Efficient Object Search Through Probability-Based Viewpoint Selection

Alejandra C. Hernandez, Erik Derner, Clara Gomez, Ramon Barber, and Robert Babuška

Presented at the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), October 25-29, 2020, Las Vegas, NV, USA (Virtual)

http://ras.papercept.net/images/temp/IROS/files/1402.pdf

2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Efficient Object Search Through Probability-Based Viewpoint Selection

Alejandra C. Hernandez[1], Erik Derner[2,3], Clara Gomez[1], Ramon Barber[1], and Robert Babuška[2,4]

***Abstract*— The ability to search for objects is a precondition for various robotic tasks. In this paper, we address the problem of finding objects in partially known indoor environments. Using the knowledge of the floor plan and the mapped objects, we consider object–object and object–room co-occurrences as prior information for identifying promising locations where an unmapped object can be present. We propose an efficient search strategy that determines the best pose of the robot based on the analysis of the candidate locations. We optimize the probability of finding the target object and the distance travelled through a cost function.**

To evaluate our method, several experiments in simulated and real-world environments were performed. The results show that the robot successfully finds the target object in the environment while covering only a small portion of the search space. The real-world experiments with the TurtleBot 2 mobile robot validate the proposed approach and demonstrate that the method performs well also in real environments.

***Index Terms*— Object search, service robots, viewpoint selection, semantic scene understanding.**

I. Introduction

Object search in human-inhabited environments has drawn a lot of research attention recently. Many tasks in service robotics, such as fetch-and-carry, require the robot to autonomously search for an object in a dynamic environment. A map of the environment typically includes certain objects such as cabinets and tables, while the locations of other objects such as cups and laptops are often unknown and can be easily changed.

In this paper, we focus on the task of using mobile robots to find objects in indoor environments. Given a 2D map of the environment including walls and mapped objects, the objective is to identify the most likely places from which the robot can see the target object. Semantic information is included in the map: rooms are labeled by room types, such as bathroom or bedroom, and mapped objects are assigned to classes, e.g., table, chair, or laptop. The search can benefit from putting together the semantic information and prior knowledge, which is given as probabilities of finding an object of a given class in a particular room type or near an object of another class. For example, a cup is likely to be present in the kitchen and next to other kitchenware or on the table.

[1] Robotics Lab, Carlos III University of Madrid, Spain. {alejhern, clgomezb, rbarber}@ing.uc3m.es
[2] Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Czech Republic. {erik.derner, robert.babuska}@cvut.cz
[3] Department of Control Engineering, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic.
[4] Cognitive Robotics, Delft University of Technology, The Netherlands.

The challenge we address in this work is to find an unmapped object in a partially known environment using the prior knowledge. We propose a novel method for object search guided by co-occurrence probabilities to first determine the most likely room where the object can be found. This is further refined by choosing specific places in the room to be visited, so that the distance travelled is minimized. The core of the search process is in the analysis and selection of the best locations called *viewpoints*.

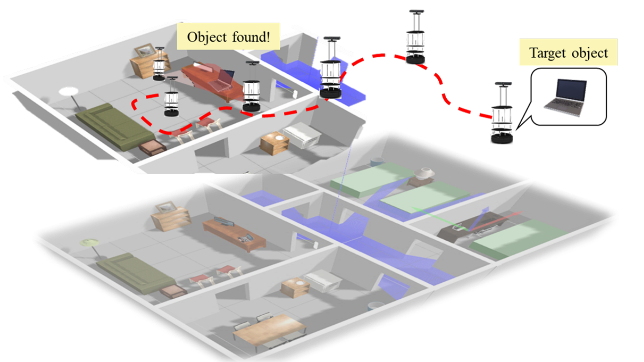


Fig. 1. An illustration of the proposed object search method. The best viewpoint is determined using the semantic information and viewpoint analysis to optimize the search strategy.

An overview of the proposed method is shown in Fig. 1. Due to the size of the room and the limitations of the robot sensors, the object sought usually cannot be spotted at the moment of entering the room. An exhaustive exploration of the whole room would be time and energy consuming. Hence, we employ a method that generates viewpoints from which the target object is likely to be seen. Choosing the right viewpoint has a significant impact on the robot performance. That's why promising robot orientations are identified as well to further speed up the search. If the object is not found, probabilities are updated for the next best viewpoint analysis. The main contributions of this paper are:

- Development of a novel object search method combining semantic information about the environment with prior co-occurrence probabilities.
- Probabilistic analysis and selection of best viewpoints to efficiently find objects in partially known environments.
- The use of multi-objective optimization to maximize the probability of finding the target object and to minimize the distance travelled.
- Implementation of the object search framework on a mobile robot and its evaluation in real-world experiments.

The remainder of the paper is organized as follows. In Section II, we review the related work in object search and viewpoint selection. Section III introduces a general overview of the proposed method, while Sections IV, V, and VI explain its individual parts in detail. Experimental results are presented in Section VII. Section VIII concludes the paper and outlines the directions for future work.

II. Related Work

Object search methods can be classified into direct and indirect methods. Direct methods are based on searching for the target object, whereas indirect approaches search for an intermediate object first. The intermediate object is supposed to be related to the object sought, easier to be found than the object sought, and the two objects are expected to be located close to each other. Once the robot finds the intermediate object, it searches for the target object in the intermediate object's vicinity.

