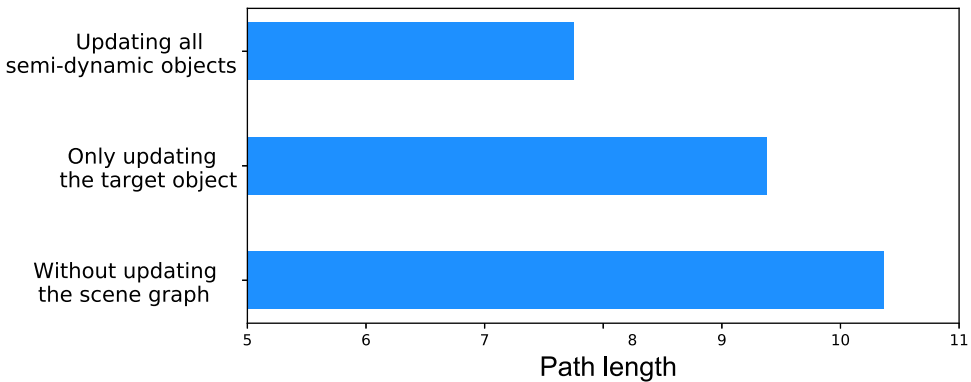


Figure 6. The robot consumes the path length according to the updated scene graph.

process, the robot will also find other semi-dynamic objects. If all semi-dynamic objects are found in the search process, updated scene graph will help the robot to quickly find the newly discovered semi-dynamic objects in next tasks. It can be found that updating other objects during the search process can significantly help the next search task.

Figure 7 shows in detail the trajectories of the robot using the updated and the un-updated scene graph. If the robot finds an object on the planned path, it will perform a local 3D semantic reconstruction of the target object and find the navigable viewpoint and orientation closest to the target object to approach the target object. As shown by the blue line, when the robot reaches the cup in location 2, it will perform a local 3D semantic reconstruction of the target object and find the navigable viewpoint and orientation closest to the target object and reach location 3.

6.2. Real-world experiment

To verify the effectiveness of the long-term target search method in a real-world indoor environment. We construct an experimental scene of about 60 square meters in an office area, including two rooms and a bathroom. The robot is custom built on a business general base, with a 2D Hokuyo Lidar used for localization, and a Kinect RGBD camera for perception. To resolve the limited computation resources on the robot in long processing times by the computer vision software, we use Intel NUC11PHKi7C with Nvidia RTX2060 GPU. A picture of the robot can be seen in Fig. 8(a). After semantic reconstruction, the semantic map of the scene is as shown in Fig. 8(b). In addition, we also artificially set up an obstacle in each task.

Figure 9 shows the path of our robot looking for dynamic objects. We place the cup next to the sink and the scissors on the dining table. During the first search for the cup, the robot plans the path according to the initial scene graph. Although the robot can find the object, it has a long walking route as shown in

