Active Visual Object Search in Unknown Environments Using Uncertain Semantics

Alper Aydemir, Andrzej Pronobis, Moritz Göbelbecker, and Patric Jensfelt

***Abstract*—In this paper, we study the problem of active visual search (AVS) in large, unknown, or partially known environments. We argue that by making use of uncertain semantics of the environment, a robot tasked with finding an object can devise efficient search strategies that can locate everyday objects at the scale of an entire building floor, which is previously unknown to the robot. To realize this, we present a probabilistic model of the search environment, which allows for prioritizing the search effort to those parts of the environment that are most promising for a specific object type. Further, we describe a method for reasoning about the unexplored part of the environment for goal-directed exploration with the purpose of object search. We demonstrate the validity of our approach by comparing it with two other search systems in terms of search trajectory length and time. First, we implement a greedy coverage-based search strategy that is found in previous work. Second, we let human participants search for objects as an alternative comparison for our method. Our results show that AVS strategies that exploit uncertain semantics of the environment are a very promising idea, and our method pushes the state-of-the-art forward in AVS.**

***Index Terms*—Active vision, semantic mapping, visual object search.**

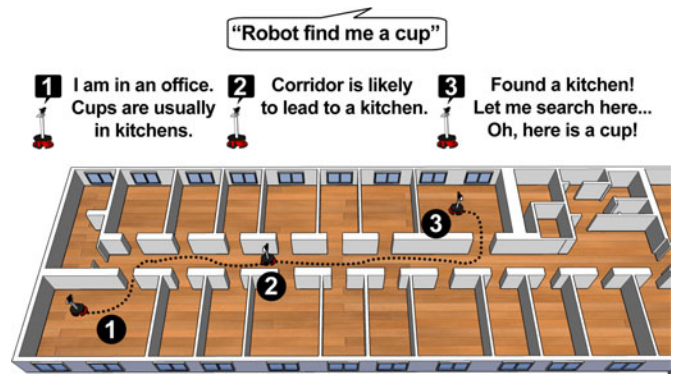


Fig. 1. Floorplan of the Center for Autonomous Systems, KTH Royal Institute of Technology. The object search scenario that is investigated in this paper is concerned with finding objects in large-scale environments that are unknown to the robot at the start of the search.

I. Introduction

THE recent advances in the fields of robot localization, mapping, navigation, and human–robot interaction brought the promise of autonomous systems, such as service and assistive robots, operating in large-scale spaces. Such systems cannot rely on the assumption that all objects relevant to the current task are already present in view. Locating objects and other points of interest become a prerequisite for many robotic tasks such as mobile manipulation or fetch and carry. To this end, this paper focuses on the robot's ability to find objects in large-scale environments using primarily visual sensing. We call this problem active visual search (AVS).

One of the most important factors that determine the overall performance of an AVS system is its search strategy. Imagine a scenario that is depicted in Fig. 1 in which a mobile courier robot is tasked with finding and fetching an object that is located somewhere in an unknown office floor. With the limited field of view of robotic sensors, it is unreasonable to assume that the robot will exhaustively examine the whole space in order to locate the object. Exhaustive search in such an environment (i.e., the entire building floor) requires capturing and analyzing millions of images, rendering the system unusable in practical applications. However, semantic information about the object and the environment can often be obtained and used to derive a more efficient strategy. In such case, a robot that is equipped with general world knowledge could use it to reason about possible locations of the object and prioritize those locations during the search process.

In most realistic scenarios, the robot needs the ability to acquire knowledge about the environment in which it operates autonomously. This adds another level of complexity to the problem of AVS. The semantics of the environment can be used in order to optimize the search strategy. However, knowledge acquisition during an AVS task in unexplored environments introduces additional cost. The problem requires deciding between exploring the environment for discovering additional spatial semantics and more places to search and searching a part of space that is already explored. In this paper, we assume a realistic scenario in which the robot is tasked with finding an object in a large-scale environment as presented in Fig. 1, with minimum cost defined as total time. The robot has no previous knowledge

Manuscript received July 11, 2012; revised December 13, 2012; accepted March 29, 2013. Date of publication April 24, 2013; date of current version August 2, 2013. This paper was recommended for publication by Associate Editor C. Torras and Editor D. Fox upon evaluation of the reviewers' comments.

A. Aydemir was with the Center for Autonomous Systems, Royal Institute of Technology, 100-44 Stockholm, Sweden. He is now with Computer Vision Group, Jet Propulsion Laboratory, National Aeronautics and Space Administration, Los Angeles, CA 91109 USA (e-mail: aydemir@kth.se).

A. Pronobis was with the Center for Autonomous Systems, Royal Institute of Technology, 100-44 Stockholm, Sweden. He is now with the Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195-2350 USA (e-mail: pronobis@csc.kth.se).

M. Göbelbecker is with the University of Freiburg, 79110 Freiburg, Germany (e-mail: goebelbe@informatik.uni-freiburg.de).

P. Jensfelt is with the Center for Autonomous Systems, KTH Royal Institute of Technology, 100 44 Stockholm, Sweden (e-mail: patric@csc.kth.se).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TRO.2013.2256686

about the environment and, instead, relies on a semantic prior about generic indoor spaces.

Our contributions in this study are twofold. First, we propose an AVS system that can efficiently locate objects, which are described in terms of their category (e.g., a cornflakes box), in an unknown large-scale indoor environment. Second, we present quantitative analysis of the performance of our method in an office environment containing 12 office rooms, a kitchen, a meeting room, and a corridor, constituting a total search space of 33 m $\times$ 12 m.

Our approach is unique in that it extensively exploits spatial semantics in novel maps for search. The system is capable of extracting semantic cues from appearance, geometry, and topology of the environment and combine it with general semantic knowledge of indoor spaces in order to reason about locations of interest. Furthermore, the system reasons not only about the already explored parts of the environment but about unexplored space as well. This allows us to direct the search toward areas that are more likely to lead to the target object or location. The search and exploration actions are governed by a planning algorithm. This permits efficient tradeoff between exploration versus knowledge exploitation and constitutes another unique feature of our approach. Finally, our method is fully probabilistic. We build on a semantic mapping algorithm that employs probabilistic graphical models [1], and use a planning technique that consists of a combination of partially observable Markov decision process (POMDPs) and a traditional continual planner for better scalability. The probabilistic framework allows us to combine an imperfect semantic prior that is obtained from Internet databases with uncertain sensing of semantic cues, and better trade between possible planning solutions.

We perform an extensive qualitative and quantitative evaluation, and compare the performance of our method with that obtained by a greedy coverage-based search strategy found in previous work and with that obtained by humans on the same search task. We demonstrate that the inclusion of semantic knowledge results in a drastic improvement of the efficiency of the search. Moreover, generated solutions become comparable with those chosen by humans in terms of the length of the robot's trajectory and time required to complete the search.

II. Analysis of the Problem

We can think of the active search problem as a decision process with a goal state and a set of actions that can take the robot from the current to the end state. Since our observations are inaccurate and stochastic, one can formulate the problem as a POMDP. In a POMDP, a probability distribution over possible world states is modeled, instead of directly representing the true state since the latter is not directly observable. This is called a belief state. The solution to a POMDP is a policy that specifies the optimal action at any belief state. The optimality comes at the price of computational complexity since the dimensionality of the POMDP belief state space is equal to the number of possible world states that may result in a computationally intractable policy computation step.

Let us first analyze the problem as in [2], assuming a fully explored search environment, and overlaying a 3-D grid on the entire map, with each grid cell holding the probability of the target being there. In that case, the number of states is equal to the number of target positions in the 3-D grid. As part of the POMDP formulation, we define one action that is moving the camera to a certain position and orientation, and performing sensing and recognition in a single cell. The observations correspond to the outcome of the recognition algorithm, i.e., the presence or absence of the target.

This leads to a computationally challenging formulation. The environment in the experimental evaluation in this paper is 33 m $\times$ 12 m, with roughly 3 m from floor to ceiling. With a cell size of, e.g., 0.1-m cube, this results in 1.2×10^6 cells. As discussed in the context of object search in [3], most general POMDP solvers can handle the number of states in the order of thousands, i.e., several orders of magnitude lower. Additionally, such an approach requires a perfectly consistent 3-D mapping framework. Relaxing the fully explored world assumption and searching in a partially explored environment necessitate a new exploration action in addition to the search action. Deciding when to search and explore and reasoning about the outcome of an exploration action only adds to the computational complexity.

A naive cell-by-cell search strategy would be extremely time consuming. A common way to reduce the search space when searching for objects is to limit the search to only occupied regions in space. In [2], the search space is limited to areas around a known table and shelf, while in [4] and more recently in [5] and [6], only regions of space where a laser scanner detects obstacles are used. In our example, such a method would reduce the number of cells from 1.2×10^6 to 8×10^4. Assuming that the camera has a 45° opening angle and it needs to be located no more than 2 m from the object for reliable detection, 3×10^4 views are required to cover the space. This corresponds to approximately 12 h of search, assuming that each view (including motion of the robot) takes 5 s, and the object is found half-way through the search. This is prohibitively slow for most realistic applications.

In order to make the search practical, we must find a heuristic that guides the search more efficiently than only using obstacles. We can get inspiration by analyzing human behavior. In a specific environment and when looking for a specific object, most humans would rely on detailed instance models, e.g., Patric's mug is likely to be located on Patric's desk. A robot assistant could gather similar statistics over time. In this paper, we assume an unknown environment. Therefore, we cannot use any specific knowledge about the objects therein. However, most humans tasked with finding a mug in an unknown environment would still not use exhaustive search. We would make use of very strong, domain unspecific priors for the location likely to contain the object. For example, experience points out that there is a strong correlation between mugs and kitchens. Instead of looking for the mug exhaustively in the floor of a building, we would first search for a kitchen. This can be generalized to exploiting spatial correlations between object categories and room categories. We argue that efficient search in human environments should make use of such knowledge, as in [7] and [8].

In a robotic system, this kind of search can be realized by employing hierarchical planning to compute a search plan. The system can first decide on regions of the search space that are promising for finding the target object before computing search strategies that take into account lower level aspects of the search such as occlusion, movement cost, and viewpoint computation. In this case, a robot would first identify the rooms to search for a mug, and then compute a search strategy for individual rooms. In this paper, we have adopted a similar approach.

Finally, it is important to keep in mind that we consider exploration of unknown space as part of the problem. That is, we want to find an object without knowing the entire extent of the environment being searched. This requires a principled way of trading exploration of the unknown against search of what is already known. In order to exploit semantic information, the system needs to be able to reason not only about the semantic spatial concepts associated with objects in the already explored part of the environment, but also about what might lay ahead. In the next sections, we will present the design of our active search system based on this analysis, first focusing on the search space modeling and pruning, and then on actions and control.

III. Related Work

Despite the recent interest in the problem of active visual object search with a mobile robot, there are no extensive surveys in the literature on this topic. However, there are notable surveys on the broad topic of active vision [9]–[11]. In particular, Chen *et al.* [9] present an excellent overview of the previous work on active vision in general, with cited contributions ranging from selecting next best views for 3-D object modeling to surveillance and inspection tasks. For this reason, we start with a comprehensive treatment of the early and current work related to the object search scenario that is considered in this paper. This allows us to show how our work fits into this body of research, and how our contributions push the state of the art in this area forward.

