2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
Daejeon Convention Center
October 9-14, 2016, Daejeon, Korea

Efficient Object Search for Mobile Robots in Dynamic Environments: Semantic Map as an Input for the Decision Maker

Tiago S. Veiga, Pedro Miraldo, Rodrigo Ventura, and Pedro U. Lima

Abstract— **In this work we study the efficient search of objects in domestic environments, using probabilistic logic to represent uncertainty about object location and partially observable Markov decision processes (POMDP) for the decision-making process regarding the movements to be carried out by the robot to improve its belief about the object locations. We propose the use of a semantic map that stores information about the knowledge in the system and updates it, by an inference process, with sensor information received from the object recognition module. However, semantic maps are not capable of actively search for more information in the environment. For that reason a decision-making module, based on a POMDP framework, is integrated in the system. Several experiments were made in a realistic apartment test bed using every day objects and a mobile robot, showing that this hybrid solution makes the search process more efficient.**

I. Introduction

In order to accomplish more complex tasks robots need to be able to sense their environment and represent it semantically [1]. Moreover, for most of domestic tasks some kind of reasoning about the environment is required and, although humans do this seamlessly, doing it with an autonomous domestic robot is a challenge that has not yet been answered in full.

Semantic maps are a powerful tool to address this issue as they provide a framework to represent knowledge about dynamic environments and can leverage the construction of environment maps where objects can be relocated by external factors (e.g., by human intervention) and uncertainty in observations is an important factor. However, such framework allows to infer from received observations but is not able to actively search for objects. Thus, in this paper we propose a more general framework that combines the elegance of probabilistic knowledge representation and the power of planning under uncertainty with a decision-theoretic framework for an intelligent search for objects in a home environment.

The probabilistic representation of the semantic map is based on the probabilistic logic programming language *Problog* [2]. This language extends *Prolog* with the possibility of attaching probabilities to statements. A probabilistic statement is a Prolog statement with a probability attached, meaning that there is a certain probability of that statement being true. A *Problog* engine can perform several probabilistic inference tasks. In this work we used solely the probabilistic inference of a query given evidence, *i.e.,* observations. For instance, consider this example of Bayesian inference: given a statement, that expresses the probability of observing an object at a given location and an evidence consisting in a positive observation at that location, *Problog* can be used to infer the probability of the object actually being there.

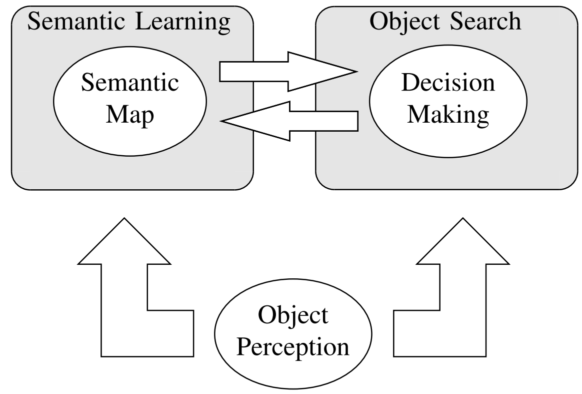


Fig. 1: Information flow of the method proposed in this paper. In our method, the semantic map is used as a input of the decision maker for a proficient object search.

The framework for planning under uncertainty that we consider in this work is the partially observable Markov decision processe (POMDP) [3], that models the interaction of an agent with a stochastic and partially observable environment. Despite their scalability limitation on larger problems, POMDPs have proven to work well in small to medium sized problems by computing approximate solutions [4], [5]. In this work we avoid this limitation by considering smaller POMDPs that model only one room. If the initial belief for each POMDP is obtained from the current probability distribution in the semantic map, then it already includes the past experience on the system. By repeatedly infer from observations and call the respective POMDP controller for each room this process may continue indefinitely and efficiently perform the system's task while overcoming scalability issues.