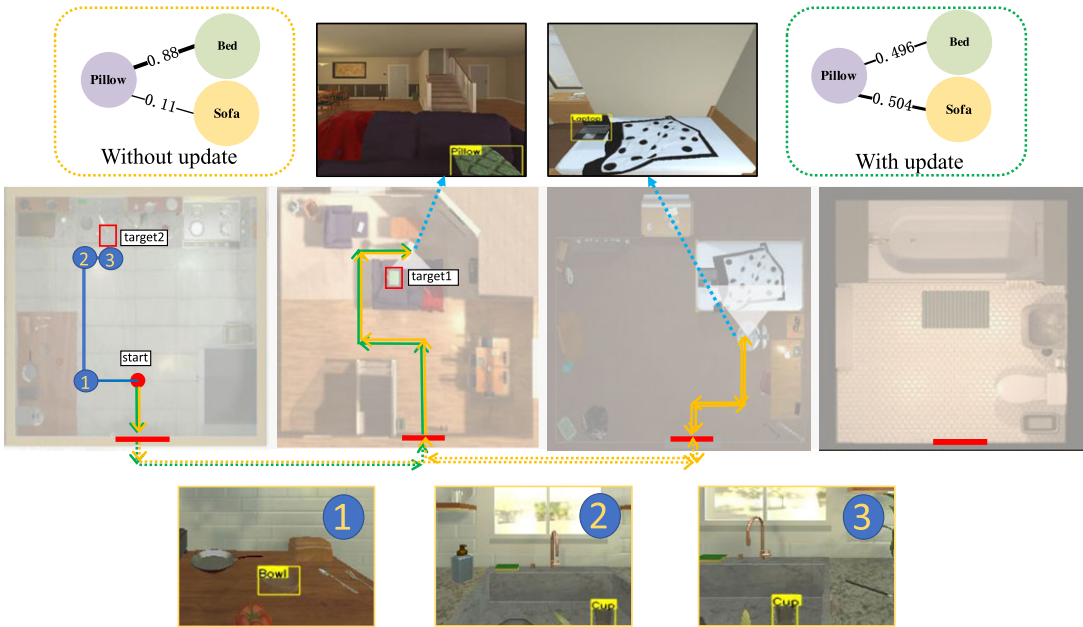


Figure 7. The trajectory of without updating (yellow) and with updating the scene graph (green) to find pillows. The trajectory of blue to find cup.

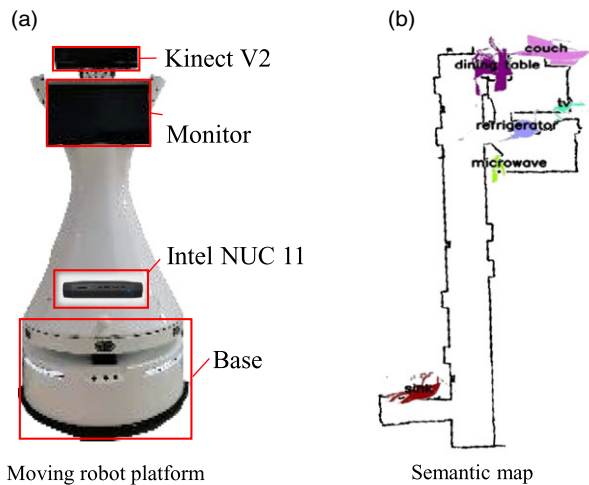


Figure 8. Our moving robot platform and semantic map of the real-world experimental scene.

Fig. 9(a). Because the robot finds the cup next to the sink during the last task, the relationship of the cup will be updated. When the robot looking for the cup for the second time, the robot is more inclined to look for it next to the sink as shown in Fig. 9(b). It is worth noting that in the process of searching for the cup first time, the scissors are found on the table, so the robot can quickly find the scissors in the task of finding the scissors in Fig. 9(c). If the connection between the table and the scissors is not established, the robot does not know where it is more likely to find. According to the distance to the static object, the robot first searches inside the refrigerator and microwave and then searches near the table. The results shall be better viewed in the supplementary videos.