In a seminal paper, Bajcsy introduced the term active perception [12]. The motivation for employing an active perception strategy is that perceptual processes often *seek* for the desired percepts; in the author's words: "We do not just see, we look." In a system that employs active perception, passive sensors, such as a camera, can be utilized in an active manner by adjusting various parameters: zoom factor, depth of field, position, and orientation in the 3-D world.

Building upon Bajcsy's introduction to active perception, Tsotsos analyzed AVS [13]. Some of the advantages of an active strategy discussed are robustness to occlusions, possible increase of resolution, and use of motion to disambiguate vision-related aspects of the world such as varying illumination conditions. Tsotsos and Ye analyzed the computational complexity of the AVS problem and found it to be NP-complete [14]. A significant lesson from this analysis is that AVS strategies are more efficient than their passive equivalents. Achieving this increased efficiency requires a more complex search process. Active search strategies often use a prior that defines likely positions of the target, which are used to direct the sensing. Additionally, a planning approach that makes use of this prior together with the current world state to select the next action is part of most active perception systems. Realizing this interplay between sensing and acting is far from trivial, the following points need to be addressed.

1) How to build a prior for the task that reflects the state of the world?
2) How to model the search actions to come up with a plausible search plan?

These design questions are of great importance to the performance of the system. In this paper, we will show how a prior can be modeled, computed, and utilized by an autonomous robot searching for an object in an unknown world.

Research focusing on the computation of the aforementioned prior appeared in the literature as early as 1976. Garvey presented an implementation of a vision system capable of finding objects in scenes by making use of certain assumptions about the semantic scene structure [15]. One example of a search run is given where the target object is a telephone. The system realizes that due to the small size of the target object, searching the whole image would be wasteful. Instead, the system plans to *search for a table first*, and then searches in the image region that corresponds to the table top for the telephone. Garvey calls this type of search *indirect search*. Wixson and Ballard [16] provide quantitative results by comparing two search strategies, with and without indirect search for the same task.

In a series of papers, Ye and colleagues discuss computing the next best view to move the camera to in an object search task [6], [17]. A probability distribution over the 3-D space is assumed to be given and is tessellated into identically sized cells. Each cell contains the occupancy information as a binary state, and the probability of the center of the target object being in this cell. Knowing the field of view of the camera, the probability mass covered by each view can be computed by summing over the probabilities of cells that are located in the field of view. This probability sum is a measure of how good a certain view is for the task at hand. A number of views are sampled from the free space, and the system greedily picks the view that has the highest probability sum. In the most recent state-of-the-art visual search system from the same research group, Andreopoulos *et al.* [2] employ a similar strategy to object search using the humanoid robot ASIMO in a 4 m $\times$ 4 m $\times$ 1.5 m search space. In parallel, Ma *et al.* [18] use a probability map to guide the search and determine where to move, in a similar fashion to [17]. The authors present a scale-invariant feature transform (SIFT)-based method to find and estimate the six-degree-of-freedom pose of a target object.

Ekvall *et al.* [19] present an object detection method that can be used to compute likely positions for a given object in an image. The authors use a two-step approach where the first step is to find regions in the image that are likely to contain the object. Then, the system zooms into these areas and searches at a finer scale. The authors present a mobile robot system for searching for objects in multiple rooms of an office floor. However, the map and the location to search from are known *a priori* as in [7].

Similar to [19], the idea of first finding object hypotheses with a fast visual algorithm and, then, zooming into likely object locations to perform more expensive computation are revisited by Forssén *et al.* [20]. The object search task is divided into three separate sequential steps. First, the mobile robot system explores the environment to build an occupancy map. In the second step, the robot attempts to cover the environment as much as possible, this time with its peripheral cameras. During this step, object hypotheses are computed based on depth from stereo and spectral residual saliency described in [21].

The method, which is described in [7], utilizes object–object co-occurrence probabilities as a way to shape the prior on the object location over the search space. The map of the environment is fully known *a priori*. The system then plans a path in the map that once traversed by the robot has a high probability of spotting the object. There is no view planning involved, and the sequence of images, while the robot is traveling along this path, is analyzed to find the target object. The system is evaluated with three objects: chair, bicycle, and monitor. The biggest limitation of the system is the assumption of a known map and previously detected objects scattered throughout the whole environment.

Viswanathan *et al.* [8] have shown a similar system in which a method for place labeling is used to bootstrap the search. As in [7], this approach also uses the semantics of the environment to make the search more efficient. Simulation experiments of search indicate that making use of the environment semantics results in fewer analyzed views compared with an uninformed coverage-based search strategy.

The aforementioned methods provide different ways of constructing priors with various assumptions about the initial state of the robot and the environment. As stated earlier, another important point of an active perception system is the need to *plan* what sensing or moving actions are required to achieve the task. This is generally called *view planning* [22], which requires constant monitoring of the world and replanning if necessary. We will now focus on the literature that deals with this aspect of the visual object search problem.

In its simplest form, we can think of the view planning problem as covering a certain search space with sensors that have limited field of view. Often, minimizing the number of sensing actions and movement cost is desirable for increased search efficiency. Art gallery algorithms deal with this exact problem. Given a 2-D polygon representing an art gallery (the search space), and a limited amount of guards to protect the artworks (viewpoints from which part of the environment is visible), what is the best way to place guards to cover the polygon fully? This problem has been well researched in the computational geometry literature, and an extensive survey of the results can be found in [23] and more recently in [24] for mobile guards.

A number of researchers adopted the algorithms from the art gallery literature to mobile robotics. González-Banos *et al.* [25], [26] present a randomized art gallery algorithm for mobile robots that are tasked with covering an environment. Sarmiento *et al.* [27] present a heuristics-based method to find an object in a 2-D polygon world. In a follow-up work, the authors present a sampling-based algorithm, which is similar to [25], to find an object in a 3-D environment [28]. Such coverage-based solutions provide an accurate description of the problem when the sensing capabilities of the robot are deemed noise-free, and the world state is assumed to be completely known; in a typical robotics scenario, there are uncertainties associated with sensing and action.

Some recent papers tackled the problem of uncertainty by drawing inspiration from the planning literature. Hollinger *et al.* apply a POMDP planner to the problem of object search with single or multiple searchers [3]. In order to model the object search problem as a POMDP, the continuous 3-D search space needs to be discretized carefully due to the high computational complexity of most state-of-the-art POMDP solvers. The authors discretize an entire simulated office building floor into 69 rooms as possible object locations. They make the assumption that whenever the robot and the object are in the same room, the object is detected. This is a big simplification of detecting an object with a camera since the task of finding an object in a place as big as a room involves many difficulties such as calculating a good viewing position, dealing with occlusions, and detecting objects that appear small in the image. The authors provide simulation results and a proof-of-concept implementation where a mobile robot is tasked with finding cups in an already known environment with known search positions that the robot may choose to stop and take a picture from.

Sridharan *et al.* propose a POMDP framework for planning a sequence of visual operators in a scenario where the robot converses with a human about objects on a table top [29]. The task involves the system to find objects referred by a human in various ways through natural language. The authors demonstrate the usage of a decision theoretic framework for deciding which visual operators to apply on which regions of the image in order to answer queries about the table top scene.

Similar to [19], Masuzawa and Miura [30] present a approach where a mobile robot attempts to detect as many objects as possible in an environment of known size but unknown obstacles. The system uses SIFT features to detect object candidates, and then employs what the authors call a verification planning algorithm to confirm the presence of these candidates. Further, Boussard and Miura [31] present early results on modeling the search problem as a constrained Markov decision process (MDP). The planning problem is constrained in the sense that the authors allow a certain amount of time during which the robot has to detect as many objects as possible. The results, which are shown in simulation, indicate the plausibility of the approach.

Recent research works in [32] and [33] investigate the usage of radio-frequency identification (RFID) sensors for the object search problem. Although visual search poses challenges, such as illumination and viewpoint changes and object detection that RFID sensors do not suffer from, RFID antennas also have limited field of view. The system presented in [32] searches for certain product shelves in a supermarket setting. The environment is represented as a connectivity graph. The method exploits the default knowledge about supermarkets in that related products are stored in nearby shelves. The authors compare their results with human search performance measured in path length during search. Deyle *et al.* [33] coin the term RF vision for building and analyzing images of the environment where each pixel represents the signal strength of a certain RFID tag in the corresponding direction. This image is used to infer the 3-D location

TABLE I
TABLE COMPARING APPROACHES TO OBJECT SEARCH WITH MOBILE ROBOTS

			[2]	[18]	[19]	[20]	[7]	[8]	[34]	[28]	[3]	[30]	[32]	[33]	this work
Scenario		Large-scale space	-	-	✓	-	✓	-	✓	-	-	-	✓	-	✓
		Realistic real-world environment	-	✓	-	-	✓	-	✓	✓	✓	✓	✓	✓	✓
		Quant. eval. of search method	✓	✓	-	-	✓	✓	-	-	-	-	✓	✓	✓
Knowledge Rep.	W. State	Environment map	●	●	○	●	○	○	○	○	○	●	●	●	●
		Object information	○	○	○	○	●	○	●	○	○	○	●	○	●
		Place information	●	●	○	●	○	●	○	○	○	○	●	○	●
	Prior	Object-object relation	-	-	-	-	✓	-	-	-	-	-	✓	✓	✓
		Object-place relation	-	-	-	-	-	✓	✓	-	-	-	✓	-	✓
		Place-place relation	-	-	-	-	-	-	-	-	-	-	-	-	✓
Actions / Planning		Automatic viewpoint estimation	✓	✓	✓	✓	-	✓	-	✓	-	✓	-	-	✓
		Planning multiple steps ahead	-	-	-	-	✓	-	-	-	✓	✓	✓	-	✓
		Optimal plan (POMDP)	-	-	-	-	-	-	-	-	✓	-	-	-	-
		Autonomous exploration	✓	✓	-	✓	-	-	-	-	-	✓	✓	-	✓
		Exploration vs exploitation	-	-	-	-	-	-	-	-	-	✓	✓	-	✓
		Goal-directed exploration	-	-	-	-	-	-	-	-	-	✓	✓	-	✓

Legend: ○ - Given before the search started, ● - Acquired during the search process.

of the target object by the fusion of three sensory modalities: 1) an RF antenna; 2) a low-resolution camera; and 3) a tilting laser scanner. The authors describe a method for fetching an object equipped with an RFID tag from a signal strength image of the scene.

Kunze *et al.* present an object search system that utilizes default knowledge about typical object locations in indoor environments [34]. The authors utilize a semantic map framework in order to reason about where to search for a given object. The search locations in the map are assumed to be known in advance, and the robot picks the order to visit these locations to find the object.

Previously, we have implemented and evaluated a number of approaches for an AVS task. In [5], we have revisited the idea put forward in [15], and investigated the usage of spatial relations to efficiently locate objects when the environment is known. In this study, no planning was involved, and the search space consisted of a 6 m $\times$ 5 m furnished room. The conclusion from this study was that although using spatial relations greatly increases search efficiency, it is often hard to detect objects that are in relation to the target object, such as tables and bookshelves.

Later on, in [35], we have instead used an MDP planner to choose between which locations to search. Finally, in [36], we relaxed the assumption of a perfectly known world, and have taken object search to a much larger scale with qualitative results. The semantics of the environment represented in a probabilistic framework was key to reducing the search space in this study. This paper is based on our recent previous work [36], and provides more detailed explanations of various parts of the AVS strategy employed, and presents a thorough quantitative evaluation.