A. Related Work

Conceptual representations of indoor environments using mobile robots were studied [6], but do not include the flexibility given by our decision-making modules. POMDPs and semantic maps have been previously combined, but only for mobile robot active localization [7]. In turn, we focus

All authors are with Instituto de Sistemas e Robótica, Instituto Superior Técnico, Universidade de Lisboa. This work was partially funded by grant SFRH/BPD/111495/2015. project CMUP-ERI/HCI/0051/2013, FCT project FCT UID/EEA/50009/2013 and EC contract H2020-ICT-688441.
{tsveiga,pmiraldo,rodrigo.ventura,pal}@isr.tecnico.ulisboa.pt

978-1-5090-3761-2/16/$31.00 ©2016 IEEE

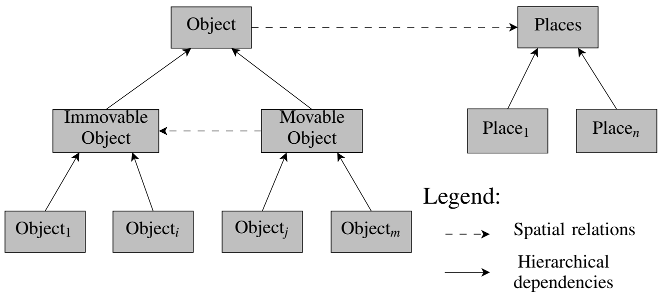


Fig. 2: Example of knowledge base which can be used to represent objects and places in a domestic environment.

on the problem of active object search and perception that has received attention in previous works ([8], [9], [10]) but not providing the integrated framework, capable of dealing with occlusions and with the scalability that our approach offers. Closest to our work, probabilistic conceptual maps and probabilistic planning have been combined in object search tasks [11]. Besides accounting for occlusions by other objects, we are able to simplify the planning phase by separating the decision-making process in different rooms.

B. Contributions

In this paper we provide a unified framework for an efficient search for objects in a domestic environment, composed with three modules: object perception, semantic map, and decision-making. The main contributions of the paper are:

- An integrated system that actively searches for objects in an environment using previously learned knowledge;
- A probabilistic framework for semantic map representation with hierarchical relations;
- A decision-theoretic model for active object search that takes into account possible occlusions due to different object sizes.

We start by giving a general overview of the framework in Sec. II, and proceed to discuss each of the modules in detail in Sec. III. Finally, we will present experimental results and draw final conclusions in Secs. IV and V respectively.

II. General description

The proposed system includes a probabilistic semantic map module that infers its knowledge from information perceived by the perception model and a decision-making module that plans the best next location to search for objects, given the current knowledge of the semantic map. Together, they complement each other and provide a more proficient and scalable framework for object perception. In Fig. 1, we show the information flow through the different models.

In general, the probabilistic semantic map is the module for reasoning in the whole environment, i.e., in our target environments, it maintains information about how the objects of interest are distributed over rooms and inside each of them, as well as providing occlusion rates based on this distribution. This module is passive, providing learning but not planning capabilities. Therefore, the decision-making module provides the system with decisions for an intelligent search for objects. This module decides how to act locally, inside each room, which makes our approach more scalable since we may have many simple decision-making modules instead of a large complex one. It receives an initial distribution and occlusion rates from the semantic map and returns information about which objects are located or not inside the actual room. Both modules communicate with the object perception module, which returns the set of objects visible in front of the robot at each particular moment.

In section III we will provide a more detailed description of each module and discuss its particular features.

III. Detailed description

A. Object Perception

The object recognition module is based on the 3D recognition framework of the Point Cloud Library [12] and comprises a training and recognition sub-modules. Point clouds were acquired with an RGB-D camera on top of the robot. We provide a brief description of each of the sub-modules:

- The training sub-module imports models for each object class which are then rotated and converted to point clouds from each different view. Several keypoints and descriptors are extracted;
- The recognition process segments the scene's point cloud, filters the area of interest and applies a clustering algorithm to the remaining point cloud. Finally, a recognizer algorithm, previously trained with the imported models, is run to detect objects of interested.

B. Semantic Map

The semantic map is divided in two main parts: the knowledge base and the reasoning engine. The knowledge base stores the acquired information as an abstract knowledge concept or as information regarding a specific instance in the environment. The reasoning engine is where the information from the perception module is processed and integrated into the knowledge database.

