Object search guided by semantic spatial knowledge

A. Aydemir[1], A. Pronobis[1], K. Sjöö[1], M. Göbelbecker[2], P. Jensfelt[1]

***Abstract*—In this work we consider a mobile robot tasked with finding an object in an unknown office floor. Object search in realistic large environments is a crucial step for various mobile robot missions. We describe our spatial representation which grounds human-level spatial concepts in lower level representations for efficient informed object search and exploration. We present a principled planning approach to the visual search problem. Finally we perform real world experiments, in a larger and more complex environment than found in previous work.**

I. Introduction

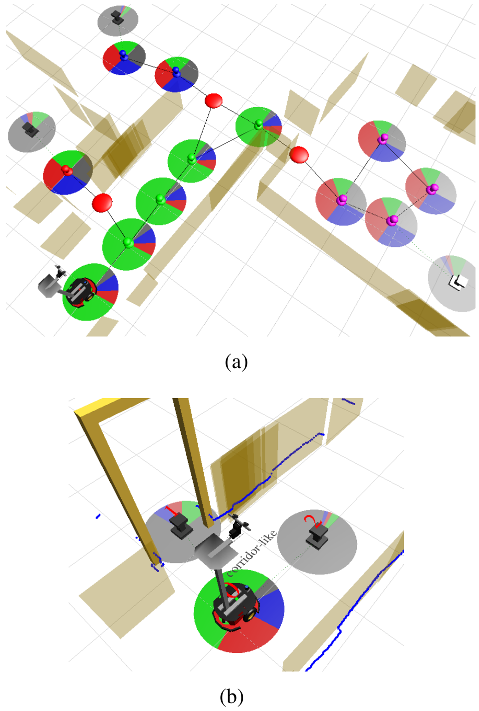
Finding objects in large environments with complex scenes is a necessary step for various mobile robot tasks. It is evident that a strategy that involves examining every possible scene is not feasible. A key insight is that the robot needs a relevant spatial representation that allows for efficient search as well as suitable vision algorithms. Furthermore the representation should take into account high level semantic components of space and ground them in the lower level spatial concepts. Finally to make use of such a representation, a principled way of selecting which action to take in order to bring the target object in the limited field of view of the robot is needed. In this work, we bring these components together to build a searcher robot. Tsotsos [9] showed that the problem of optimal visual search is NP-hard. Kollar and Roy [3] uses object co-occurence histograms to locate objects in the environment represented as a SLAM map. Let Ψ be a 3D search region and s be a sensing action for localizing an object o. The parameterization of s consists of camera position (x_c, y_c, z_c), pan-tilt angles (p, t), focal length f and a recognition algorithm a; $s = s(x_c, y_c, z_c, p, t, f, a)$. Let S be the set of all possible sensing actions with $S = s_0 ... s_t$ within Ψ. Also let P_o be the probability density function over S whose structure is unknown. We define the object search problem as calculating the subset of S which is most likely to bring the target object in the field of view of the robot.

II. Representing Space

The proposed spatial representation consists of three subrepresentations focusing on different aspects of space, from low-level sensory input to high-level conceptual symbols. Firstly, we maintain a 3D metric map which supports viewpoint selection for object search as well as obstacle avoidance and path planning. Then, a topological map called *place map* is generated which maintains the topology of the environment. Finally, all these sources of information are integrated in a conceptual map which ties symbols representing instance knowledge about the environment (e.g. room1, object1) with spatial concepts such as objects (a book), room categories (a kitchen) or appearances (a kitchen-like appearance) and enables inference about those concepts.

In the place map, the world is represented by a finite number of basic spatial entities called *places* created at equal intervals as the robot moves. Places are connected using paths which are discovered by traversing the space between places. Together, places and paths represent the topology of the environment. This abstraction is also useful for a planner since metric space would result in a largely intractable planning state space.

The places are futher segmented into rooms by detecting doors in the environment. In addition, unexplored space is represented in the place map using hypothetical places called *placeholders* defined in the boundary between free and unknown space in the metric map. Placeholders are used to represent candidate exploration actions to drive exploration.