IV. CONTRIBUTIONS

There are several aspects of the work presented in this paper that push it beyond the current state of the art (see Table I). We work with large-scale environments of the size of an 18-room building floor. This is in sharp contrast with [2] where the environment is a single room. Previous work in large-scale environments has assumed that the environment is already known [7], or is completely explored in a first step [8], or have made use of nonvision sensors such as RFID [32]. We work with the case where the robot starts with no information about the specific environment. What sets our approach aside from, for example [8], is that we interleave exploration and exploitation (i.e., search in the known part of the environment) and do not start by exploring the entire environment, which could be quite time consuming. As a way to deal with object search in unknown environments, we introduce a model that describes a distribution over possible extensions to the currently known world, which allows the system to reason about whether to exploit the known part or explore the unknown part of space in a principled way. We present a thorough experimental evaluation with both qualitative and quantitative analysis. Finally, we provide a gold standard for comparison by letting human participants perform the same task using the robot embodiment (remote control and observing the environment through the cameras).

V. PROBLEM FORMULATION

The problem of active search, which is addressed in this paper, is that of finding an efficient strategy to localize a certain object in a large-scale unknown 3-D indoor environment we will refer to as Ψ following [37]. Concretely, we look for a strategy that decides what sequence of actions to execute to localize the point of interest, while minimizing the total cost, where cost is defined as time. The robot can execute motion actions and sensing actions in the space of Ψ. The sensing actions are characterized by the pose of the robot, camera parameters, and recognition algorithm.

Additionally, let $P_\Psi(X)$ be the probability distribution for the position of the target X in the search space Ψ. Depending on the level of *a priori* knowledge of Ψ and $P_\Psi(X)$, there are three extreme cases of the active search problem.

1) If both Ψ and $P_\Psi(X)$ are fully known, the problem is that of sensor placement and coverage maximization, given limited field of view and cost constraints.
2) If $P_\Psi(X)$ is unknown, but Ψ is known (e.g., the robot has acquired a map of the environment with no additional object-related information), the agent needs to gather information about the environment similar to the

aforementioned case. However, in this case, the exploration is for learning about the target specific characteristics of the environment. Knowing Ψ also means that the robot can reason on whether or not to execute a costly search action at the current position, or move to another more promising region of space in a straightforward manner.

3) If both Ψ and $P_\Psi(X)$ are unknown, the agent needs the ability to explore. The agent needs to select which parts of the environment to explore first depending on the target properties. Furthermore, the agent needs to tradeoff between executing a sensing action and exploration at any given point (i.e., should the robot search for the target in the partially known Ψ or explore further). This is classically known as the exploration versus exploitation problem.

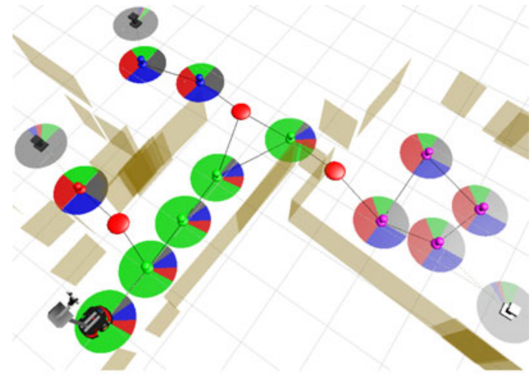
In this study, we consider the third case in which Ψ and $P_\Psi(X)$ are both unknown, and the robot is required to explore the environment. We provide the robot with probabilistic semantic common-sense knowledge, which is not environment specific and encodes relationships between high-level human concepts and functions of space. Typically, the common-sense knowledge encodes correspondences between objects, landmarks, other properties of space, and semantic room categories. Such information is valuable in limiting the search space and helps humans efficiently search in unknown environments. Our goal is to also leverage this to achieve similar efficient behavior in artificial systems.

VI. Search Space

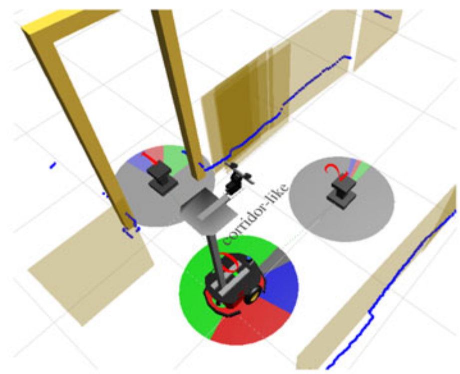
As pointed out previously, the ability to reduce the search space is crucial for practical applications. We choose to deal with this problem by directing the search toward locations that are likely to contain the object.

Indoor environments are typically organized into rooms, each fulfilling a specific function of everyday life. At the same time, the category of a room is often strongly correlated with the actions afforded by the objects found therein (e.g., a book is more likely to be found in offices rather than in kitchens). We argue that rooms are an important spatial concept for efficiently pruning large amounts of search space in typical indoor environments.[1] Our idea is to exploit the correlation between room category and objects as part of the semantics of the environment. Rooms have frequently been used in the past as nodes in topological representations [39]–[41]. Here, we make use of rooms as a means to implementing a divide-and-conquer strategy for the object search. Once a decision to search a room is made, the system can then analyze the room through a more detailed search, involving view planning by calculating where exactly to move the camera in this smaller part of the search space. Our assumption, which will be confirmed by the experimental evaluation, is that the cost of classification of rooms is more than compensated by the benefits.

Since we assume no initial knowledge of the specific environment in which the robot operates, the categories of rooms found in the environment have to be inferred based on observations acquired by the robot during the search. To this end, we have recently proposed a probabilistic semantic mapping framework [1]. The system bootstraps the probabilistic models by extracting default knowledge from databases. As we will explain in the next section in more detail, this allows us to reason about object presence in the known and unknown parts of space, by combining different types of observations (e.g., of objects and room appearance), and predicting existence of rooms of certain categories even in unexplored space.



(a)



(b)

Fig. 2. (a) Place map with several places and placeholders shown as large circular disks and three detected doors shown as smaller disks. The places have circular pins at the center of disks and placeholders have rectangular pins. Colors on disks indicate the probability of a place being in a room of a certain category in the form of a pie chart. Here, green is *corridor*, red is *kitchen*, and blue is *office*. (b) Start of a search run where two placeholders are detected with different probabilities of leading into new rooms of certain categories. The size of the color indicates the probability value that the placeholder leads to a new room of a certain category (gray represents the case that there is no new room). One of them is behind a door hypothesis, therefore having a higher probability of leading into a new room.

A. Modeling the Search Space on the Environment Scale

Our modeling of the search space is as follows. On the large scale (e.g., a whole building floor containing multiple rooms), we represent the search space as an undirected graph called the *place map*. The nodes of the graph correspond to discrete *places* in the environment, and are created at equal intervals as the robot moves. Edges in the graph represent direct paths between places. Together, places and paths represent the topology of the environment. An example of a place map is shown in Fig. 2(a).

[1]We note that rooms, as defined here, do not have to have physical boundaries, such as walls and doors, as demonstrated in [38].

The places in the place map are further grouped into rooms by detecting doors in the environment. In addition, unexplored parts of the environment are represented in the place map using hypothetical places called *placeholders*, which are defined in the boundary between free and unknown space in the metric map [42], [43]. Both places and placeholders are associated with beliefs about room categories estimated based on the available knowledge about the explored part of the environment. To assist in deciding which room to search or which placeholder to explore in an object search task, we estimate two probability distributions related to object presence in the already discovered rooms and in unexplored space.

1) $p(O^{o_i}_{r_j}|\boldsymbol{\theta})$, $O^{o_i}_{s,r_j} \in \{0,1\}$: distribution indicating whether an object of the category o_i exists in not yet searched area of one of the known rooms r_j, derived from all the observations $\boldsymbol{\theta}$ collected by the robot up to this point.
2) $p(O^{o_i}_{h_j}|\boldsymbol{\theta})$, $O^{o_i}_{s,h_j} \in \{0,1\}$: distribution indicating whether an object of the category o_i exists in a potential new room, which can be discovered after exploring in the direction of placeholder h_j, derived from all the observations $\boldsymbol{\theta}$ collected by the robot up to this point.

As noted previously, in order to calculate the above, we exploit the relationship between the room category and object presence of a certain category. Therefore, we calculate two types of room category probabilities, for explored and yet unexplored space.

1) $p(C_{r_j}|\boldsymbol{\theta})$, $C_{r_j} \in \{c_k\}_{k=1}^{N_C}$: distribution over room categories (for N_C categories in total) for a given known room r_j and all the observations $\boldsymbol{\theta}$ that the robot gathered up to this point.
2) $p(C^{c_i}_{h_j}|\boldsymbol{\theta})$, $C^{c_i}_{h_j} \in \{0,1\}$: distribution indicating whether the placeholder h_j leads to a new room of the category c_i upon exploration. The knowledge about unexplored space is derived from all the observations $\boldsymbol{\theta}$ gathered by the robot in the explored part of space.

This information can be used to decide whether to explore one placeholder instead of another, or simply perform fine-grained search for an object in one of the previously discovered rooms. A visualization of the distributions is presented in Fig. 2.

B. Assigning Probabilities

In order to calculate the aforementioned probability distributions for the partially explored environment, we used the probabilistic semantic mapping framework, which was recently proposed in [1]. The joint distribution that represents the dependences between object categories and room categories for known rooms is modeled as a probabilistic chain graph model [44]. Chain graphs are a natural generalization of directed (Bayesian Networks) and undirected (Markov Random Fields) graphical models. As such, they allow for modeling both directed causal as well as undirected symmetric or associative relationships, including circular dependences. The structure of the model is presented in Fig. 3, and is adapted at run time according to the state of the underlying topological map.

The semantic mapping framework relies on several properties or attributes of space obtained from various modalities. Those properties characterize each of the places and contribute to the knowledge about room categories. We use the following properties in our implementation: geometrical room shape and size obtained from laser range data, and general room appearance captured by a camera. In the chain graph model, which is shown in Fig. 3, the values of those properties are represented as a set of variables $(SH_{p_i}, SI_{p_i}, A_{p_i})$ for shape, size, and appearance properties, respectively. They are generated automatically for each discovered place p_i based on the topology of the place map.

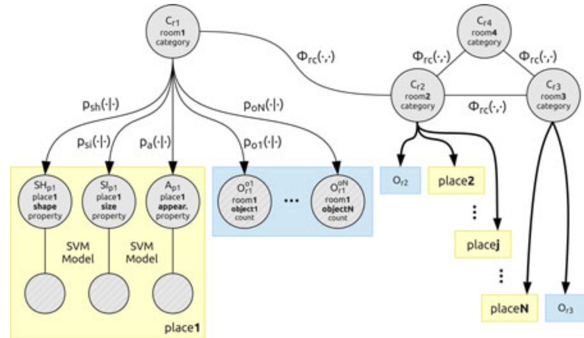


Fig. 3. Structure of the chain graph model that represents the search space at the large scale. The vertices represent random variables. The edges represent the directed and undirected probabilistic relationships between the random variables. The textured vertices indicate observations that correspond to sensed evidence. The yellow rectangles group variables that represent spatial properties of a single place. The blue rectangles group variables that represent object information for a single room.