1) Knowledge Base: The knowledge base accommodates three types of knowledge: knowledge instances, object information type and relations. Knowledge instances are used to represent common sense information about specific concepts, object information types are the physical representation of the concept and relations represent the interconnections between knowledge instances. Figure 2 represents an example of a knowledge base that categorizes objects and places. Here, any given object can be immovable (e.g., table) or movable (e.g., cereal box) and be in a given place. Besides, we may easily understand the spatial relations between instances. For example, although not being a place a cereal box may be on a table which, in turn, is located in a given room. By building instances with these kind of relations the semantic map represents its knowledge.

2) Reasoning Engine: The reasoning engine is based on the *Problog* language. For better visualization, an example of the code used is shown in Figure 3. The first block contains properties of objects, e.g., a coke is a sodacan,

```
sodacan(coke).                    cereal_box(cereal).
volume(coke, 6.60).               volume(cereal, 16.60).
mobility(coke, 1).                mobility(cereal, 1).

 movable_object(X) :- cereal_box(X).
 movable_object(X) :- sodacan(X).
         object(X) :- movable_object(X).
immovable_object(X) :- table(X).
     occlusion(X,Y) :- volume(X,VX), volume(Y,VY), VX>VY.

0.33::is_in(X,counter) ; 0.33::is_in(X,table) ; 0.33::is_in(X,bench) :- object(X).
P::seen(X,Y,T1,T2) :- is_in(X,Y), mobility(X,M), P is 0.6+0.1exp(-(T1-T2)/10M).
P::seen(X,Y,T1,T2) :- \+is_in(X,Y), mobility(X,M), P is 0.30.1exp(-(T1-T2)/10M).
```

Fig. 3: *Problog* code representing the proposed reasoning engine. See text for an overview and a description of its structure, Section III-B.2.

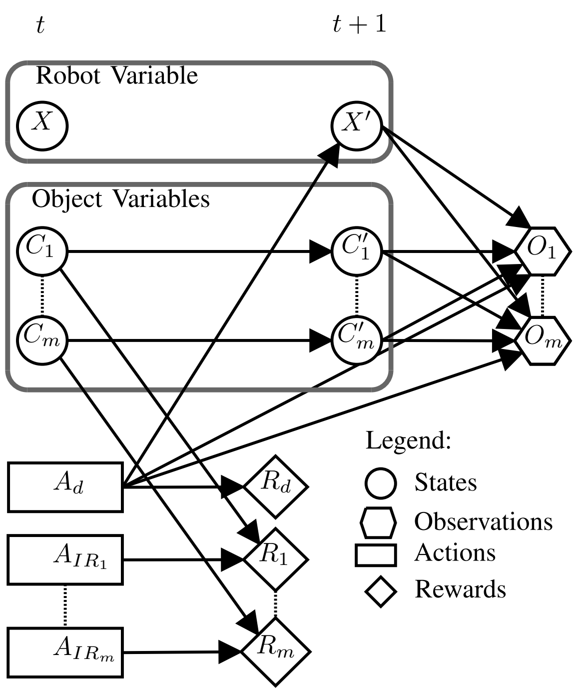


Fig. 4: Two-stage dynamic Bayesian representation of the proposed POMDP model for the decision-making module.

the volume of a cereal box is 16.60, etc. The second block represents deterministic relations between predicates, e.g., a cereal box is a movable object, a movable object is an object, etc. The third block models the probabilistic rules: the first one states a uniform *a priori* distribution of locations for any object X, while the second and third ones represent a probabilistic observation model conditioned to the presence of the object, e.g., if object X is on location Y, it has a probability of $P = 0.6 + 0.1e^{-(T_1-T_2)/(10M)}$ of being seen, where $T_1 - T_2$ is the time elapsed between observations, and M is an heuristic mobility degree. The two last rules correspond to a true positive and to a false positive case respectively ('\+' denotes logical negation).