(a)

(b)

Fig. 1. (a) A place map with several places and 3 detected doors shown as red. (b) Shows two placeholders with different probabilities for turning into new rooms: one of them is behind a door hypothesis therefore having a higher probability of leading into a new room. Colors on circular discs indicates the probability of room categories as in a pie chart: i.e. the bigger the color is the higher the probability. Here green is *corridor*, red is *kitchen* and blue is *office*.

The system employs categorical models which perform abstraction of the sensory information into an environment independent set of spatial properties such as room shape (elongated and square) and appearance (office-like, kitchen-like, corridor-like and meeting room-like). These models are used to infer the room category of each place in the place map.

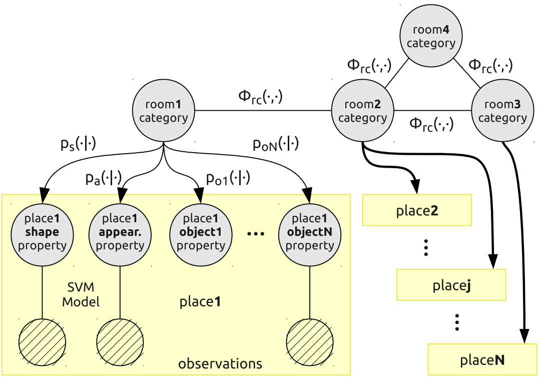


Fig. 2. Schematic image of spatial representation modeled as a chain graph

We use the models proposed in [7] to represent the properties which combine Support Vector Machines with global visual features extracted from camera images (in case of appearance) and simple geometrical features extracted from laser range data (in case of shape).

All higher level inference is done according to a unified model integrating the conceptual knowledge with instance knowledge about the environment. That unified model is expressed using a *chain graph* [4], whose structure is adapted online according to the state of underlying topological map.

The structure of the chain graph model is presented in Fig. 2. Each discrete place is represented by a set of random variables connected to variables representing semantic category of a room. Moreover, the room category variables are connected by undirected links to one another according to the topology of the environment. The potential functions $\phi_{rc}(\cdot,\cdot)$ represent the type knowledge about the connectivity of rooms of certain semantic categories.

The remaining variables represent shape and appearance properties of space and presence of a certain number of instances of objects as observed from each place. These can be connected to observations of features extracted directly from the sensory input. These links are quantified by the categorical models in the place map. Finally, the functions $p_s(\cdot|\cdot)$, $p_a(\cdot|\cdot)$, $p_{o_i}(\cdot|\cdot)$ correspond to size, appearance and object presence of places, respectively. In our work the common sense knowledge is handcrafted, ideally this can originate from various sources [3]. However we note that, any initial probability will be inevitably inaccurate for a specific environment. Therefore, a plausible set of initial probabilities suffices and the emphasis should be on building adaptive representations.

For planning, the chain graph is the sole source of belief-state information. The underlying inference is approximate, and uses the fast Loopy Belief Propagation [5] procedure.

III. Exploration

Each placeholder represents a candidate exploration action. By adding hypothetical places instead of placeholders in the chain graph, we calculate a probability distribution over room categories for each placeholder, which is then used by the planner to decide which placeholder to pursue. The calculation of the probability distribution for placeholders has three steps. In the first step, a set of hypotheses about the structure of the unexplored space is generated. 6 hypotheses for each placeholder are evaluated: (1) placeholder does not lead to new places, (2) placeholder leads to new places which do not lead to a new room, (3) placeholder leads to places that lead to a single new room (4) placeholder leads to places that lead a room which is further connected to another room, (5) placeholder leads to a single new room directly, and (6) placeholder leads to a new room directly which leads to another room. In the second step, the hypothesized rooms are added to the chain graph just like regular rooms and inference about their categories is performed. Then, the probability of any of the hypothesized rooms being of a certain category is obtained. Finally, this probability is multiplied by the likelihood of occurrence of each of the hypothesized worlds estimated based on the amount of open space behind the placeholder and the proximity of gateways. The outcome for a simple case can be seen in Fig. 1(b).