The spatial property variables for all places in a single room r_j are connected to a random variable C_{r_j} that represents the functional category of the room. The relations between place properties and room categories $(p_{sh}(SH_{p_i}|C_{r_j}), p_{si}(SI_{p_i}|C_{r_j}), p_a(A_{p_i}|C_{r_j}))$ are derived from the default knowledge. The shape, size, and appearance properties can be observed by the robot in the form of features extracted directly from the robot's sensory input. As proposed in [1], the links between observations (textured vertices in Fig. 3) and the place property variables are quantified by categorical models of sensory information implemented using support vector machines (SVM). A separate SVM model is trained for each spatial property (shape, size, and appearance). The models are trained from sequences of images and laser range data recorded in multiple instances of rooms belonging to different categories and under various illumination settings. During recognition, confidence measures are derived from the distances between the classified samples and SVM hyperplanes [45]. The confidences are accumulated across all views acquired within a place [41] and then normalized to form probabilities.

Additionally, for each room, there is a set of variables that represents the presence of a certain set of objects of each category in the already searched space inside the room $(O^{o_1}_{r_j}, \ldots, O^{o_{N_o}}_{r_j}$, $O^{o_i}_{r_j} \in \mathbb{N}_0)$ (e.g., for reasoning about finding another cup in a kitchen, after having found one cup). Those variables are linked to the corresponding room category variable C_{r_j}. This relation represents the default knowledge about canonical object locations (e.g., that a coffee machine is likely to be found in a kitchen). Since we do not implement an observation model for the objects, the values of those variables are directly observed and set to a certain value depending on the number of objects of a certain category detected in the room.

Finally, the potential functions $\phi_{rc}(C_{r_i}, C_{r_j})$ describe knowledge about typical connectivity of two rooms of certain categories (e.g., that kitchens are more likely to be connected to corridors than to other kitchens). Those connections propagate semantic knowledge between rooms represented in the topological map.

The default knowledge about room connectivity, shapes, sizes, and appearances was acquired by analyzing the annotations of the COLD database, typically used for experiments with place categorization [1], [46]. The database consist of floor plans and images captured in various environments labeled with room categories. The database was additionally labeled with the values of spatial properties (shape, size, general appearance). Then, co-occurrences between room categories of neighboring rooms as well as room categories and property values were counted and later normalized to form distributions.

The conditional probability distributions $p_{o_i}(O^{o_i}_{r_j}|C_{r_j})$ that relate the number of objects ($O^{o_i}_{r_j} \in \mathbb{N}_0$) of a certain object category o_i present in an already searched part of a room r_j to the category of r_j (C_{r_j}) are represented using Poisson distributions (e.g., probability of finding another cup in a kitchen after having searched for one). The rationale behind this is that each occurrence of a certain object category in a room is conditionally independent from each other, with an expected total number of objects for that room category. The Poisson distribution allows us to easily model the expected number of object occurrences in a room through its parameter λ

$$p_{o_i}(k|c_j) = \frac{(\beta\lambda_{o_i,c_j})^k e^{-\beta\lambda_{o_i,c_j}}}{k!}. \quad (1)$$

The parameter k is the actual number of object occurrences that we are interested in (e.g., what is the probability of finding *two* books in this room?), and β indicates the percentage of the room already searched by the robot (e.g., half of the room). In our model, λ_{o_i,c_j} is estimated separately for each object type and functional room category. It is calculated from the probability of existence of an object of type o_i in a room of category c_j obtained from commonsense knowledge databases. The process is first bootstrapped using a part of the *Open Mind Indoor Common Sense* database[2] from which potential pairs of objects and their locations are extracted. Those pairs are then used to generate "*obj* in the *loc*" queries to an online image search engine. The number of returned hits is then used to obtain the probability value. More details about this approach can be found in [47]. Once the probability of existence of an object of a specific type in a room of a specific category is obtained, the λ_{o_i,c_j} is calculated so that $\sum_{k=1}^{\infty} p_{o_i}(k|c_j)$ is equal to that probability.

Given observations of some of the objects and properties of space for the explored part of the environment, the distribution $p(C_{r_j}|\boldsymbol{\theta})$ over room categories of a room r_j can simply be calculated by marginalizing over all other variables in the chain graph model. In the following, we describe the models that are used for reasoning about unsearched and unexplored parts of the environment.

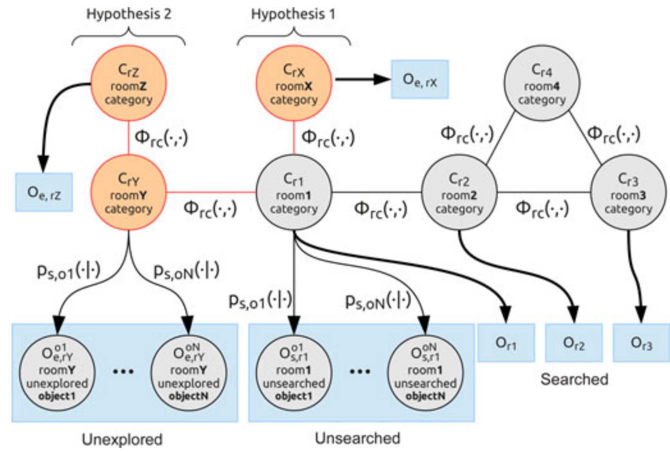


Fig. 4. Examples of extensions of the search space model permitting reasoning about unexplored space behind placeholder located in room 1.

C. Reasoning About Unsearched Parts of the Environment

Given the model built for the explored and searched part of the environment, we can now use it to reason about the presence of objects and rooms in yet unsearched or unexplored space behind a placeholder. To this end, the chain graph model is extended in two ways.

First, for the unsearched space, as shown in Fig. 4, we add a set of variables $O^{o_1}_{s,r_j}, \ldots, O^{o_{N_o}}_{s,r_j}, O^{o_i}_{s,r_j} \in \{0,1\}$ that allows us to reason about the presence of objects of various types in unsearched parts of known rooms. The distributions $p_{s,o_i}(O^{o_i}_{s,r_j}|C_{r_j})$ are represented in a very similar fashion to (1), however, this time focusing on the remaining unsearched portion of space $1-\beta$. Since, in order to direct the search, we are only interested in the presence of at least one instance of the object, $p_{s,o_i}(O^{o_i}_{s,r_j}|C_{r_j})$ simplifies to

$$p_{s,o_i}(O^{o_i}_{s,r_j}{=}1|C_{r_j}{=}c_l) = 1 - e^{-(1-\beta)\lambda_{o_i,c_l}}. \quad (2)$$

Then, the probability $p(O^{o_i}_{s,r_j}|\boldsymbol{\theta})$ is obtained by marginalizing over all the other variables in the chain graph model.

Second, in order to reason about unexplored space behind a placeholder, we hypothesize potential room configurations in the topological map of the environment. For each configuration, we extend the chain graph from the room in which the placeholder exists with variables representing categories of hypothesized rooms. Then, the categories of the hypothesized rooms are calculated by performing inference on the chain graph, and the probability of existence of a new room of a certain category behind the placeholder is obtained by summing over the room category inference results for all possible configurations.

In our system, we consider three hypotheses[3]: 1) placeholder leads to a single new room; 2) placeholder leads to a new room connected to another new room; 3) placeholder does not lead to a new room. For the cases 1) and 2), we extend the chain

[2]http://openmind.hri-us.com/

[3]These are based on the observation that in typical indoor environments, you can reach most rooms in two steps, thanks to "connector room" like corridors.

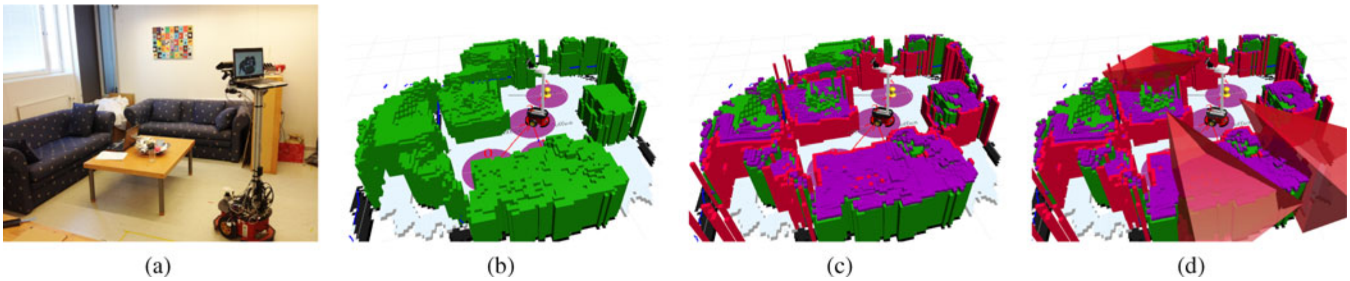


Fig. 5. (a) Robot during a search run in a room. (b) Corresponding 3-D map of the room built by the robot, where green areas represent obstacles. (c) Spatial probability distribution over 3-D space shown in purple, where occupied regions of space have a likelihood of containing the object. (d) Three out of several viewpoints computed for this room are shown.

graph model, as shown in Fig. 4, by adding variables C_{r_X}, C_{r_Y}, and C_{r_Z} connected to the variable representing category of the room in which the placeholder is located. The probability of there being a new room of a certain category c_i behind the placeholder h_j is then calculated as follows:

$$p(C_{h_j}^{c_i}=1|\boldsymbol{\theta}) = p(r_{h_j}^1)\, p(C_{r_X}=c_i|\boldsymbol{\theta}) + p(r_{h_j}^2) \sum_{y=i \vee z=i} p(C_{r_Y}=c_y, C_{r_Z}=c_z|\boldsymbol{\theta}) \quad (3)$$

where $p(r_{h_j}^1)$ and $p(r_{h_j}^2)$ are priors assigned to each of the hypotheses. If we assign equal prior to the cases 1) and 2), it is sufficient to calculate a probability of the placeholder leading to at least one room $p(r_{h_j})$. This can be estimated as follows: $p(r_{h_j}) = p(h_{h_j})(1 - p(d_{h_j})) + p(d_{h_j})$, where $p(h_{h_j})$ denotes the probability that the placeholder h_j leads to another placeholder and, thus, potentially to another room, and $p(d_{h_j})$ is the probability of there being a doorway behind the placeholder obtained from a door detector. The value $p(h_{h_j})$ can be estimated from the amount of open space in the direction of the placeholder estimated from the laser range data. The outcome can be seen in Fig. 2(b).

D. Modeling the Search Space on the Room Scale

We maintain a 3-D metric map for each room that supports viewpoint selection for object search as well as obstacle avoidance and path planning. This map is represented as a 3-D grid consisting of equally sized grid cells. Each cell holds the occupancy information, and the probability of the target object being in this cell, as in [17].

The sum of the probability value of all the cells given a room comes from the chain graph model, namely, the estimated value of $p(O_{r_j}^{o_i}|\boldsymbol{\theta})$ described earlier. The total probability is uniformly distributed over all occupied cells as possible locations for the target object's center point (see Fig. 5). In this way, we connect the object probabilities at the place map to the finer 3-D metric representation of the same space.