C. Decision Making

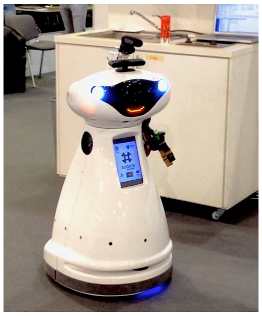
The decision-making module is modeled under the POMDP-IR [13] that allows to model standard POMDPs for information gain. We present in Figure 4 a dynamic Bayesian network representation of the model used to capture the dynamics of the problem that follows the POMDP-IR formulation. In the following, we present a more detailed description of this model.

1) States and transitions: The model considers two different types of variables, dubbed robot and object variables. In particular, a robot variable X models the location of the robot in the environment and object variables $C_1 \dots C_m$ model the location of each object of interest, where m is the number of objects currently existing in the semantic map. In this model the robot is allowed to move to k waypoints that represent the position of immovable objects. To account for occlusions there must be at least two opposite waypoints around each immovable object. Objects can be located in any of the same k waypoints or plus an additional $none$ position. This additional location is used to take into account the possibility that a given object is not in the actual room.

For the purpose of our task we consider a simple deterministic state transition model both for robot movement and object location. In practice, this means that we consider that the robot's navigation is always successful and that objects remain static in their initial position during each run. Those are realistic assumptions given that, in one hand, nowadays there are reliable and accurate navigation algorithms that work well in our kind of environments and can be separated from the decision-making task. In the other hand, in these environments it is not expected that objects are relocated too often while the robot performs a search task and, even if it happens, it will be detected in the following searching episode.

2) Observations and occlusion model: The definition of observation variables and their respective observation model are the key for the behavior we want to achieve with this model, as the algorithm will be guided by them to plan the best perception actions. Thus, the model includes one observation variable O_i for each object of interest ($1 \leq i \leq m$). Every time a perception action is triggered each of the observation variables takes a value yes or no, indicating whether that object was detected by the perception module or not.

The observation model considers all possible occlusions



(a)

(b)

Fig. 5: Representation of the setup used in the paper, for the experimental results. Fig (a) shows a picture of the robot used in the experiments and Fig. (b) shows the test bed (which is usually used for @Home Robot Competitions), installed in the ISR/IST's facilities.

between objects. That is, every time the perception model is called the decision maker takes into account all different phenomena that may affec the perception algorithm. First, there is an error rate associated with vision algorithms. Second, there is a probability that an object is occluded behind another in the same location, according to the proportion between their sizes. In our model, we consider that an object may be occluded only if it is smaller than some other given object.

3) Actions: Following the POMDP framework used, there are two kind of actions: domain A_d and information-rewarding A_{IR_i} actions. Domain actions are those which physically interact with the environment, that in our model includes moving actions, that guide the robot through the different locations of interest in the environment, a perception action that calls the perception module and perceives objects in the actual location, and an additional $doNothing$ actions that indicates the end of a searching episode.

The set of information-rewarding actions include one for each object of interest. This extra actions indicate whether an object is believed to be in some location in the actual room, not found in this room, or $null$ if there is not enough information.

4) Reward model: The set of rewards used follow the set of actions previously mentioned. Therefore, the model implements a small cost of 0.5 for each moving action and 0.1 for a perception actions. There is neither a cost nor reward for $doNothing$ action.

For the reward model of information-gaining actions, we use the reward model equivalent of a belief threshold of $\beta = 0.9$, meaning that an informative-gaining action will be triggered every time the belief for that particular state variable (object) is above that threshold. This will guide the system to increase the available information about object locations.

IV. Experiments

For the experimental results we considered a robot in a realistic scenario (a test bed representing a real apartment), which was asked to find some objects.