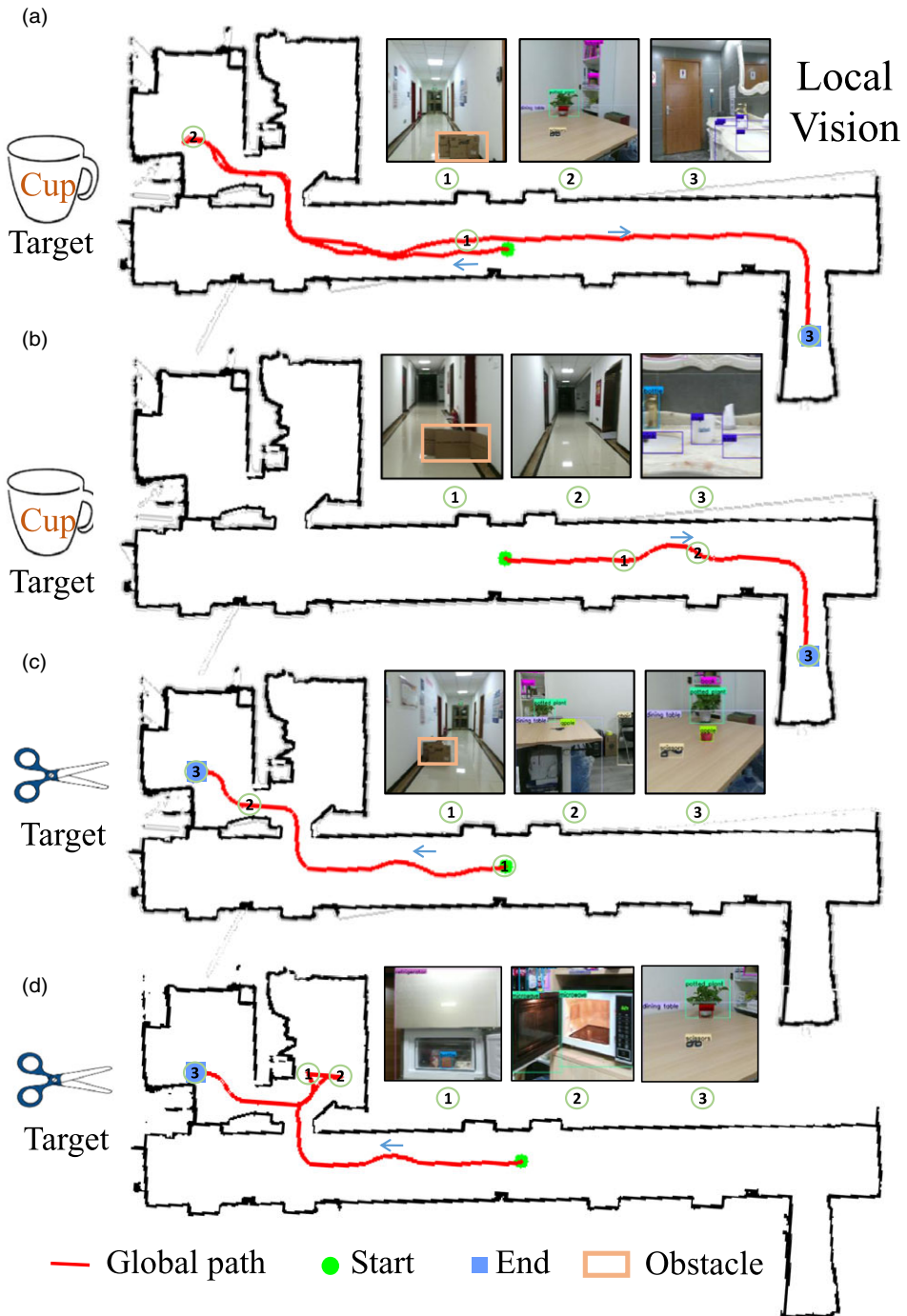


Figure 9. The trajectory of the robot in the real environment.

7. Conclusions

In this paper, we propose a long-term target search method in a semi-dynamic indoor scene, and it can effectively find objects that are easy to move. First, the robot builds a navigation map for path planning,

a semantic map containing the location information of static objects, and a scene graph containing the closeness of the relationship between static objects and semi-dynamic objects. Then the path length weighting method is used to balance the distance and the probability of finding an object. Finally, to establish a scene-specific scene graph, we propose a method of scene memory to be applied to update the scene graph. We conduct experiments in both simulation and physical environments, which demonstrate the effectiveness of our method. However, when semi-dynamic objects are placed in areas without static objects, it will be difficult for the robot to find the target. In future work, our main work is how to find semi-dynamic objects that are less related to static objects.

Acknowledgment. This work was supported in part by the National Natural Science Fund for Distinguished Young Scholars under Grant 62025304 and was supported by Joint Fund of Science & Technology Department of Liaoning Province and State Key Laboratory of Robotics, China (2020-KF-22-06).