Furthermore, changes in this 3-D spatial probability distribution should also influence the probabilities in the place map. As an example, processing a viewpoint inside a room without finding an object reduces the probability values of the cells that are visible from this viewpoint. This change needs to be reflected in the place map as well since the decision making algorithms need to operate on the place map level and not at the fine-grained 3-D map level, for reasons discussed in Section II.

To this end, as explained previously, we introduce the term β to the chain graph model that represents the ratio of the space searched by the robot in a room. By updating this value accordingly during the search process, the system can reason about the tradeoff between continuing to search the current room or execute another action such as exploration or search in another location.

VII. Actions and Control

In this section, we will first describe the set of actions that makes use of the above described model of the search space, and then present the planning algorithm that decides which actions to execute. However, our approach to devising a search strategy is not dependent on the specific planning algorithm employed. At any point in time, the robot can search an already explored room or explore more of the environment. This choice should be driven by the belief that doing so is the most cost efficient way to find the object, by making use of the probabilities related to space and objects explained previously.

For illustration, assume that we are given certain knowledge that the target object is never found in a kitchen, and the robot happens to be in a kitchen. The only rational decision is to explore the area outside of the current room (if any) in the hope of finding another room. This should fall out of the algorithm and not be scripted. In addition, the decision will never be as clear cut as this; there will be a certain probability of finding the object in the known part of space and some in the unknown (as long as we have not found the object).

We design the set of actions required for such intelligent search behavior, regardless of the specific planner being used. The planner needs to reason with probabilities, and it needs to operate at a wide range of levels of abstraction, i.e., from high-levels decisions, such as moving between rooms, to the precise placement of the camera to acquire an image.

A. Modeling Actions

We define four actions: 1) Move; 2) ProcessView; 3) CalculateViews; and 4) SearchObject. The Move action moves the robot to a place or a placeholder. The ProcessView action moves the robot to a viewpoint and runs an object detection algorithm on the image acquired from this location. The object is deemed as found when the object detection algorithm reports a true positive.

The action CALCULATEVIEWS is used to calculate a set of viewpoints in a single room. Each viewpoint consists of a position on the 2-D plane at the camera's height, a pan, and a tilt angle. As introduced previously, let Ψ be the 3-D search space tessellated into a set of cells $C = \{c_0 \dots c_m\}$, and $P_\Psi(c_i)$ be the probability of the object's center being in the ith cell. The set of candidate viewpoints is determined by randomly sampling the reachable space in Ψ and successively picking views from the sample set

$$\underset{j=1..M}{\operatorname{argmax}} \sum_{i=1}^{n} p(c_i) S(c_i, j) \tag{4}$$

where M is the number of candidate sensing actions and S is defined as

$$S = \begin{cases} 1, & \text{if } c_i \text{ is covered by the } j\text{th viewpoint} \\ 0, & \text{otherwise.} \end{cases}$$

Each viewpoint covers a set of cells in Ψ. The total probability sum of a viewpoint's cells is the robot's likelihood of finding the object upon processing this viewpoint. We compute successive views in the same way until the probability sum of all covered cells is above a given threshold as in [25]; an example is shown in Fig. 5. The reason for introducing the threshold is that often it is not possible to cover 100% of all probability due to occlusions, the robot's limited field of view, and clutter in the environment, making full coverage impossible. In this way, enough views are calculated that covers regions of the search space likely to contain the object.

If the object is not found after a PROCESSVIEW action, we set the probability of the cells covered in this viewpoint to a low value based on the sensor model, and update the β parameter accordingly to indicate that a part of the currently searched room is covered.

The SEARCHOBJECT action is for forming a subproblem for the POMDP planner whose action set consists of MOVE and PROCESSVIEW to search for an object in a single room. Its prerequisite is that viewpoints are already calculated for this room by the CALCULATEVIEWS action.

B. Planning

As we discussed in Section II, the state space of the object search task for large environments is prohibitively large for most POMDP planners without greatly simplifying the search space. It would also require us to perform many unnecessary calculations beforehand (such as calculating viewpoints for all rooms, even those which seem unpromising). This could be alleviated by creating a hierarchical approach, i.e., using one POMDP to select the room, and then a second one to select which viewpoints to visit in which order. However, if the task requires exploration, this becomes difficult, as we then have to model all possible worlds in a POMDP framework. The *switching planner* [48] we use can be regarded as a hierarchical planning algorithm, although it differs from the outline above in two major ways: the high-level planner is a fast classical planner that operates according to the continual planning paradigm [49]. In addition, the decomposition of the low-level POMDPs is not static, but computed by the switching planner on the fly, depending on the current world state. The planner takes the problem description, in a dialect of PPDDL [50], called *decision-theoretic PDDL* (DTPDDL). See [48] for a detailed discussion on the planner and the input language.

```
(:assumption object-in-room
   :parameters (?r - room ?c - category
                ?o - visualobj ?l - label)
   :condition (and (= (category ?r) ?c)
                   (= (label ?o) ?l))
   :effect (probabilistic (P-exists ?l ?c)
               (assign (location ?o) ?r)))
```

Fig. 6. Assumption operator for placing objects in rooms. The probability ($\texttt{P}-\texttt{exists?l?c}$) is equal to $1 - p_{?l}(O^{?l}_{?r} = 0 | C_{?r} = ?c)$ (c.f. VI-B), i.e., the probability of finding at least one object of type ?l in a room of category ?c.

The switching planner alternates between two phases (called *sessions*). In a *sequential session*, the classical planner (we use a modified version of Fast Downward [51]) computes a single plan that reaches the goal with as little costs as possible. In classical planning, this usually means minimizing the number of actions (in the case of uniform action costs) or the sum of all action costs in a plan. Lowest costs is a suitable optimization criterion for fully observable deterministic tasks. However, this is not what we need in our situation. The cheapest plan would always be to search for an object at the robot's current location, no matter the likelihood of actually succeeding. Therefore, the switching planner uses a modified planning model, where in addition to costs, actions can have associated *success probabilities* $\mathrm{p}(a)$. We can then define the following objective function, which the continual planner tries to minimize

$$c = \sum_{a \in \pi} \operatorname{cost}(a) + R\left(1 - \prod_{a \in \pi} \mathrm{p}(a)\right)$$

for a plan π with R being the reward for reaching the goal. For small values of R, the planner will prefer cheaper but more unlikely plans; for larger values, more expensive plans will be considered if they have a higher probability of succeeding. Costs and success probabilities can be defined as per-operator constants in the DTPDDL domain, or they can be provided by the search space model (e.g., as in Fig. 6).

In our setting, the uncertainty is mostly part of the initial state. To capture this in the sequential plans, we force the planner to make *assumptions*. Assumptions are a type of *all outcome determinization* [52], a way for a classical planner to solve (some) problems that involve uncertainty. The planner may need to perform actions that depend on uncertain facts in the world (either as an action precondition or as part of a conditional effect). Assumptions are internal actions that the planner can use to establish that a fact is supposed to be true, allowing later actions in the plan to depend on this fact. They have the following properties: 1) an assumption may only be scheduled before any physical action so that all parts of the actual plan are based on the same set of assumptions and 2) if an assumption established

```
(:action SearchObject
   :parameters (?rb - robot ?o - visualobj
                ?l - label ?r - room)
   :precond (and (= (in-room (pos ?rb)) ?r)
                 (= (label ?o) ?l)
                 (viewpoints-created ?r ?l))
   :effect (when (= (location ?o) ?r)
                 (known-location ?o)))
```

```
(:action ProcessView
   :parameters (?rb - robot ?l - label
                ?v - viewpoint)
   :precond (and (= (pos ?rb) ?v)))

(:observe VisualObject
   :parameters (?rb - robot ?o - visualobj
                ?l - label ?v - viewpoint)
   :execution (ProcessView ?rb ?o ?v)
   :precond (and (= (label ?o) ?l))
   :effect (when (= (visible-from ?o) ?v)
                 (observed-location ?o)))
```

Fig. 7. High- and low-level search actions and observation model. The SEARCHOBJECT action is used in sequential sessions to select the room to search in. The PROCESSVIEW action is used, together with the observation model, by the POMDP planner when selecting the order in which to process the views.

a fact f, no other assumption may be used that establishes a different fact f' so that f and f' are mutex. For example, the planner may not make assumptions that a room is a kitchen and an office at the same time.

As assumption can (and usually does) have associated probabilities, making many (or unlikely) assumptions will lead to higher costs of the plan.

Whereas the original switching planner creates the assumptions from a grounded representation of the robot's belief state, we allow for an *implicit* representation of that belief state by allowing assumptions to be created from arbitrary operators. Such an operator that ties an object of a type `?l` to a room instance `?r` is shown in Fig. 6. Similar operators exist for partially searched rooms, or to describe the probabilities of finding a new room beyond an unexplored frontier. The associated probabilities for these assumptions (i.e., the probability of a room being a kitchen, or the probability of a placeholder yielding to a kitchen) come directly from the chain graph model, which was explained previously.

Once the planner tries to execute an action that depends on an uncertain fact (e.g., the effect of the SEARCHOBJECT action in Fig. 7 depends on the `location` of the object), a *contingent session* is started. The switching planner creates a POMDP for the subproblem of achieving the effect of the action that triggered the session. To keep the POMDP tractable, it only contains those facts that are relevant to the subgoal and which do not cause the initial belief state to exceed a given size (50 in our implementation). As the initial belief state is so small, states can be represented and computed explicitly. The resulting POMDP can contain all actions that are available to the continual planner (as described in Section VII-A) plus goal actions that provide a reward for correctly achieving the subgoal. However, most actions will usually not be executable in the subproblem, as their preconditions are not satisfiable in the restricted state of the subproblem. Observations are described in DTPDDL by an *observation model*. The (usually conditional) effect in an observation model describes the signal the POMDP planner will receive given that one or several facts hold in the true world state. In Fig. 7, the VISUALOBJECT model states that the planner will receive an `observed − location` signal for every object that is visible from a processed viewcone.

In contrast with the continual planning that takes place in a sequential session, the POMDP planner operates in a *closed-loop* manner. It does not compute a complete policy, but only outputs one action, waits for the resulting observations, and only then computes the next action. A contingent session can be either terminated by the POMDP planner itself by executing a goal action, or by the switching planner if an action results in a state that is outside the POMDP subproblem (e.g., if a move action places the robot in a previously unknown place), or causes the rest of the sequential plan to become invalid. In either case, the planner continues with the previous sequential session, replanning if necessary.

Consider the case where the robot is asked to find a coffee mug and has just started in a corridor with two placeholders, as illustrated in Fig. 2(b). As no viewpoints have been created at this time, the only way the planner can satisfy that goal (making `known − location` true for a `mug` object) is executing the SEARCHOBJECT action in a room, with the assumption that the object is in that room. Together with the action's precondition, the planner has to satisfy three subgoals: 1) the robot is in that room (first precondition); 2) viewpoints must be created (second precondition); and 3) the assumption that the object is in the room must be established (effect condition).

In this case, it might assume that the current room is a kitchen, since kitchens have the highest likelihood of containing a coffee mug among the categories and searching a kitchen is the most promising lead toward finding the object. It would then start exploring the room, as this is a direct precondition of the CALCULATEVIEWS action. During this exploration, it may quickly become clear that the room is not likely to be a kitchen after all, causing the planner to replan. It may then switch to a plan that assumes that it finds a new kitchen beyond one of the placeholders, and will proceed to explore in that direction.