Indirect methods, such as [1] and [2], use an object search strategy based on qualitative spatial relations (QSR). In [2], the probability of the target object is defined through relations such as left to, in front of, etc., which are set between the sought object and other objects. During the search process, supporting surfaces (such as tables) are found and after that, the target object is sought. The selection of viewpoints is based on the number of voxels that form a part of a supporting plane. Authors in [3] use object–object relations to find the target object via a chain of intermediate objects. In [4] and [5], the search process is based on spatial relations between objects of the environment to obtain the probability distribution of the target object in a room. In [6], the search process starts by looking for environmental objects (defined as objects with less movability) and then looking for the dynamic target object. Although the purpose of indirect strategies is to reduce the search space in order to decrease the computational complexity, these methods have some drawbacks. Searching for an intermediate object can be sometimes as difficult as searching for the target object itself. In addition, relations between intermediate and target objects are not always available.

For these reasons, some authors focus on direct methods, approaching the search problem as an exploration problem. In [7], a search for specific objects is performed directly in large and unknown environments. Relations between objects and rooms are defined to build a probabilistic model of the environment, revealing the promising areas where the target object can be located. As the exploration is a part of the problem presented, a cost function is used to decide where to explore and search at the same time. In [8], a reactive search strategy for unknown environments is presented, which predicts promising directions for the robot based on decision trees. In [9] and [10], an unknown environment is also considered. The proposed methods only consider as prior information the assumption that the target object is located on one of the tables of the environment. Also in [9], multiple cost functions are evaluated to reduce not only the distance travelled, but also the number of actions performed by the robot. In [11], the search problem is modeled as a Partially Observable Markov Decision Process (POMDP). Only object–room relations are used and are encoded in a belief map.

As neural networks have become widely used in the past years, the object search can be solved through deep learning techniques. Works such as [12] and [13] present a framework based on deep reinforcement learning. In [12], the model learns policies considering only visual inputs (sequence of RGB images) without any contextual information. Given an image of the target object, the robot decides the best actions to find the object through a reward function based on the size of the bounding box of the target object. In [13], an object proposal method is applied to create a hypothesis about the target object location. The environment is represented as a voxel grid and in each cell, the probabilities of the target object being there are updated. The main drawback of these approaches based on deep learning is that they require a highly time-consuming training. Computational complexity also increases if the object detector needs to be continuously running.

In the sequel, we focus on works addressing viewpoint generation and selection. In [14], a set of poses reachable by the robot are generated. Then, a visibility analysis of the object candidate from each viewpoint is performed. The analysis considers the occlusions and the visible object features to define the best viewpoint. In other works, such as [1], [2], [7], and [10], the viewpoints are randomly generated in the reachable space. Every viewpoint is computed until a covered area threshold is reached. In [1], the selection of the best pose is based on the maximum probability of finding the target object. In [2], the full area covered by each viewpoint is evaluated until the object is found. In [10], the target object is expected to be placed over a flat surface and the selection of the best viewpoint considers the highest portion of the room covered by the viewpoint. In some of the aforementioned methods, a fixed number of viewpoints is generated in advance and the selection of the best location is made at random. In other cases, the pose selection strategy deems only a single objective, e.g., the highest visibility of the object, or the minimum time or distance travelled.

To overcome these shortcomings, we propose a direct search method that combines the semantic information included in the map and prior knowledge about the object–object and object–room relations. It uses co-occurrence probabilities to infer the best viewpoints from which the target object could be seen. New positions of the viewpoints are generated in case the object is not found at the first attempt. Two criteria, the probability of finding the target object and the distance travelled, are optimized simultaneously through a cost function. The method is computationally lightweight and it can be run on a low-cost mobile robot.

III. Object Search Method Overview

In this section, a conceptual overview of the proposed object search method is given. The implementation details about the method follow in Sections IV–VI.

A schematic overview of the method is shown in Fig. 2. There are three inputs of the algorithm. The first one is a 2D floor plan previously built by the robot that includes the room layouts and the information about the position and class of the mapped objects. The second one is a probabilistic representation of the object–room and object–object co-occurrences. Finally, the method is given the class of the target object to be found, e.g., a cup. Even though the method can be used to search also for mapped objects, we focus mainly on the more challenging case when the location of the object sought is not known in advance.

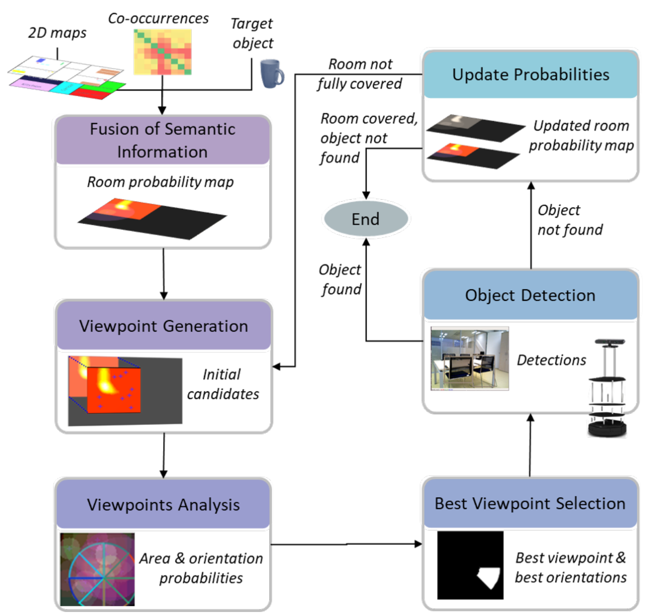


Fig. 2. The proposed object search method. First, the semantic information and prior knowledge is fused. Then, an exhaustive analysis of the generated candidate viewpoints is performed to obtain an optimal strategy to search for the target object. If the object is not found, the probabilities are updated and a set of new candidate viewpoints is generated.