IV. Representing object location

Central to the task of active visual search is the model used to represent the possible locations of sought objects (as well as other objects that may have a relation to a sought object), and the probabilities associated with these locations. The search space is considered to be divided into *locations* $\mathcal{L}$. A location is either a *room* $\mathcal{R}$ or a *related space*. Related spaces are regions connected with a *landmark object* o, either *in* or *on* the landmark. The related space "in" o is termed $\mathcal{I}_o$ and the space "on" o $\mathcal{O}_o$.

The use of spatial relations *on* and *in* provide a meaningful way to cut down the search space. This aspect is crucial since searching everywhere would be highly inefficient. Furthermore, we utilise a natural hierarchical organisation of human-designed spaces, which tend to be conspicuously abundant with both containers and surfaces for placing other objects on. We consider locations to be exclusive, so that if o is in $\mathcal{R}$, an object in $\mathcal{I}_o$ is *not* in $\mathcal{R}$. The hierarchy of object locations structure space efficiently and in a way it's straightforward to communicate to humans. Given a location to be searched, a series of concrete view points must be determined. Let $\Psi_\mathcal{L}$ be the portion of tesselated 3D metric space which corresponds to a location $\mathcal{L}$. We calculate the probability distribution over $\Psi_\mathcal{L}$, $P_\mathcal{L}$, using the perceptual definitions in our previous work [8]. This gives us a distribution over the fine grained metric space. From this distribution, view points (i.e., a camera position and orientation, along with the 3D cone it covers) are calculated so as to cover $P_\mathcal{L}$ to a certain threshold as presented in [1].

We assume that locations contain objects independently of each other and independently of other objects in the same location. Furthermore, we model $P_\mathcal{L}$ over the metric grid map for each location as a Poisson process. A Poisson process describes the outcome of N independent events, in our model each event is the probability of the target object being at a certain cell in $\Psi_\mathcal{L}$. After the robot has processed a sensing action, the probability distribution over the observed location is observed is updated according to our sensor model [1].

V. Planning

In order to exploit the spatial model, a good decision making component is needed. Two different domain independent planners are used for different parts of the task presented in detail [2]: A *classical continual planner* (CP) to decide the overall strategy of the search (for which objects to search in which location) and a *decision theoretic planner* (DT) to schedule the view cone actions using a probabilistic sensing model. Both planners use the same planning model and are tightly integrated. The planner has access to three physical actions: Move can be used to move to a navigational *node*, CreateViewcones creates view cones for a *label* in *relation* to a specified *location*, ProcessViewcone moves to a *viewcone* and uses the vision module to detect the object the cone was created for. There is also the virtual SearchForObject action that triggers the decision theoretic planner. Each action has an associated cost. For the Move and ProcessViewcone action this corresponds to movement cost, SearchForObject's cost is dependent on object's size and CreateViewcones action has a fixed cost.

VI. Experiments

Experiments were carried out on a Pioneer III wheeled robot, equipped with a Hokuyo URG laser scanner, and a camera mounted at 1.4 m above the floor. Experiments took place in 12x8 m environment with 3 different rooms, *kitchen*, *office1*, *office2* connected by a corridor. Target objects (*cerealbox*, *stapler* and *whiteboardmarker*) were trained using [6].

To highlight the flexibility of the planning framework evaluated the system with 6 different starting positions and tasked with finding different objects in an unknown environment. We refer the reader to http://www.csc.kth.se/~aydemir/avs.html for videos. Each sub-figure in Fig. 3 shows the trajectory of the robot.

In the following we give a brief explanation for what happened in the different runs.