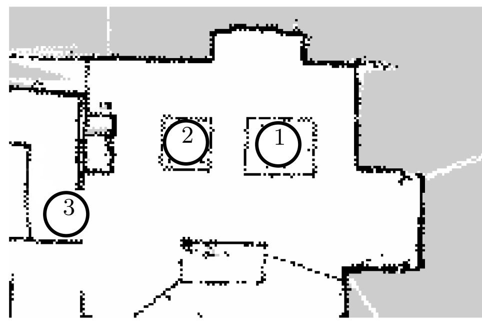


Fig. 6: Kitchen map. Locations: 1 - *Table 1*; 2 - *Table 2*; 3 - *Entrance*

A. Experimental setup

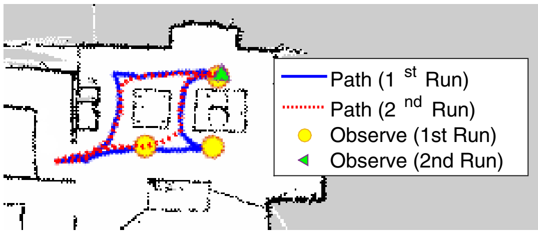
For the experiments with real data, we use the MBot platform (see Fig. 5(a)), in a test bed, located at the Institute for Systems and Robotics, IST, Lisbon[1] (see Fig. 5(b)). In particular, our experiments consider the kitchen area of the house, which can be visualized in Fig. 6. Here, there are two tables where we can find two particular objects: a cereal box and a coke can. All sensors are onboard and all processing is autonomously performed in the robot computers.

B. Experimental results

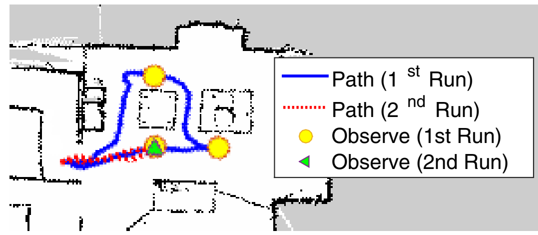
We experimented the system with different object locations in 4 different tests. The robot's path and the evolution of the system knowledge, represented as the belief of each object, for each experiment are shown in Figure 7. Every experiment consist of two consecutive runs. Note that the belief at the beginning of each run is the one inferred by the semantic map and sent to the POMDP controller. In the first run this is always an uniform belief, while in the second run it is already updated according to the observations made during the first run.

In experiment A we tested the system's behavior when both objects were on *Table 1*. In the first run the robot stops at both tables to search for objects. In particular, note that on

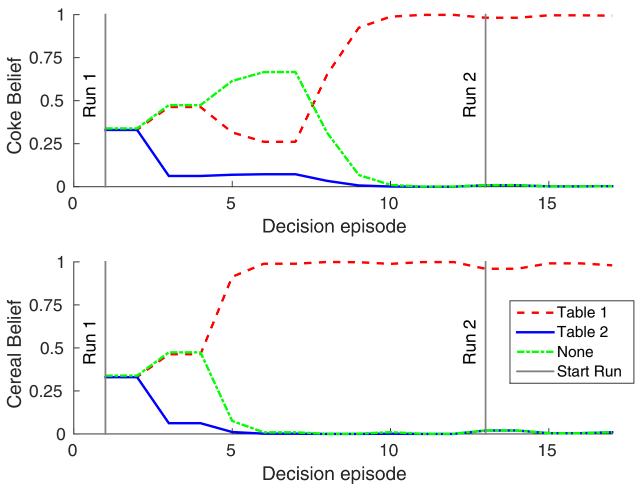
[1]http://welcome.isr.tecnico.ulisboa.pt/isrobonet/