When the search process starts, the semantic information included in the map and the prior co-occurrence probabilities are fused to create an initial probability map. Note that for a more intuitive understanding, top-view maps of the environment as well as the probability maps calculated by the method can be represented as images, where a pixel corresponds to a real-world square with a fixed size (e.g., $0.1 \times 0.1\,\mathrm{m}$). If the target room to search for the object is not explicitly specified by the user, we start in the room with the highest probability of containing the target object. This information can be inferred from the object–room co-occurrences. The probability map of the room, which can be visualized as a heat map, encodes the promising areas of the room where the target object is likely to be present.

In the second step, a set of random candidate viewpoints is generated with the goal of maximizing the room coverage. Then, in the third step, the viewpoints are analyzed, taking into account the visibility model of the camera. It provides the information about the minimum and maximum distances for the perception. As a result, the probability of finding the target object in the area covered by each viewpoint is calculated. The area covered by each viewpoint is afterwards divided into a given number of segments, according to the horizontal field of view (FOV) of the camera. The probability of the target object being in each segment of the viewpoint is calculated. The fourth step consists in selecting the best viewpoint and, in turn, the best segment. The best viewpoint selection maximizes the probability of finding the target object while minimizing the distance travelled by the robot. To achieve this, we have designed a utility function and automatically tuned its parameters.

Finally, the best segment is sent to the robot navigation system. When the robot reaches the desired pose, an object detector based on deep learning is executed. If the object is found, the process ends. If not, the robot moves to the next best segment and attempts to detect the object again. If there are no more candidate segments available and the target object has not been found yet, the probabilities in the explored area are updated and then, the process returns to the second step to determine a new best viewpoint. Also, the covered area is calculated and used as a termination condition. When the room is considered covered, the process ends. This may indicate that the target object is not in that room and the search process can start again in other rooms.

IV. Initial Probabilities Assignment

An efficient object search approach is fundamental to reduce the search space and thus minimize the length of the robot trajectory, decrease power consumption and computational costs. A possible way to achieve this is by using prior knowledge about the environment [15]. Prior information available to the robot has several sources: a 2D floor plan of the environment, including the type l_j of each room r_j (e.g., a kitchen), and the information about the previously seen objects o_s in the room. The latter includes the object class c_s, the detection confidence $p(o_s)$ given by the object detector, and the space occupied by the object in the map. This space is represented as a function $g(o_s, P_{x,y}) \rightarrow \{0,1\}$, which is equal to one if the object o_s is present in the pixel $P_{x,y}$ of the top-view floor plan and zero otherwise.

The last component of the prior knowledge are the co-occurrence probabilities. We use two publicly available datasets to obtain the co-occurrence probability values. The probabilities for the object–room co-occurrences, denoted $p(o_s|r_j)$, have been extracted from the NYU-Depth V2 dataset [16]. Object–object co-occurrences $p(o_s|o_{s'})$ for all pairs of object classes c_s, $c_{s'}$ were calculated using the COCO dataset [17].

Using the co-occurrence probabilities between the room type and the target object class, the most likely room r_j^* is selected:

$$r_j^* = \operatorname*{argmax}_{j \in \{1,\dots,m\}} p(o_\tau|r_j)\,, \tag{1}$$

where m is the number of rooms in the environment. The robot will explore the room r_j^* first.

Next, we generate the room probability map, which identifies the most likely areas where the target object o_τ can be located. The probability of finding the target object in each pixel of the room is defined as:

$$p(o_\tau|P_{x,y}) = \\ = \begin{cases} \sum_{s=1}^{n} f(o_s, P_{x,y}) p(o_\tau|o_s) & \text{if } \sum_{s=1}^{n} f(o_s, P_{x,y}) > 0\,, \\ p(o_\tau|r_j^*) & \text{otherwise}\,, \end{cases} \quad (2)$$

where $f(o_s|P_{x,y}) = g(o_s, P_{x,y})p(o_s)$. The number of all mapped objects in the environment is n. The values of $p(o_\tau|P_{x,y})$ are normalized. Then, a Gaussian filter is applied around each detected object. As pixels are further from the detected object, they receive a gradually decreasing probability. Fig. 3 shows the room probability map for four target objects visualized as a heat map. The brighter the area, the more likely it is to encounter the target object.

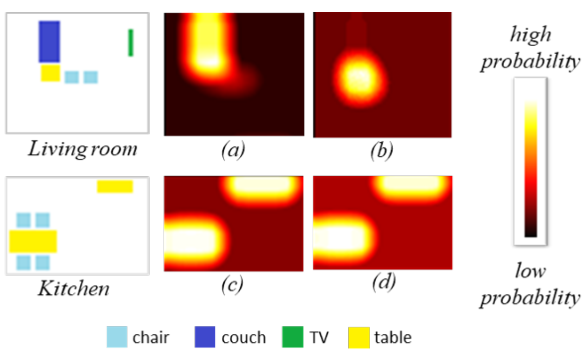


Fig. 3. Room probability maps of four target objects: (a) *laptop*, (b) *cup*, (c) *bowl*, and (d) *bottle*. Darker areas represent lower probabilities, whereas lighter areas indicate promising zones where the target object can be located.