- Fig. 3(a) Starts: *corridor*, Target: *cerealbox* in *kitchen*
  The robot starts by exploring the *corridor*. After completing one exploration action, two more placeholders are generated. The placeholder behind the detected doorway has a higher probability of yielding into a kitchen (as opposed to not yielding to any new room) and the robot enters *office1*. As the robot acquires new observations the CP's kitchen assumption is violated. The robot returns to exploring the corridor until it finds the kitchen door. Here the CP's assumptions are validated and the robot searches this room. The DT planner plans a strategy of first finding a table and then the target object on it. After finding a table, the robot generates view cones for the $\mathcal{O}_{table,cornflakes}$ location. The cerealbox object is found.
- Fig. 3(b) Starts: *office2*, Target: *cerealbox* in *kitchen*
  Unsatisfied with the current room's category, the CP commits to the assumption that exploring placeholders in the corridor will result in a room with category kitchen. The rest proceeds as in Fig. 3(a).
- Fig. 3(c) Starts: *corridor* Target: *cerealbox* in *kitchen*
  The robot explores until it finds *office2*. Upon entry the robot categorises *office2* as kitchen but after further exploration, *office2* is categorised correctly. The robot switches back to exploration and since the kitchen door is closed, it passes kitchen and finds *office1*. Not satisfied with *office1*, the robot gives up since all possible plans success probability are smaller than a given threshold value.
- Fig. 3(d) Starts: *office1* Target:*stapler* in *office2*
  After failing to find the object in *office1* the robot notices the open door, but finding that it is kitchen-like decides not to search the kitchen room. This time the *stapler* object is found in *office2*
- Fig. 3(e) Starts: *kitchen* Target: *cerealbox* in *kitchen*
  As before it tries locating a table, but in this case all table objects have been eliminated beforehand; failing to detect a table the robot switches to looking for a counter. Finding no counter either, it finally goes out in the corridor to look for another kitchen and upon failing that, gives up.
- Fig. 3(f) Starts: *corridor* Target: *whiteboardmarker* 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. As before, the robot finds its way to *office1* and launches a search which results in a successful detection of the target object.

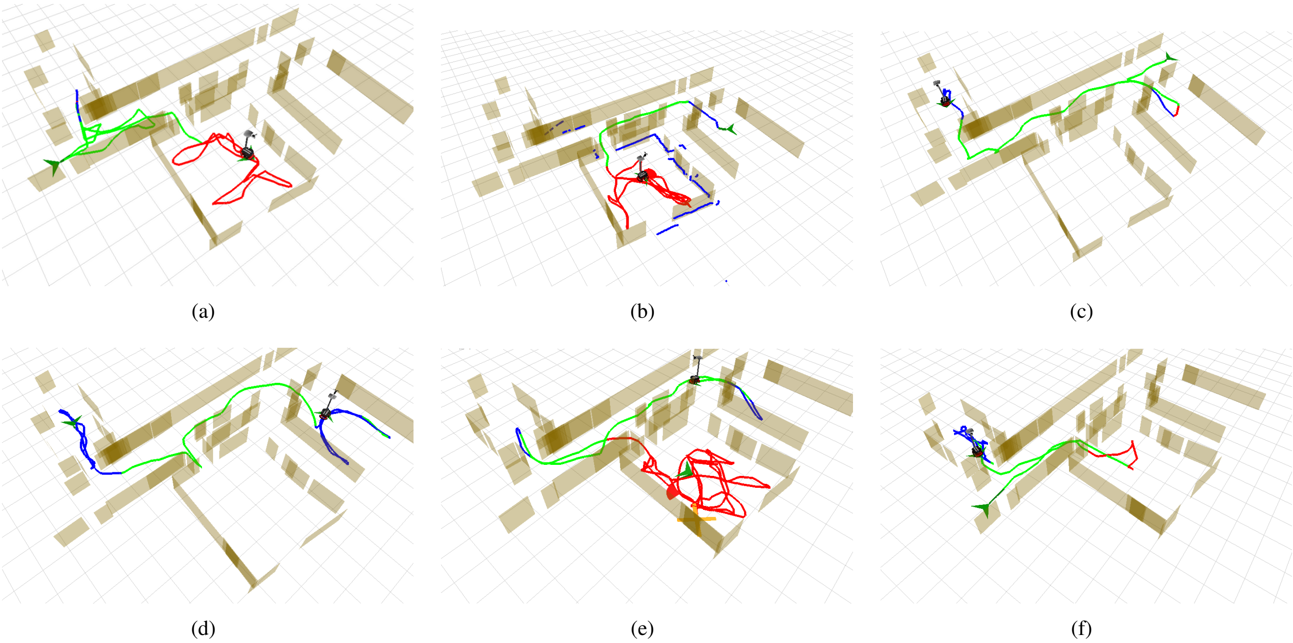
In the following, we describe the planning decisions in more detail for a run similar to the one described in Fig. 3(a), with the main difference being that the cereals could not be found in the end due to a false negative detection.