The search in a single room is handled by the POMDP planner. Once the planner tries to execute the SEARCHOBJECT action, it will switch to a contingent session, as the action's effect depends on the (unknown) fact (`locationmug`). When the robot decides to search a room, a subproblem for the POMDP planner is created, containing only the places and viewcones in the current room, as well as the object that the robot searches. This has two benefits. First, by forming simple problems for the POMDP planner on the fly, we employ a divide-and-conquer approach to avoid formulating too complex planning problems. Second, we use optimality where it matters the most in the system, that is, choosing the order in which viewpoints are processed.

Fig. 8. Objects used during the object search experiments (stapler, cereal box and coffee mug) are shown. Appearance models for these objects are known to the robot before each search run.

The movement between viewpoints constitutes by far the most expensive part of the whole object search execution. We believe this way of mixing optimality and nonoptimal fast planning is a good fit for a robotics scenario. If all viewpoints in a room are processed without finding the object, the β parameter is updated (as explained in Section VI-D), and the continual planner is invoked to compute a new plan with the updated probabilities.

VIII. Experiments

Experiments took place in a 33 m $\times$ 12 m environment with 15 different rooms of which 12 are office rooms, one is a kitchen, and one is a meeting room connected by a corridor. The mobile robot platform utilized is a Pioneer III wheeled robot, which is equipped with a Hokuyo URG laser scanner, a Microsoft Kinect camera, and a higher resolution camera mounted at 1.4 m above the floor.

Three different objects were used during experiments: a cereal box, a stapler, and a coffee mug (see Fig. 8). The BLORT vision toolkit [53] was used to detect all objects. The robot had *a priori* knowledge of the object models beforehand. The default knowledge indicates that the cereal box is mostly expected to be in a kitchen, the stapler in office rooms, and the coffee mug can be in almost any room in the environment except the corridor.

The whole system is implemented as separate components using the CAST Robotics Middleware Framework [54]. We have used the CURE software library for building a map of the environment and for path planning [55]. The 3-D grid map of the rooms consists of 10 cm $\times$ 10 cm $\times$ 10 cm cubes. The Microsoft Kinect camera, together with the laser scanner, are used to perform 3-D mapping and obstacle avoidance. The robustness in navigation gained from this allowed us to run our experiments in a dynamic cluttered real-world office floor.

A. Experimental Setup

It is generally very hard to obtain ground truth data on the task of searching for objects in large environments for quantitative analysis of the system. There are several reasons for this. First, there are no established datasets, as the active nature of the problem makes this hard, and there are no well-established simulation environments in the literature in order to compare the few systems that are designed to search for objects in large-scale environments. Second, in the absence of a benchmark dataset for visual search tasks, different systems need to be evaluated in the same conditions (e.g., same environment, same objects, and object placements) for a meaningful comparison of results.

We have implemented a method for comparison, later referred to as *uninformed search*, which does not make use of the semantics of the environment. With uninformed search, we aim to recreate the greedy search strategy that is employed in the most recent state-of-the-art systems on AVS such as [2], [6], and [18]. In this case, the robot, at each newly discovered room, first explores the room fully, and calculates viewpoints that cover the entire room. The robot then proceeds to process each viewpoint one by one, in a greedy fashion. The search continues until the object is found or there are no rooms left that have not been searched.

Furthermore, we also compare our method to a human performing the AVS task. We believe this provides a gold standard on testing the efficiency. The idea of comparing an object search method against human participants has been explored recently in [32]; however, in our work, we let the human remote control the robot and perceive the environment using the same sensors as the robot.

Finally, we have implemented the method proposed in this paper, later referred to as *informed search*, that uses the semantics of the environment to guide the search task by using the actions and the spatial representation presented in this paper.

Our experiment setup is as follows. We invited 12 people to the Center for Autonomous Systems (CAS) Laboratory. The participants are picked such that they had not seen the test environment beforehand and were not familiar with this work and robotics in general.

First, the target objects were shown to each participant from all view angles. This corresponds to learning the object models in the system. Then, each participant is given a driving practice of steering the robot with a joystick at another location, until they are comfortable in maneuvering the robot. The participants passed certain tests that proved they were able to drive the robot comfortably. This included moving between rooms and to designated places in the environment. Once the participants were able to control the robot with ease using the joystick, they then sat in front of a computer, which displayed a live feed video from the robot's cameras. The robot is placed in the test environment, which is the entire sixth floor of CAS. The participants were then asked to find one of the objects with the starting point for the robot, and the object location is picked randomly for each run.[4]

After a participant has completed a run, we changed the location of the object and asked the participant to perform another search task. This corresponds to the case when the robot has a known map of the environment at the start of the search task. For each of the human runs, we have run informed and uninformed object search methods in exactly the same conditions regarding the robot's starting position, presence or absence of an *a priori* map, and location of the target object. Repeating this process

[4]The cereal box object was placed only in the kitchen, the stapler was placed in one of the office or meeting rooms, and coffee mug was placed in any of the rooms except the corridor, commensurate with the default knowledge.

for the three objects, in total, resulted in 108 real-world test runs of an object search task for all three methods.

B. Quantitative Experiments

We have recorded the search trajectory and time during runs for both the unknown map and known map cases. Fig. 9 shows the average trajectory lengths and search times for all three cases for both when the map is known *a priori* and when it is unknown. To compare with the robot performance, we have only considered instances where the object is successfully found by the robot. The uninformed search is by far the most inefficient way of locating objects by traveling on average 112 m. In contrast with the use of semantics of the environment in order to guide the object search, the robot nearly halves the total search trajectory length. As expected, human runs have the shortest trajectory length. It is worth noting that the difference between the human performance and the informed search is smaller when the map is known. This shows that the method, which is presented in this paper, can utilize the already known information about space without additional algorithmic or implementation changes.

The difference in trajectory length between the human and the informed search methods is approximately 25 m in the unknown map case, and 18 m in the known map case. The main reason for this is humans are far better in visual tasks than robots, and can recognize objects and categorize rooms from far away in contrast with the robot. Therefore, we would expect that the search strategy and modeling of the search space presented in this paper would automatically benefit from better visual processes in categorizing rooms and detecting objects in images.

Looking at the object search task performance from the search time point of view, we see a similar ordering in terms of the three methods tested. The differences between the methods are larger than in the trajectory length case. The reason for this is that moving more in the environment requires turns and twists, which takes a longer time that does not manifest itself in the trajectory length metric. As can be seen from the graph, in the uninformed search case, it takes approximately 36.4 min to find the target object for the unknown map case, and 28.5 min for the known map case. This is clearly an unacceptable time to wait for an intelligent autonomous robot living in human spaces. On the other hand, by utilizing the semantics of the environment, as described in this paper, on average, we are able to find the object 15.8 min in the unknown map case, and 7.8 min in the known map case. This is a huge gain in efficiency. As expected, in the known map case, human participants were very adept at learning the environment. The search time for the human participants for the unknown and known map cases were 8.55 and 3.6 min, respectively. We note that some of the difference in time between the human and informed search is caused by humans navigating the robot more expertly around obstacles than the autonomous navigation algorithm. Fig. 10 shows the trajectories of human, uninformed, and informed search methods from a single run. The target object is the cereal box and the map is unknown. The starting position for all of the runs is the same, which is the leftmost position in the corridor indicated in Fig. 10.

The search trajectory of the human participant is shown in Fig. 10(a). At the start of the search, the human participant steered the robot into the office room. This was a commonly occurring behavior with most human participants. We argue that the reason for this is that the participant first required some initial information on the type of the environment. Upon realizing that the room is a typical office room, and most likely all other office rooms in this floor look alike, the participants typically have constructed their idea of what kind of an environment this is, therefore what types of rooms they should anticipate. Therefore, for the remaining part of the trajectory, they only peeked through the doors of other rooms to inquire its category, until they have found the kitchen room. The cereal box object was placed on a table in plain sight in the kitchen.

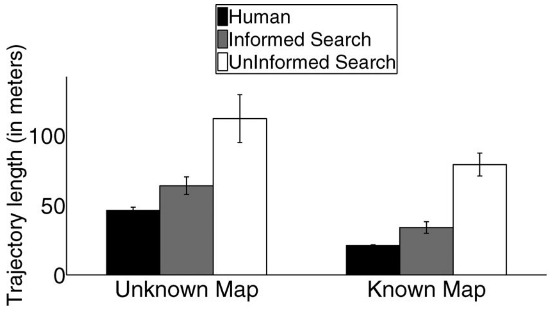
Fig. 10(b) shows the search trajectory of the method presented in this paper. The robot here starts out in the corridor and by exploration it enters each room. One difference between the human case is that, while humans can deduce a room's category by peeking through the doors, the robot has to enter the room and accumulate observations. However, after gathering evidence that the category of the room is not what the planner expected, the robot continues with the exploration in the corridor with the hopes of finding a kitchen room. As expected, the informed search method traverses a significantly shorter trajectory compared with the uninformed search method.

Finally, Fig. 10(c) shows the uninformed search trajectory. As expected, this strategy covers the whole environment without making use of spatial characteristics of the visited places. This results in a significantly less efficient search compared with other two methods.

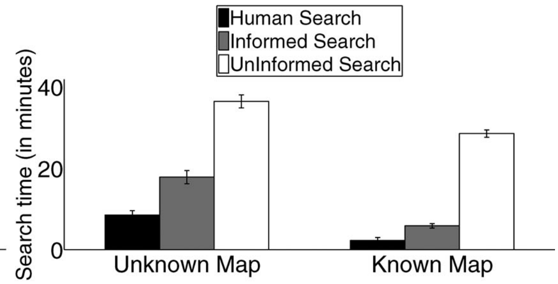
C. Qualitative Experiments

In order to show the adaptability of our system to different search conditions and provide better understanding of typical search missions, we have also run our system in a smaller three-room environment and logged the input/outputs and the resulting trajectories. We have evaluated six different scenarios with different starting positions, object locations, and map status (from completely unknown to partially known at the start of search) each time. Fig. 11 shows the search trajectory of all the runs. The trajectories are color coded. The colors indicate robot's room category estimates for the current position. Red, green, and blue corresponds to kitchen, corridor, and office, respectively. In the following, we give a brief explanation for what happened in the different runs.