Once the initial room probability map is calculated, the next step is the generation of a set of candidate locations on the map, and the selection of the best one in order to execute a detector that locates the target object.

V. Viewpoint Generation and Selection

The viewpoint selection is one of the most important steps in the search strategy. Choosing the right viewpoint has a significant impact on the robot performance. In this section, a detailed explanation about the process of generating, analyzing and selecting the best viewpoint is presented.

A. Candidate Viewpoints Generation

In this step, an initial set of N candidate viewpoints is generated in the room r_j^*. Each candidate viewpoint $V_{x,y}$ represents a position in the map. This process considers the hardware limitations of the camera: the minimum and maximum distance d_{min} and d_{max}, from which the sensor can perceive the objects, and the horizontal FOV of the sensor, see Fig. 4.

To cover the room with candidate viewpoints, we follow an iterative process. The process starts by placing an initial viewpoint at a random position in the free space $\mathcal{F}(r_j^*)$ of the room r_j^*. The free space is defined as:

$$\mathcal{F}(r_j) = \left\{ P_{x,y} \;\middle|\; \sum_{s=1}^{n} g(o_s, P_{x,y}) = 0 \right\}. \quad (3)$$

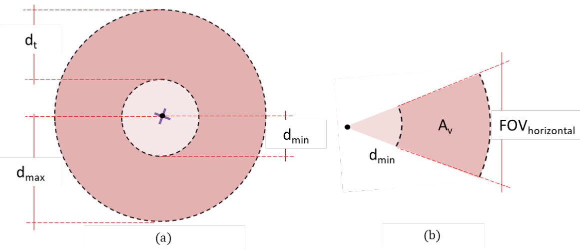


Fig. 4. Visibility model of the camera. In (a), d_{min} and d_{max} represent the limits of the coverage area of each viewpoint. In (b), A_v denotes the search area covered by a particular segment of a viewpoint.

Next viewpoints are randomly generated in the free space while keeping a minimum Euclidean distance $d_{\min}$ from the previously generated viewpoints.

Each candidate viewpoint $V_{x,y}$ covers a set of pixels in r_j. The union of all pixels inside the detection zone determined by d_t, see Fig. 4, represents the area $\mathcal{A}(V_{x,y})$ covered by the viewpoint $V_{x,y}$. The total room area $\mathcal{R}(r_j)$ is defined as the union of all pixels $P_{x,y}$ belonging to that room. The coverage $C(r_j)$ for the room r_j is then calculated as:

$$C(r_j) = \frac{\left|\bigcup_{V_{x,y}} \mathcal{A}(V_{x,y})\right|}{|\mathcal{R}(r_j)|}. \quad (4)$$

The coverage $C(r_j)$ is updated after the insertion of each viewpoint. A threshold ρ determines when the room is assumed to be sufficiently covered. The viewpoint generation process terminates once $C(r_j) > \rho$.

B. Viewpoints Analysis

Each candidate viewpoint is analyzed to determine the most promising location from which the target object can be seen. First, the probability of finding the target object inside the area covered by each viewpoint is calculated:

$$p(o_\tau|V_{x,y}) = \frac{\sum_{P_{x,y} \in \mathcal{A}(V_{x,y})} p(o_\tau|P_{x,y})}{|\mathcal{A}(V_{x,y})|}. \quad (5)$$

This measure represents how good a certain viewpoint $V_{x,y}$ is for the search task.

Next, we analyze the segments of each viewpoint. According to the visibility model of the camera, the area covered by a viewpoint $V_{x,y}$ can be divided into Q segments $\theta_{x,y,q}$. In this work, $Q = 8$ segments have been used, given the $\mathrm{FOV}_{\mathrm{horizontal}}$ of the camera 58°. Note that the segments are partially overlapping, which is desirable for improved performance as the chances of detecting objects that are in the limit between two segments are increased.

The probability of finding the target object o_τ in a given segment $\theta_{x,y,q}$ of a viewpoint $V_{x,y}$ is defined as:

$$p(o_\tau|\theta_{x,y,q}) = \frac{\sum_{P_{x,y} \in \mathcal{S}(\theta_{x,y,q})} p(o_\tau|P_{x,y})}{|\mathcal{S}(\theta_{x,y,q})|}. \quad (6)$$

Analogously to viewpoint probability calculation (5), $\mathcal{S}(\theta_{x,y,q})$ denotes the set of pixels that belong to the segment $\theta_{x,y,q}$.

C. Best Viewpoint Selection

Using the information available at this point, the best viewpoint can be selected. To maximize the probability of finding the target object while minimizing the distance travelled, we adapt the utility function $U_{x,y}$ from [9]. It is calculated for a given viewpoint $V_{x,y}$ as follows:

$$U_{x,y} = p(o_\tau|V_{x,y})\left(1 + \frac{\beta}{d(V_{x,y})}\right), \tag{7}$$

where $d(V_{x,y})$ is the distance between the current (starting) pose of the robot and the viewpoint $V_{x,y}$.

In [9], the parameter β is set to 1 to relax the requirement of distance minimization. In our work, we have decided to apply the Levenberg-Marquardt algorithm [18] [19] in order to find the optimal β value. The algorithm was provided with a set of 100 measured data points. To obtain this data, the search method has been applied with different β values and looking for different o_τ. Using the ground truth location of the target object, the objective function is minimizing the distance travelled until the target object is found.