The first plan, with the robot starting out in the middle of the corridor, looks as follows:

Assume-leads-to-room place1 kitchen
Assume-object-exists table In new-room1 kitchen
Assume-object-exists cerealbox On new-object1 table kitchen
Move place1
CreateViewcones table In new-room1
SearchForObject table In new-room1 new-object1
CreateViewcones cerealbox On new-object1
SearchForObject cerealbox On new-object1 new-object2

Here we see several virtual objects being introduced: The first action assumes that *place1* leads to a new room *new-room1* with category kitchen. The next two assumptions hypothesize that a table exists in the room and that cornflakes exist on that table. The rest of the plan is rather straightforward: create view cones and search for the table, then create view cones and search for the cereal box.

After following the corridor, the robot finds the office, and plans to return to the corridor to explore further. It finally finds a room which has a high likelihood of being a kitchen.


(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3. Trajectories taken by the robot in multiple experiments.The color coded trajectory indicates the room category as perceived by the robot: red is kitchen, green is corridor and blue is office. The green arrow denote the start position of the robot.

Assume-Category room3 kitchen
Assume-object-exists table In room3 kitchen
Assume-object-exists cerealbox On new-object1 table kitchen
Move place17
Move place18
Move place16
CreateViewcones table In room3
SearchForObject table In room3 new-object1
CreateViewcones cerealbox On new-object1
SearchForObject cerealbox On new-object1 new-object2

The new plan does not assume the existence of a new room but the category of an existing one. After view cones are created, the decision theoretic planner is invoked. The DT planner processes view cones until it eventually detects a table and returns to the continual planner.

Assume-object-exists cerealbox On object1 table kitchen
CreateViewcones cerealbox On object1
SearchForObject cerealbox On object1 new-object2

During the run, the continual planner created 14 plans in total, taking 0.2 – 0.5 seconds per plan on average. The DT planner was called twice, and took about 0.5 – 2 seconds per action it executed.

VII. Conclusion and Future Work

We have presented a spatial representation and a planning framework fit for the object search task in large environments. Present work consists using 3D shape cues from depth imaging in order to prime the search process in a single scene and learning over environment topologies to perform more informed exploration.

References

[1] Alper Aydemir, Kristoffer Sjöö, John Folkesson, and Patric Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *Int. Conf. Robotics and Automation (ICRA)*, 2010.

[2] Moritz Göbelbecker, Charles Gretton, and Richard Dearden. A switching planner for combined task and observation planning. In *AAAI 2011*, August 2011.

[3] Thomas Kollar and Nicholas Roy. Utilizing object-object and object-scene context when planning to find things. In *ICRA*, 2009.

[4] S. L. Lauritzen and T. S. Richardson. Chain graph models and their causal interpretations. *J. Roy. Statistical Society, Series B*, 64(3):321–348, 2002.

[5] J. M. Mooij. libDAI: A free and open source C++ library for discrete approximate inference in graphical models. *J. Mach. Learn. Res.*, 11:2169–2173, August 2010.

[6] T. Mörwald, J. Prankl, A. Richtsfeld, M. Zillich, and M. Vincze. BLORT – The Blocks World Robotic Vision Toolbox. In *Proc. ICRA Workshop Best Practice in 3D Perception and Modeling for Mobile Manipulation*, 2010.

[7] Andrzej Pronobis, Oscar M. Mozos, Barbara Caputo, and Patric Jensfelt. Multi-modal semantic place classification. *The Int. Journal of Robotics Research (IJRR)*, 29(2-3), February 2010.

[8] Kristoffer Sjöö, Alper Aydemir, David Schlyter, and Patric Jensfelt. Topological spatial relations for active visual search. Technical Report TRITA-CSC-CV 2010:2 CVAP317, CAS, KTH, Stockholm, July 2010.

[9] J. K. Tsotsos. On the relative complexity of active vs. passive visual search. *International Journal of Computer Vision*, 7(2):127–141, 1992.