1) Fig. 11(a) Starts: *corridor*, Target: *cereal box* in *kitchen*
   The robot starts by exploring the first placeholder in the *corridor*. After this, two more placeholders appear: one continuing in the corridor, the other on the left behind a doorway. The latter placeholder has a higher probability of leading into a kitchen due to it being nearer to the doorway, and the robot enters *office1*. The robot then starts exploring other placeholders appearing in *office1*. After exploring the second placeholder in *office1*, the room category of *office1* is deemed an *office*. Since default knowledge indicates cereal boxes are seldom in office rooms, the

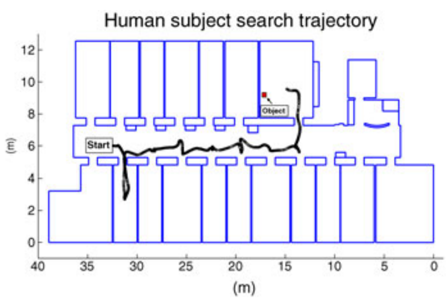


(a)

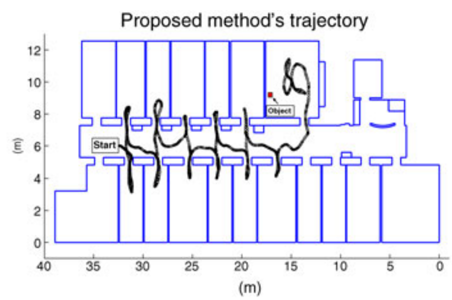


(b)

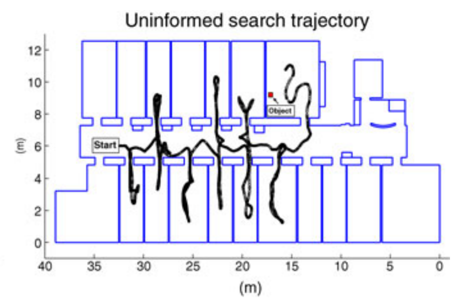
Fig. 9. (a) Average trajectory length and (b) average search time for human, uninformed, and informed search methods for both the unknown and known map cases over 108 search runs.



(a)

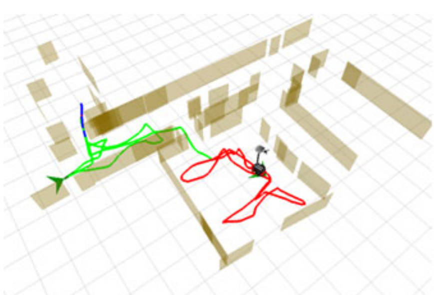


(b)

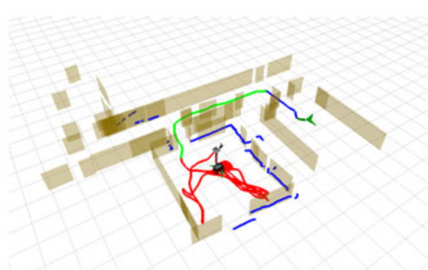


(c)

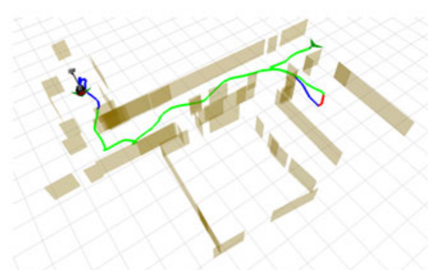
Fig. 10. Trajectories overlaid on the floorplan of the test environment from a visual search run for the object cereal box with unknown map. (a) Trajectory for one of the human runs. (b) Trajectory taken by the method presented in this paper. (c) Trajectory of the uninformed search method. The robot missed visiting two rooms in this example uninformed search run due to the inaccuracies in the robot's occupancy map, which resulted in missing placeholders for these rooms.



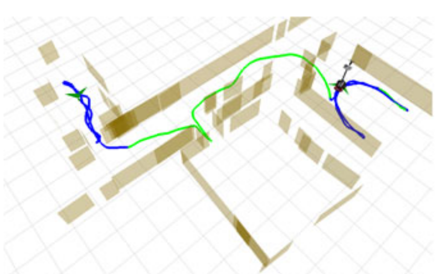
(a)



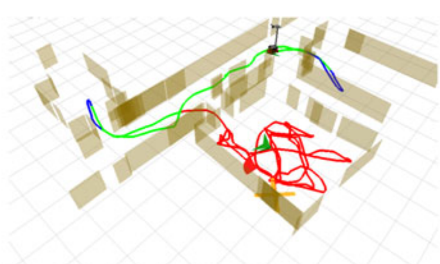
(b)



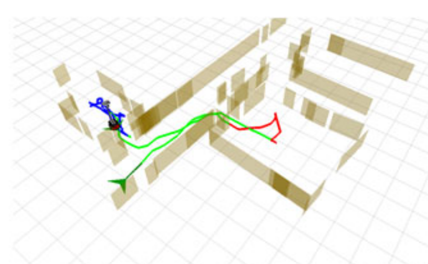
(c)



(d)



(e)



(f)

Fig. 11. Search trajectories that resulted from different object search runs, while utilizing the methods presented in this paper. The environment consists of three rooms. The colors indicate robot's estimate on the most likely room category with red, green, and blue corresponding to kitchen, corridor, and office, respectively. Each trajectory corresponds to an object search instance with varying target object placement and room accessibility, hence resulting in different search behaviors. This shows the flexibility of our approach in coping with different search conditions. A video of the above search runs can be viewed at http://csc.kth.se/ aydemir/Active-Visual-Search.html.

robot returns to exploring the corridor until it finds the *kitchen* door, and explores the placeholder near this door. This time, exploration of kitchen goes without interruption since cereal boxes have a high probability of being located in rooms with category kitchen. Finally, the robot computes views in this room with the CALCULATEVIEWS action. After processing some view positions, the cereal box object is found.

2) Fig. 11(b) Starts: *office2*, Target: *cereal box* in *kitchen*
Similar to Fig. 11(a), after exploring a few placeholders, the robot does not issue the search command in the current room, and continues with exploration until it finds the corridor. Eventually, the robot finds the room kitchen, and the rest proceeds as in Fig. 11(a).

3) Fig. 11(c) Starts: *corridor*, Target: *cereal box* in *kitchen*
The robot explores until it finds *office2*. Upon entry, the robot categorizes *office2* as kitchen but after further exploration, *office2* is categorized correctly. As a result of this, the robot switches back to exploration and since the kitchen door is closed, it passes the kitchen and finds *office1*. Similarly, after determining the category of *office1*, the robot sets out to explore more; however, there are no more placeholders to explore, and therefore, the search is stopped.

4) Fig. 11(d) Starts: *office1*, Target:*stapler* in *office2*
The robot starts by exploring the current room and meanwhile categorizes the room correctly as an office room. Since *stapler* has a high probability of being in offices, the robot launches a search in this room. However, the object is placed in *office2*, and the robot fails to find the object. After failing to find the object in *office1*, the robot continues with exploration, which leads it to the corridor. The robot then finds the room *kitchen* but after realizing that it is kitchen-like, decides not to search the kitchen room and continue its exploration. The robot then finds the room *office2*. After determining the category of this room, the robot launches a search, and this time, the *stapler* object is found in *office2*.

5) Fig. 11(e) Starts: *kitchen*, Target: *cereal box* in *kitchen*
As before, realizing that the current room is promising for cereal box, the robot calculates viewpoints in this room. After processing the views its visual algorithms fail to detect the object. After processing all views, it finally goes out into the corridor to look for another kitchen. However, the environment is fully explored and the search stops. This is a case where the search strategy has successfully brought the object into the field of the view of the robot however there was a failure in object detection.

6) Fig. 11(f) Starts: *corridor*, Target: *stapler* in *office1*
The robot is started in the corridor and driven to the kitchen by a joystick; thus, in this case, the environment is largely explored already when the planner is activated. The part of the corridor leading to *office2* has been blocked deliberately. By exploration, the robot finds its way to *office1*, and launches a search that results in a successful detection of the target object.

D. Comparison With Previous Work

In this section, we will compare our approach with those in previous works that are closest to our work. In short, we will focus on three different lines of work [7], [17], [18], [20]. We note that a quantitative comparison is not possible since it is either not possible to recreate the exact search environments in which these works have produced their results, obtain the same target objects or search conditions. Therefore, we will aim to give an extensive discussion on how the method, which is described in this paper, presents a contribution in the light of these research works.

The pioneering work by Tsotsos *et al.* in AVS with mobile robots introduced the first ideas on view planning in 3-D space with a moving agent and a spatial probability distribution defined over the search space [17]. In later work from the same authors, the environment is assumed to be unknown in advance as it is in this paper [6]. The robot exhaustively covers the search space in this work until the object is found, or the whole environment is covered. A very recent visual search system presented in [18] uses a similar greedy search strategy. The uninformed search method implemented and evaluated in this paper approximates to this type of search. As shown, such a method is highly inefficient for the search space and target objects depicted in this paper. In contrast, we utilize semantics of the environment to prune the search space, and guide the search toward more promising areas of the search space.

The system described in [20] takes on a different approach by first identifying candidate locations that are visually similar to the target object. After this first step, the robot then visits each of these locations to run a more computationally expensive and powerful object recognition algorithm. While this approach exploits the visual similarity, it still first needs to cover the whole space in order to generate candidate locations. In our case, this would mean exploring the entire floor, which would be prohibitively costly from a task completion point of view.

Finally, the work by Kollar and Roy [7] is closest to our work in the sense that environment semantics are utilized to guide the search. In this case, object–object co-occurrence properties are exploited in order to compute paths in the environment that lead to the target object. In order to accomplish this, the environment is first explored, and various objects are discovered. These objects later indicate where the target object might be in the environment. As an example, if the robot is looking for a chair, then an area where there are lots of tables can be a good candidate place. A path is computed to this area. There is no view planning involved, the target object is deemed found if the images while traversing the path contain the said object. In comparison, our method does not rely on first exploring and discovering a dense set of objects in the environment. Instead, we utilize default knowledge about object locations and room categories (e.g., cereal box is likely to be found in the kitchen) to guide the search.

IX. CONCLUSION AND FUTURE WORK

In this paper, we have argued that a search strategy that effectively reduces the search space, allowing for successful object search otherwise intractable, can be devised by exploiting the semantics of the environment. To this end, we have demonstrated how to build and use uncertain environment semantics in order to efficiently search for objects. Further, we have proposed a way of dealing with unexplored environments by reasoning about possible worlds in the same spatial model. We define search actions that allow efficient search over the whole environment by taking full advantage of this model of the search space. We show that these set of actions allow a flexible search system that can handle different starting positions and environments with ease.

Future work will focus on calculating a more informed 3-D probability distribution over the metric space by incorporating our work [56]. We think that this has the potential to further increase search efficiency at the viewpoint level. Another interesting future research direction is more sophisticated reasoning about the unexplored part of indoor environment, as in our previous work [57]. For this, we plan to use the learned indoor models from a large annotated floor plan database to help guide the robot in goal-directed exploration.

Furthermore, we would like to explore the case when a robot *lives* in the environment, visiting the same rooms and objects over large periods of time, and searching for objects occasionally during this time. We think that the search strategies, environment models, and models on object locations that can handle not only spatial but also spatiotemporal aspects of object search is an interesting direction for future research.

References

[1] A. Pronobis and P. Jensfelt, "Large-scale semantic mapping and reasoning with heterogeneous modalities," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 3515–3522.

[2] A. Andreopoulos, S. Hasler, H. Wersing, H. Janssen, J. K. Tsotsos, and E. Korner, "Active 3D object localization using a humanoid robot," *IEEE Trans. Robot.*, vol. 27, no. 1, pp. 47–64, Feb. 2011.

[3] G. Hollinger, D. Ferguson, and S. Srinivasa, S. Singh, "Combining search and action for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 800–805.

[4] K. Sjöö, D. Gálvez López, C. Paul, P. Jensfelt, and D. Kragic, "Object search and localization for an indoor mobile robot," *J. Comput. Inf. Technol.*, vol. 17, no. 1, pp. 67–80, 2009.

[5] K. Sjöö, A. Aydemir, and P. Jensfelt, "Topological spatial relations for active visual search," *Robot. Auton. Syst.*, vol. 60, no. 9, pp. 1093–1107, 2012.