The best viewpoint $V^*_{x,y}$ is selected as follows:

$$V^*_{x,y} = \underset{V_{x,y}}{\operatorname{argmax}}\, U_{x,y}\,. \tag{8}$$

The viewpoint $V^*_{x,y}$ specifies the most likely position in the map from which the robot is expected to be able to detect the object sought. To specify also the orientation of the robot, a set of best segments is determined, given an empirically chosen threshold σ:

$$\theta^*_{x,y,q} \in \{\theta_{x,y,q} \mid p(o_\tau|\theta_{x,y,q}) > \sigma\}\,. \tag{9}$$

This way, the segments with lower probabilities are discarded, making the search process more efficient. The robot visits the best segments, as for mobile robots it is typically faster to turn on the spot. As soon as the best segment is determined, it is sent to the robot navigation system. Once the robot reaches it, an object detector is run to identify the objects present in the segment.

VI. Probability Map Update

In case the processing of the best viewpoint has terminated and the target object has not been found, the room probability map and the percentage of the room's covered area should be updated for a new iteration. This update allows for deciding whether to continue the search process in the current room or whether other actions, such as searching in another room, should be taken. If the object detector reports that o_τ is not found for a given viewpoint, the next viewpoint generation considers the areas already explored by the robot and the detection results in order to avoid looking at these parts again. In some cases, areas of the viewpoints may overlap. Because of this, the probability of finding the target object in the pixels belonging to an already explored viewpoint has to be reduced. We introduce a discrete time variable t and we define how $p(o_\tau|P_{x,y})_t$ decreases in a new iteration:

$$p(o_\tau|P_{x,y})_{t+1} = 0.5\, p(o_\tau|P_{x,y})_t\,. \tag{10}$$

Through (10) the room probability map is updated in each pixel. As a termination condition, the covered area of the room $C(r_j)$ is updated according to (4). In this work, the threshold ρ has been empirically set to 0.85 to declare when the room has been completely covered. It is not always possible to cover 100 % of the environment, in particular in cases when the free space $\mathcal{F}(r_j)$ in the room r_j is limited by many obstacles. If the threshold is not reached, the search process begins a new iteration. This time, the new set of viewpoints is generated taking into account the current pose of the robot and considering the updated probabilities after an unsuccessful detection.

VII. Experiments

To evaluate the validity and efficiency of our approach, the proposed search strategy has been tested in both simulated and real environments.

A. Experimental Setup

We have selected the widely-used mobile robotic platform TurtleBot 2 to perform the real-world experiments. It is equipped with an ASUS Xtion Pro Live camera and a Hokuyo URG-04LX-UG01 laser scanner. For object detection, we have implemented an object recognition framework based on deep learning. The model architecture is based on ResNet-101[20] and it has been trained with the COCO dataset [17]. The components of the framework are integrated through the middleware ROS. To build a map of the environment, the ROS gmapping package has been used. For path planning, the Adaptive Monte-Carlo Localization (AMCL) algorithm has been implemented.

B. Simulated Experiments

We set up a simulated environment inspired by real-world homes that includes common objects and different types of rooms. The 12 m × 8 m environment was created using Gazebo simulator (Fig. 5) and consists of a typical house with 6 different rooms: a bedroom, a child's room, a corridor, a bathroom, a living room, and a kitchen. We have tested the method with four target objects: a laptop, a cup, a bowl, and a TV.

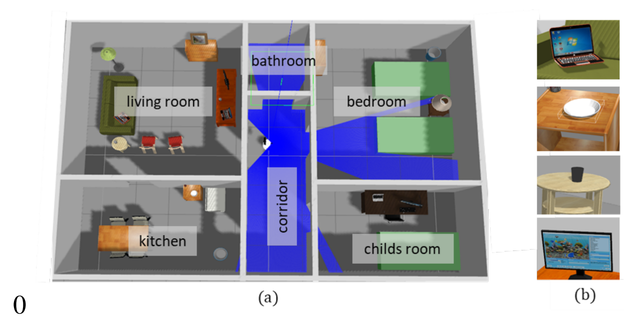


Fig. 5. The simulated home environment used in the experiments. (a) A home environment with six rooms. (b) The target objects selected for the experiments.

We explain the working of the method on an example of searching for a cup. Fig. 6 illustrates the steps to select the best viewpoint during the search process. At first, the robot has determined the room that most likely contains the object sought and is standing at its entrance. The prior knowledge about the object–object co-occurrences and the information about the mapped objects are fused into a room probability map. The map provides the initial prediction of the promising areas where the target object can be located. In Fig. 6(a), the lighter areas are the most probable locations for the target object. In this example, the cup is on a small dining table in the living room. Then, a set of random viewpoints is iteratively generated considering the distance between the viewpoints and the covered area of the room (b). After that, the viewpoint analysis begins. The probability of finding the target object in the entire viewpoint area is calculated (c). Similarly, the probabilities of finding the target object in each segment of the viewpoint are determined (d).