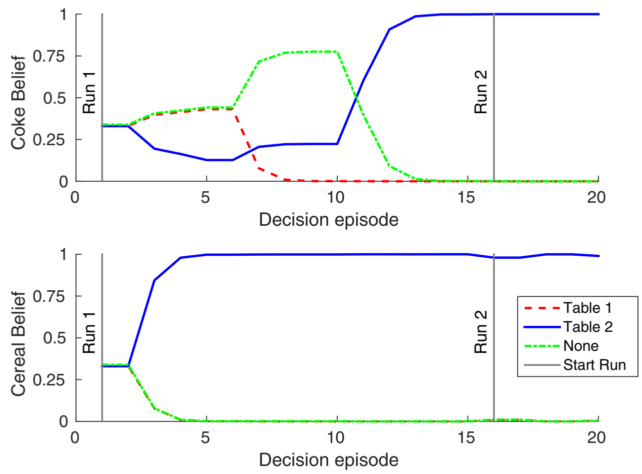
(a) Exp.A: Robot's Path



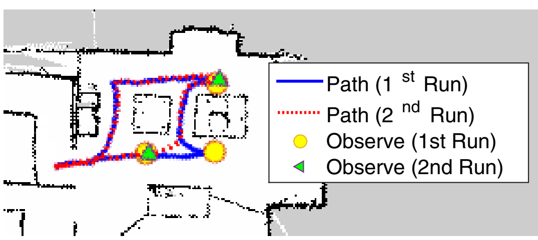
(b) Exp.B: Robot's Path



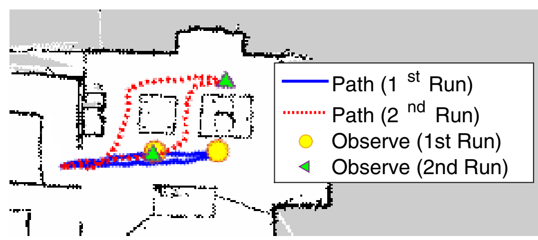
(c) Exp.A: Belief Evolution



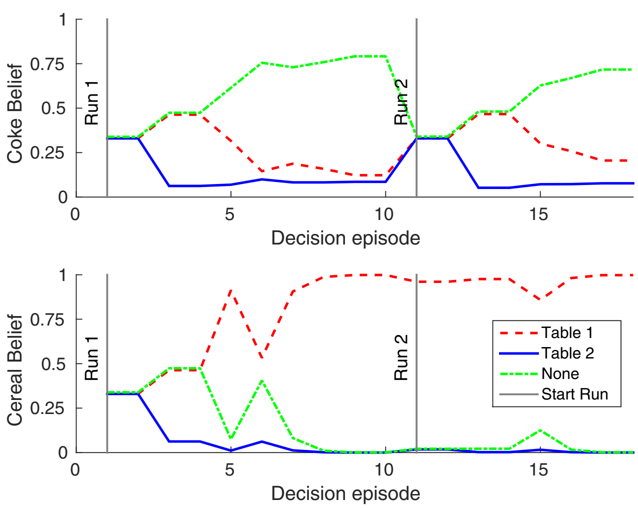
(d) Exp.B: Belief Evolution



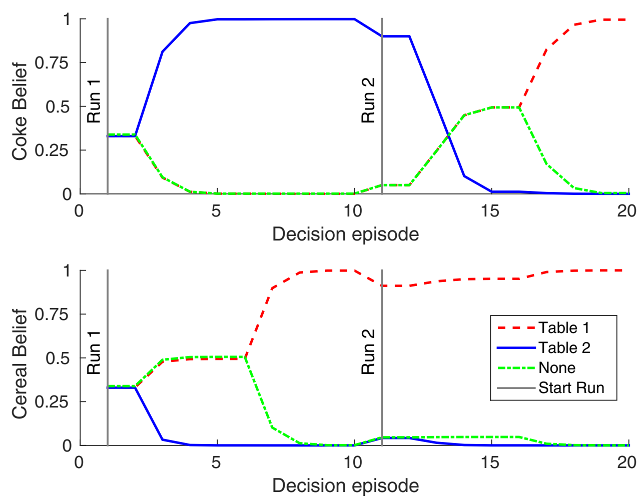
(e) Exp.C: Robot's Path



(f) Exp.D: Robot's Path



(g) Exp.C: Belief Evolution



(h) Exp.D: Belief Evolution

Fig. 7: ***Exp.A*** Cereal box: *Table 1*, Coke can: *Table 1*. ***Exp.B***: Cereal box: *Table 2*, Coke can: *Table 2*. ***Exp.C***: Cereal box: *Table 1*, Coke can: *none*. ***Exp.D***: Cereal box: *Table 1*, Coke can: *Table 2* in Run 1, *Table 1* in Run 2.