[6] K. Shubina and J. Tsotsos, "Visual search for an object in a 3D environment using a mobile robot," *Comput. Vis. Imag. Understand.*, vol. 114, no. 5, pp. 535–547, 2010.

[7] T. Kollar and N. Roy, "Utilizing object-object and object-scene context when planning to find things," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 4116–4121.

[8] P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. K. Mackworth, "Automated spatial-semantic modeling with applications to place labeling and informed search," in *Proc. Can. Conf. Comput. Robot Vis.*, May 2009, pp. 284–291.

[9] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *Int. J. Robot. Res.*, vol. 30, no. 11, pp. 1343–1377, 2011.

[10] T. H. Chung, G. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics—A survey," *Auton. Robot.*, vol. 31, no. 4, pp. 299–316, 2011.

[11] S. D. Roy, S. Chaudhury, and S. Banerjee, "Active recognition through next view planning: A survey," *Pattern Recognit.*, vol. 37, no. 3, pp. 429–446, 2004.

[12] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, no. 8, pp. 966–1005, Aug. 1988.

[13] J. Tsotsos, "On the relative complexity of active vs. passive visual search," *Int. J. Comput. Vis.*, vol. 7, no. 2, pp. 127–141, 1992.

[14] J. Tsotsos and Y. Ye, "A complexity-level analysis of the sensor planning task for object search," *Comput. Int.*, vol. 17, no. 4, pp. 605–620, Nov. 2001.

[15] T. Garvey, "Perceptual strategies for purposive vision," Artif. Intell. Center, SRI Int., Menlo Park, CA, USA, Tech. Rep. 117, Sep 1976.

[16] L. Wixson and D. Ballard, "Using intermediate objects to improve the efficiency of visual search," *Int. J. Comput. Vis.*, vol. 12, no. 2–3, pp. 209–230, 1994.

[17] Y. Ye and J. Tsotsos, "Sensor planning for 3D object search," *Comput. Vis. Imag. Understand.*, vol. 73, no. 2, pp. 145–168, 1999.

[18] J. Ma, T. H. Chung, and J. Burdick, "A probabilistic framework for object search with 6-DOF pose estimation," *Int. J. Robot. Res.*, vol. 30, no. 10, pp. 1209–1228, 2011.

[19] S. Ekvall, D. Kragic, and P. Jensfelt, "Object detection and mapping for service robot tasks," *Robotica*, vol. 25, no. 2, pp. 175–187, 2007.

[20] P.-E. Forssén, D. Meger, K. Lai, S. Helmer, J. Little, and D. Lowe, "Informed visual search: Combining attention and object recognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 935–942.

[21] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. IEEE Conf. Comp. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[22] S. D. Roy, S. Chaudhury, and S. Banerjee, "Active recognition through next view planning: A survey," *Pattern Recognit.*, vol. 37, no. 3, pp. 429–446, 2004.

[23] T. C. Shermer, "Recent results in art galleries [geometry]," *Proc. IEEE*, vol. 80, no. 9, pp. 1384–1399, Sep. 1992.

[24] B. J. Nilsson, *Guarding Art Galleries—Methods for Mobile Guards*. Ph.D. dissertation, Lund Univ., Lund, Sweden, 1995.

[25] H. González-Banos, "A randomized art-gallery algorithm for sensor placement," in *Proc. 17th Annu. Symp. Comput. Geom.*, 2001, pp. 232–240.

[26] S. M. LaValle, D. Lin, L. J. Guibas, J.-C. Latombe, and R. Motwani, "Finding an unpredictable target in a workspace with obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 1997, vol. 1, pp. 737–742.

[27] A. Sarmiento, R. Murrieta, and S. A. Hutchinson, "An efficient strategy for rapidly finding an object in a polygonal world," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2003, vol. 2, pp. 1153–1158.

[28] A. Sarmiento, R. Murrieta-Cidz, and S. Hutchinson, "A Sample-based convex cover for rapidly finding an object in a 3-D environment. in *Proc. IEEE Int. Conf. Robot. Autom.*, Apr. 2005, pp. 3486–3491.

[29] M. Sridharan, J. L. Wyatt and R. Dearden, "Planning to see: A hierarchical approach to planning visual actions on a robot using POMDPs," *Artif. Intell.*, vol. 174, no. 11, pp. 704–725, 2010.

[30] H. Masuzawa and J. Miura, "Observation planning for efficient environment information summarization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2009, pp. 5794–5800.

[31] M. Boussard and J. Miura, "Object search: A constrained MDP approach," presented at the Workshop Active Percept. Object Search Real World, San Francisco, CA, USA, 2011.

[32] D. Joho, M. Senk, and W. Burgard, "Learning search heuristics for finding objects in structured environments," *Robot. Auton. Syst.*, vol. 59, no. 5, pp. 319–328, May 2011.

[33] T. Deyle, H. Nguyen, M. Reynolds, and C. Kemp, "RF vision: RFID receive signal strength indicator (RSSI) images for sensor fusion and mobile manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2009, pp. 5553–5560.

[34] L. Kunze, M. Beetz, M. Saito, H. Azuma, K. Okada, and M. Inaba, "Searching objects in large-scale indoor environments: A decision-theoretic approach," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 4385–4390.

[35] A. Aydemir, K. Sjöö, J. Folkesson, and P. Jensfelt, "Search in the real world: Active visual object search based on spatial relations," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2818–2824.

[36] A. Aydemir, M. Göbelbecker, A. Pronobis, K. Sjöö, and P. Jensfelt, "Plan-based object search and exploration using semantic spatial knowledge in the real world," in *Proc. Eur. Conf. Mobile Robot.*, Sep. 2011, pp. 13–18.

[37] Y. Ye and J. K. Tsotsos, *Sensor Planning for Object Search*. Ph.D. dissertation, Dept. Comput. Sci, Univ. Toronto, Toronto, ON, Canada, 1997.

[38] K. Sjöö, "Semantic map segmentation using function-based energy maximization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 4066–4073.

[39] S. Vasudevan and R. Siegwart, "Bayesian space conceptualization and place classification for semantic maps in mobile robotics," *Robot. Auton. Syst.*, vol. 56, no. 6, pp. 522–537, Jun. 2008.

[40] H. Zender, O. M. Mozos, P. Jensfelt, G. M. Kruijff, and W. Burgard, "Conceptual spatial representations for indoor mobile robots," *Robot. Auton. Syst.*, vol. 56, no. 6, pp. 493–502, Jun. 2008.

[41] A. Pronobis, O. M. Mozos, B. Caputo, and P. Jensfelt, "Multi-modal semantic place classification," *Int. J. Robot. Res.*, vol. 29, no. 2–3, pp. 298–320, Feb. 2010.

[42] J. L Wyatt, A. Aydemir, M. Brenner, M. Hanheide, N. Hawes, P. Jensfelt, K. Matej, G. M Kruijff, Pierre Lison, A. Pronobis, K. Sjöö, D. Skočaj A. Vrečko, H. Zender, and M. Zillich, "Self-understanding and self-extension: A systems and representational approach," *IEEE Trans. Autom. Mental Develop.*, vol. 2, no. 4, pp. 282–303, Dec. 2010.

[43] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. Comput. Intell. Robot. Autom.*, pp. 146–151, Jul. 1997.

[44] S. L. Lauritzen and T. S. Richardson, "Chain graph models and their causal interpretations," *J. Roy. Statist. Soc. Series B*, vol. 64, no. 3, pp. 321–348, 2002.

[45] A. Pronobis and B. Caputo, "Confidence-based cue integration for visual place recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2007, pp. 2394–2401.

[46] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, P. Jensfelt, and H. I. Christensen, "Towards robust place recognition for robot localization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 530–537.

[47] M. Hanheide, C. Gretton, R. W. Dearden, N. A. Hawes, J. L. Wyatt, A. Pronobis, A. Aydemir, M. Göbelbecker, and H. Zender, "Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Jul. 2011, pp. 2442–2449.

[48] M. Göbelbecker, C. Gretton, and R. Dearden, "A switching planner for combined task and observation planning," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Aug. 2011.

[49] M. Brenner and Bernhard Nebel, "Continual planning and acting in dynamic multiagent environments," *J. Auton. Agents Multiagent Syst.*, vol. 19, no. 3, pp. 297–331, 2009.

[50] H. Younes and M. Littmann, "PPDDL1.0: An extension to PDDL for expressing planning domains with probabilistic effects," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-CS-04-167, 2004.

[51] M. Helmert, "The fast downward planning system," *J. Artif. Intell. Res.*, vol. 26, pp. 191–246, 2006.

[52] S. W. Yoon, A. Fern, and R. Givan, "FF-replan: A baseline for probabilistic planning," presented at the 17th Int. Conf. Autom. Planning Scheduling, Providence, RI, USA, 2007.

[53] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze, "BLORT—the blocks world robotic vision toolbox," in *Proc. Workshop Best Practice 3D Percept. Model. Mobile Manipulat.*, 2010.

[54] N. Hawes and J. Wyatt, "Engineering intelligent information-processing systems with CAST," *Adv. Eng. Informat.*, vol. 24, no. 1, pp. 27–39, 2010.

[55] J. Folkesson, P. Jensfelt, and H. Christensen, "The m-space feature representation for SLAM," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 1024–1035, Oct. 2007.

[56] A. Aydemir and P. Jensfelt, "Exploiting and modeling local 3D structure for predicting object locations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2012, pp. 3885–3892.

[57] A. Aydemir, P. Jensfelt, and J. Folkesson, "What can we learn from 38,000 rooms? reasoning about unexplored space in indoor environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Oct. 2012, pp. 4675–4682.

Alper Aydemir received the Ph.D. degree in computer vision and robotics from the KTH Royal Institute of Technology, Stockholm, Sweden, in November 2012.

He is currently a Researcher with the Computer Vision Group, Jet Propulsion Laboratory, National Aeronautics and Space Administration, Los Angeles, CA, USA. His research focuses on spatial understanding for mobile robots, in particular, 3-D mapping, object search, and spatial modeling for robotic applications.

Andrzej Pronobis received the M.Sc. degree in computer science from the Silesian University of Technology, Gliwice, Poland, in 2005 and the Ph.D. degree in computer vision and robotics from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2011.

He is currently a Research Associate with the Department of Computer Science and Engineering, University of Washington, Seattle, WA, USA. His research focuses on spatial understanding for mobile robots and exploiting spatial semantics for language grounding, human–robot interaction, and generating efficient robot behavior.

Moritz Göbelbecker received the Diploma degree in computer science (Diplom-Informatiker) from the University of Freiburg, Freiburg, Germany, in June 2009.

Since August 2009, he has been with the Research Group on the Foundations of Artificial Intelligence, University of Freiburg, headed by Prof. Dr. B. Nebel.

Patric Jensfelt received the M.Sc. degree in engineering physics and the Ph.D. degree in automatic control from the KTH Royal Institute of Technology, Stockholm, Sweden, in 1996 and 2001, respectively.

Since 2012, he has been a Professor of computer science with the Centre for Autonomous Systems, KTH. He is the Co-Founder of the technology provider company Intelligent Machines Stockholm AB and has recently co-founded Volumental AB. His research interests include mapping, localization and navigation of mobile robots, spatial cognition, and system integration.