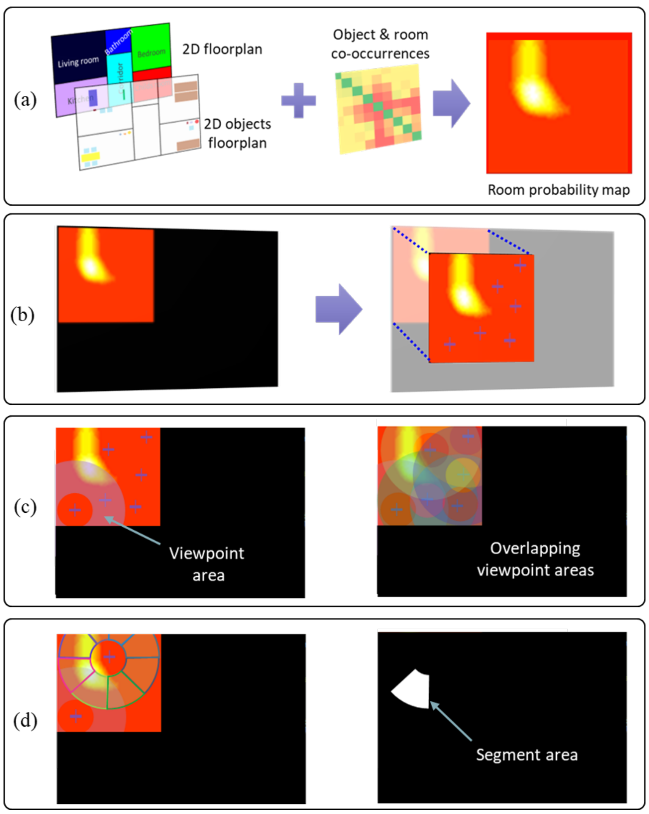


Fig. 6. The proposed search strategy operating in a home simulated environment. The most probable room is the living room. (a) The prior information is fused to generate a room probability map. (b) The random viewpoints are generated inside the room. In (c) the analysis of the all viewpoint areas is conducted. Finally, (d) shows the best candidate segment.

Through the utility function maximization, the best viewpoint is chosen. Next, the set of best segments is determined. Then, the best viewpoint is sent to the robot navigation system. The robot moves to the given viewpoint and adjusts its orientation to visit the best segment. If the object is not found, the probabilities in the explored area are updated. Fig. 7 illustrates the subtasks of the robot during the search process. In (a) and (b), the target object, the best viewpoint and the best candidate segments are shown. In (c), a representation of the covered area of the room after the processing of the first best segment can be observed. In this case, the covered area was 10.3 %. In (d), the results of the object detector are shown. Table I shows the results of the proposed method. The starting robot pose is at the entrance to the room. We calculate the covered area of the room, the time spent and the total distance travelled by the robot until it finds the object. Total viewpoints visited and the number of segments explored during the search process are also counted. We have repeated the search for each of the four target objects three times.

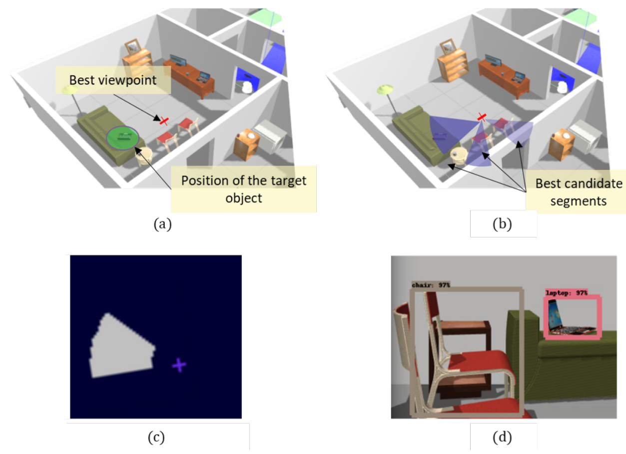


Fig. 7. Illustration of the subtasks in the search process. (a), (b) the best viewpoint and the best candidate segments are chosen; (c) the covered area of the room after exploring the best segment; (d) object detections.

TABLE I
EVALUATION OF THE PROPOSED SEARCH STRATEGY IN A SIMULATED HOME ENVIRONMENT.

Target object	Time (s)	Distance traveled (m)	Covered area	Total viewpoints	Total segments
laptop	120.66	1.27	0.21	1	2
laptop	63.43	1.18	0.10	1	1
laptop	82.20	1.11	0.10	1	1
cup	270.50	2.44	0.30	2	3
cup	245.50	2.04	0.16	1	2
cup	451.22	2.75	0.29	3	3
tv	63.01	0.56	0.19	1	1
tv	91.55	0.75	0.16	1	1
tv	118.24	1.24	0.21	2	2
bowl	152.59	0.92	0.19	1	2
bowl	135.27	1.02	0.10	1	1
bowl	186.42	1.94	0.25	2	2
Avg.	157.87	1.44	0.19	1.42	1.75

The results show that the method limits the search area through an analysis of the viewpoints and only the most promising areas are considering for searching. On average, the search process takes 157.87 seconds and the covered area of the room is 19 %.

C. Real-World Experiments

The experiments were carried out in a living room of 14 m $\times$ 4 m. The room was first mapped and the information about some objects in the room such as tables, chairs, and sofas was therefore available to the robot. Fig. 8 shows the environment and the target objects selected for the experiments. The robot has to search for two different objects: a cup and a laptop, each of them placed at two different locations in each execution. In Fig. 9, the search for a cup is illustrated: (a) shows the map of the environment, (b) shows the room probability map for the target object, and (c) the covered area of the room.