References

- [1] W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-time Loop Closure in 2d Lidar Slam," *In: 2016 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2016) pp. 1271–1278.
- [2] L. Kenye and R. Kala, "Improving RGB-D slam in dynamic environments using semantic aided segmentation," *Robotica* **40**(6), 2065–2090 (2021).
- [3] A. Handa, T. Whelan, J. McDonald and A. J. Davison, "A Benchmark for Rgb-d Visual Odometry, 3D Reconstruction and Slam," *In: 2014 IEEE International Conference on Robotics and Automation* (IEEE, 2014) pp. 1524–1531.
- [4] J. Fuentes-Pacheco, J.é Ruiz-Ascencio and J. M. Rendón-Mancha, "Visual simultaneous localization and mapping: A survey," *Artif. Intell. Rev.* **43**(1), 55–81 (2015).
- [5] A. F. K. Emrah Dnmez and M. Dirik, "A vision-based real-time mobile robot controller design based on Gaussian function for indoor environment," *Arab. J. Sci. Eng.* **4**, 1–16 (2017).
- [6] E. Dnmez and A. F. Kocamaz, "Design of mobile robot control infrastructure based on decision trees and adaptive potential area methods," *Iran. J. Sci. Technol. Trans. Electr. Eng.* **44**(2), 431–448 (2019).
- [7] Y. Wei, K. Zhang, D. Wu and Z. Hu, "Exploring conventional enhancement and separation methods for multi-speech enhancement in indoor environments," *Cognit. Comput. Syst.* **3**(4), 307–322 (2021).
- [8] Y. Masutani, M. Mikawa, N. Maru and F. Miyazaki, "Visual Servoing for Non-Holonomic Mobile Robots," *In: IEEE/RSJ/CI International Conference on Intelligent Robots & Systems 94 Advanced Robotic Systems & the Real World* (2002).
- [9] F. Okumu, E. Dnmez and A. F. Kocamaz, "A cloudware architecture for collaboration of multiple agvs in indoor logistics: Case study in fabric manufacturing enterprises," *Electronics* **9**(12), 2023–2047 (2020).
- [10] K. K. Pandey and D. R. Parhi, "Trajectory planning and the target search by the mobile robot in an environment using a behavior-based neural network approach," *Robotica* **38**(9), 1627–1641 (2020).
- [11] H. Du, X. Yu and L. Zheng, "Learning Object Relation Graph and Tentative Policy for Visual Navigation," *In: European Conference on Computer Vision* (Springer, 2020) pp. 19–34.
- [12] R. Druon, Y. Yoshiasu, A. Kanezaki and A. Watt, "Visual object search by learning spatial context," *IEEE Robot. Automat. Lett.* **5**(2), 1279–1286 (2020).
- [13] Y. Qiu, A. Pal and H. I. Christensen, "Learning Hierarchical Relationships for Object-Goal Navigation," *In: 2020 Conference on Robot Learning (CoRL)* (2020).
- [14] W. Yang, X. Wang, A. Farhadi, A. Gupta and R. Mottaghi, "Visual semantic navigation using scene priors." arXiv preprint arXiv: 1810. 06543, 2018.
- [15] M. Wortsman, K. Ehsani, M. Rastegari, A. Farhadi and R. Mottaghi, "Learning to Learn How to Learn: Self-adaptive Visual Navigation Using Meta-Learning," *In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2019) pp. 6750–6759.
- [16] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE Trans. Patt. Anal.* **24**(2), 237–267 (2002).
- [17] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.* **4**(2), 100–107 (1968).
- [18] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.* **30**(7), 846–894 (2011).
- [19] A. Kattapur and B. Purushotaman, "Roboplanner: A pragmatic task planning framework for autonomous robots," *Cognit. Comput. Syst.* **2**(1), 12–22 (2020).
- [20] J. L. Krichmar, T. Hwu, X. Zou and T. Hylton, "Advantage of prediction and mental imagery for goal-directed behaviour in agents and robots," *Cognit. Comput. Syst.* **1**(1), 12–19 (2019).
- [21] Y. Liang, B. Chen and S. Song, Sscnav: Confidence-aware semantic scene completion for visual semantic navigation, arXiv preprint arXiv:2012.04512 (2020).
- [22] D. S. Chaplot, D. P. Gandhi, A. Gupta and R. R. Salakhutdinov, "Object goal navigation using goal-oriented semantic exploration," *Adv. Neur. Inform. Process. Syst.* **33**, 4247–4258 (2020).