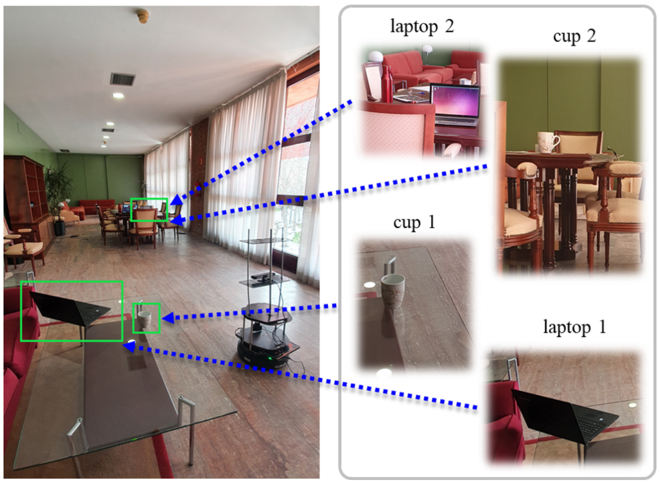
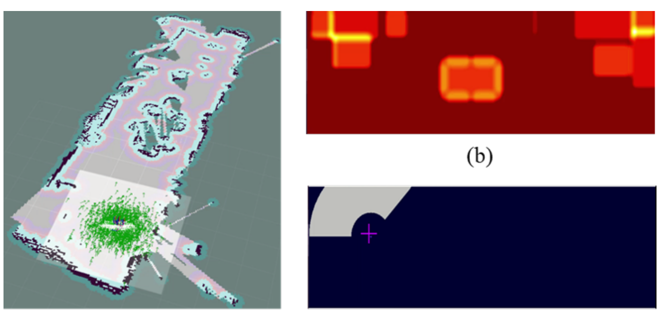


Fig. 8. The real environment used in our experiments. The robot is asked to find a cup and a laptop in different locations of the room.



(b)

(a) (c)

Fig. 9. Execution of proposed search strategy. (a) The map of the environment and the execution of the path planning. (b) The room probability map built for the target object cup. (c) The covered area of the room (grey) after the evaluation of the best viewpoint.

Table II show the results of evaluating the proposed search strategy in the real-world environment.

TABLE II
EVALUATION OF THE PROPOSED SEARCH STRATEGY IN THE REAL-WORLD ENVIRONMENT.

Target object (o_τ)	Time (s)	Distance traveled (m)	Covered area	Total viewpoints	Total segments
cup 1	275.34	0.95	0.18	2	2
cup 2	140.71	1.94	0.19	1	2
laptop 1	139.11	0.36	0.10	1	1
laptop 2	383.44	1.05	0.20	2	3
Avg.	234.65	1.08	0.17	1.5	2

In some cases, the detector fails to recognize the object, although the target object is within the field of view of the camera. This forces the robot to explore another segment or a new viewpoint from which the object detector is able to identify the object. Despite this, the results demonstrate the feasibility and the efficiency of the proposed method for the task of searching for objects in real scenarios.

D. Comparison with Other Approaches

1) Quantitative Comparison: To the best of the authors' knowledge, there is no dataset publicly available that would be specifically designed for comparison of object search methods in human-inhabited environments. To obtain valid results in a comparison, the compared methods have to be evaluated under the same conditions of the environment, target objects and the position of the objects. To overcome this issue, we have built a baseline approach to object search based on random viewpoint selection. The method has been executed 12 times in the same simulated environment and looking for the same target objects as in Fig. 5. This method does not take into account any prior information to select the best viewpoint. Table III shows the results of the baseline method compared to our proposed search strategy.

TABLE III
COMPARISON BETWEEN THE PROPOSED SEARCH STRATEGY AND THE BASELINE SEARCH METHOD.

Search method	Avg. time (s)	Avg. distance (m)	Avg. coverage	Avg. viewpoints	Avg. segments
random	395.31	4.41	0.28	2.16	9.09
our approach	157.87	1.44	0.19	1.42	1.75

The results show that, the random strategy is substantially less efficient than the proposed method. On average, finding an object through the random selection of viewpoints takes 395.31 seconds and the average distance travelled is 4.41 meters. On the contrary, with our search strategy, the task takes an average of 157.87 seconds with a distance travelled of 1.44 meters. Our approach allows to determine the best viewpoint and the best segments to search for an object in a more efficient way. The number of viewpoints explored and the covered area are also lower than those obtained with the random method. An appropriate viewpoint selection reduces the search space and the robot trajectory, yielding better results in less time.

2) Qualitative Comparison: In [2] and [10], the authors associate the target object with supporting planes to reduce the search space. The search space is limited to objects that can be found, for example, on tables. If the objects do not have associated supporting planes or if the tables are not identified, then, the target object can not be found. In our work, we incorporate object–room and object–object co-occurrences to generate promising areas where a target object can be, regardless of whether they are in supporting planes or, for instance, on a chair or a sofa. In [6], an intermediate object with a strong relationship with the target object is found first to guide the search process. The problem is that sometimes the detection of the intermediate objects can be as difficult as identifying the target object. In our approach, the robot searches for the object directly, based on a probability map of the room. Our efforts are focused on an analysis of the viewpoints to determine the most promising poses. In addition, the room probability map is updated after visiting a viewpoint if the object is not found, which influences the selection of the new best viewpoint.

As for the viewpoint generation, in [2] the number of random viewpoints is set to 20. Similarly, in [10], the fixed number of viewpoints is 1000, assuming that the area of the rooms is always the same. In our method, we build a set of random viewpoints, in which the size of the set depends on the covered area of the room. Therefore, our method ensures that the room is completely covered and that the amount of points is sufficient to carry out the search task. Another contribution of our paper is the optimization of the utility function. Our utility function is adapted from [9], where the authors set the β value to 1 to relax the distance minimization. In contrast to that, we use an optimization algorithm to find the β value to satisfy not only the maximum probability of finding the target object, but also the minimum distance travelled until the object is found.

VIII. Conclusions and Future Work

In this work, we have presented a novel search strategy to efficiently find unmapped objects in partially known environments based on prior knowledge and the analysis of candidate locations. Prior knowledge in the form of object–object and object–room co-occurrences has been employed to build a probability map with the most promising locations of the target object. The core of the method is the selection of the best viewpoint through an analysis of the probabilities of finding the target object in the area covered by the viewpoint. In addition, the best segments within the viewpoint are determined to further speed up the search and to partially address the occlusions. An optimized cost function is used to maximize the probability of finding the target object while minimizing the distance travelled.

We have evaluated the method in simulated and real-world environments, demonstrating its validity and efficiency on the task of finding the target object. The quantitative and qualitative comparison has shown the advantages of the proposed method. In the future work, we plan to do an ablation study and large-scale experiments. We will also study searching for objects in long-term operation. The room probability map should be maintained in time for future searches. Other line of research would be to adapt the method to consider the user preferences during an object search task.

Acknowledgment

This research has received funding from HEROITEA: Heterogeneous Intelligent Multi-Robot Team for Assistance of Elderly People (RTI2018-095599-B-C21), funded by Spanish Ministerio de Economía y Competitividad. This work was also supported by the RoboCity2030-DIH-CM project (S2018/NMT-4331, RoboCity2030-Madrid Robotics Digital Innovation Hub), by the European Regional Development Fund under the project Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000470), and by the Grant Agency of the Czech Technical University in Prague, grant no. SGS19/174/OHK3/3T/13.

References

[1] P. Loncomilla, J. Ruiz-del Solar, and M. Saavedra, "A bayesian based methodology for indirect object search," *Journal of Intelligent & Robotic Systems*, vol. 90, no. 1-2, pp. 45–63, 2018.

[2] L. Kunze, K. K. Doreswamy, and N. Hawes, "Using qualitative spatial relations for indirect object search," in *2014 IEEE international conf. on robotics and automation (ICRA)*. IEEE, 2014, pp. 163–168.

[3] J. Elfring, S. Jansen, R. van de Molengraft, and M. Steinbuch, "Active object search exploiting probabilistic object–object relations," in *Robot Soccer World Cup*. Springer, 2013, pp. 13–24.

[4] P. Chumtong, Y. Mae, K. Ohara, T. Takubo, and T. Arai, "Object search using object co-occurrence relations derived from web content mining," *Intelligent Service Robotics*, vol. 7, no. 1, pp. 1–13, 2014.

[5] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 2818–2824.

[6] Y. Zhang, G. Tian, J. Lu, M. Zhang, and S. Zhang, "Efficient dynamic object search in home environment by mobile robot: A priori knowledge-based approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9466–9477, 2019.

[7] A. Aydemir, A. Pronobis, M. Göbelbecker, and P. Jensfelt, "Active visual object search in unknown environments using uncertain semantics," *IEEE Transac. on Robotics*, vol. 29, no. 4, pp. 986–1002, 2013.

[8] D. Joho, M. Senk, and W. Burgard, "Learning search heuristics for finding objects in structured environments," *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 319–328, 2011.

[9] K. Shubina and J. K. Tsotsos, "Visual search for an object in a 3d environment using a mobile robot," *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 535–547, 2010.

[10] R. Izquierdo-Cordova, E. F. Morales, L. E. Sucar, and R. Murrieta-Cid, "Searching objects in known environments: Empowering simple heuristic strategies," in *Robot World Cup*. Springer, 2016, pp. 380–391.

[11] C. Wang, J. Cheng, J. Wang, X. Li, and M. Q.-H. Meng, "Efficient object search with belief road map using mobile robot," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3081–3088, 2018.

[12] X. Ye, Z. Lin, H. Li, S. Zheng, and Y. Yang, "Active object perceiver: Recognition-guided policy learning for object searching on mobile robots," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 6857–6863.

[13] J. F. Schmid, M. Lauri, and S. Frintrop, "Explore, approach, and terminate: Evaluating subtasks in active visual object search based on deep reinforcement learning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 5008–5013.

[14] C. McGreavy, L. Kunze, and N. Hawes, "Next best view planning for object recognition in mobile robotics." CEUR Workshop Proc., 2017.

[15] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2168–2173.

[16] M. Brucker, M. Durner, R. Ambruş, Z. C. Márton, A. Wendt, P. Jensfelt, K. O. Arras, and R. Triebel, "Semantic labeling of indoor environments from 3d rgb maps," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1871–1878.

[17] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[18] K. Levenberg, "A method for the solution of certain non-linear problems in least squares," *Quarterly of applied mathematics*, vol. 2, no. 2, pp. 164–168, 1944.

[19] D. W. Marquardt, "An algorithm for least-squares estimation of non-linear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.