- [23] S. Tan, G. Di, H. Liu, X. Zhang and F. Sun, “Embodied scene description,” *Auton. Robot.* **46**(1), 21–43 (2022).
- [24] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh and D. Batra, “Habitat: A Platform for Embodied Ai Research,” **In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*** (2019) pp. 9339–9347.
- [25] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng and Y. Zhang, Matterport3d: learning from RGB-D data in indoor environments. arXiv preprint arXiv: 1709. 06158 (2017).
- [26] V. Cartillier, Z. Ren, N. Jain, S. Lee, I. Essa and D. Batra, Semantic mapnet: building allocentric semanticmaps and representations from egocentric views, arXiv preprint arXiv:2010.01191 (2020).
- [27] X. Liu, G. Di, H. Liu and F. Sun, “Multi-agent embodied visual semantic navigation with scene prior knowledge,” *IEEE Robot. Automat. Lett.* **7**(2), 3154–3161 (2022).
- [28] L. Xinzhu, L. Xinghang, G. Di, L. Huaping and S. Fuchun, Embodied multi-agent task planning from ambiguous instruction (2022).
- [29] X. Li, H. Liu, J. Zhou and F. C. Sun, “Learning cross-modal visual-tactile representation using ensembled generative adversarial networks,” *Cognit. Comput. Syst.* **1**(2), 40–44 (2019).
- [30] S. Tan, W. Xiang, H. Liu, G. Di and F. Sun, “Multi-agent Embodied Question Answering in Interactive Environments,” **In: *European Conference on Computer Vision*** (Springer, 2020) pp. 663–678.
- [31] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei and A. Farhadi, “Target-Driven Visual Navigation in Indoor Scenes Using Deep Reinforcement Learning,” **In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*** (IEEE, 2017) pp. 3357–3364.
- [32] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid and J. Davidson, “Visual Representations for Semantic Target Driven Navigation,” **In: *2019 International Conference on Robotics and Automation (ICRA)*** (IEEE, 2019) pp. 8846–8852.
- [33] J. Redmon and A. Farhadi, Yolov3: An incremental improvement. arXiv preprint arXiv: 1804. 02767 (2018).
- [34] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein and L. Fei-Fei, “Image Retrieval Using Scene Graphs,” **In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*** (2015) pp. 3668–3678.
- [35] S. Lenser and M. Veloso, “Visual Sonar: Fast Obstacle Avoidance Using Monocular Vision,” **In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No. 03CH37453)***, vol. 1, (IEEE,2003) pp. 886–891.
- [36] X. Li, G. Di, H. Liu and F. Sun, “Embodied Semantic Scene Graph Generation,” **In: *Conference on Robot Learning*** (PMLR, 2022) pp. 1585–1594.
- [37] H. Zhang, Z. Kyaw, S.-F. Chang and T.-S. Chua, “Visual Translation Embedding Network for Visual Relation Detection,” **In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*** (2017) pp. 5532–5540.
- [38] Q. Wu, C. Shen, P. Wang, A. Dick and A. Van Den Hengel, “Image captioning and visual question answering based on attributes and external knowledge,” *IEEE Trans. Patt. Anal.* **40**(6), 1367–1381 (2017).
- [39] Z. Zeng, A. Röfer and O. C. Jenkins, “Semantic Linking Maps for Active Visual Object Search,” **In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*** (IEEE, 2020) pp. 1984–1990.
- [40] D. Meyer-Delius, J. M. Hess, G. Grisetti and W. Burgard, “Temporary Maps for Robust Localization in Semi-Static Environments,” **In: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*** (2010).
- [41] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein and L. Fei-Fei, “Visual genome: Connecting language and vision using crowdsourced dense image annotations,” *Int. J. Comput. Vis.* **123**(1), 32–73 (2017).
- [42] K. He, G. Gkioxari, P. Dollár and R. Girshick, “Mask R-CNN,” **In: *Proceedings of the IEEE International Conference on Computer Vision*** (2017) pp. 2961–2969.
- [43] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, A. Farhadi, Ai2-thor: an interactive 3D environment for visual AI. arXiv preprint arXiv:1712.05474 (2017).
- [44] C. Gan, Y. Zhang, J. Wu, B. Gong and J. B. Tenenbaum, “Look, Listen, and Act: Towards Audio-Visual Embodied Navigation,” **In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*** (IEEE, 2020) pp. 9701–9707.