Table 1 it first detects only the cereal box and then moves to the opposite side of the table, predicting a possible occlusion that indeed is happening. It is also observable that the belief evolution shows that the system can retrieve the location of the objects. When we start the second run the semantic map updates its information and sends a new and very confident initial belief. Therefore, the decision-making module only decides to search in *Table 1* and immediately ends the run. As a consequence, the second run has less timesteps that the first one.

Experiment B is similar to the first one, with the exception that both objects are now located on *Table 2*. Then, the robot searches in both tables, going to both sides of *Table 2* to prevent occlusions. In the second run it goes directly to *Table 2* and immediately returns to the entrance, given the high certainty on the information available. Again, this means that the second run is significantly shorter than the first in the number of timesteps.

In experiment C a different scenario was tested, in which a cereal box was placed on *Table 1* and the coke can was left out of the scene. Here, the robot searches on *Table 2* and in both sides of *Table 1* in the first run, to find that the cereal box is on *Table 1* and coke most likely is not in the room. Since the semantic map updates its belief based on what it observes, in the second run it sends a low uncertainty initial belief about the cereal but a high uncertainty belief about the coke. Thus, in the second run the system's behavior is similar to the first with exception that, since it only has a high uncertain information about the coke can position, it primarily searches for that object and takes less timestep in the search process.

Finally, in experiment D we introduced some dynamic in the environment and changed the location of objects between runs. Namely, the cereal box was located on *Table 1* during both runs while the coke can was located on *Table 2* during the first run and moved to *Table 1* in the second run. We may note that the system is able to deal with this change in the environment. Although the initial belief sent by the semantic map shows a low uncertainty belief about both objects, the decision-making module still checks their location to find that the coke is not anymore in the place where its belief was higher. That fact leads the decision-making module to drive the robot to search in other locations, more accurately updating its belief during the second run.

V. Conclusions

In this work we presented an integrated framework that combines the representation potential of reasoning with probabilistic logic with the power of decision-making under uncertainty with decision-theoretic methods. By establishing a communication between both modules we are able to perform tasks that involve search for objects in an environment, given the relation between objects built in the semantic map and given the level of knowledge about positions of objects at every moment.

We show with real experiments that such framework works well in practice, being reliable and efficient in its task. Moreover, this is a scalable approach that overcomes one of the main challenges in planning for these kind of tasks. For larger problems it is possible to build a semantic map for the whole environment with a POMDP model for each room, building a hierarchical system. This allows to overcome the known scalability issues on POMDP solving, as each model remains limited to each room and with a small size, and the lack of intelligent decisions on the semantic map reasoning engine. Also, they extend semantic mapping capabilities by allowing the system to do an intelligent search for objects, rather that only reasoning. The communication between both modules is natural since both represent their knowledge as probability distributions, or beliefs, over possible locations of objects.

In future work, we would like to extend this work to larger problems. For instance, by experimenting in a house with several divisions where the complexity of the relation between objects and locations is higher. Multi-robot systems may also provide a challenging environment where cooperation must be taken into account.

References

[1] D. Pangercic, B. Pitzer, M. Tenorth, and M. Beetz, "Semantic object maps for robotic housework-representation, acquisition and use," in *Proc. of International Conference on Intelligent Robots and Systems*, 2012, pp. 4644–4651.

[2] L. De Raedt, A. Kimmig, and H. Toivonen, "Problog: A probabilistic prolog and its application in link discovery." *Proc. Int. Joint Conf. on Artificial Intelligence*, vol. 7, pp. 2462–2467, 2007.

[3] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[4] M. T. J. Spaan and N. Vlassis, "Perseus: Randomized point-based value iteration for POMDPs," *Journal of Artificial Intelligence Research*, vol. 24, pp. 195–220, 2005.

[5] G. Shani, P. Poupart, R. I. Brafman, and S. E. Shimony, "Efficient ADD operations for point-based algorithms," in *Int. Conf. on Automated Planning and Scheduling*, 2008.

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

[7] S. Li, R. Xiong, and Y. Wang, "Active semantic localization of mobile robot using partial observable monte carlo planning," in *Proc. Int. Conference on Control and Automation*, 2013, pp. 1409–1414.

[8] N. Atanasov, B. Sankaran, J. L. Ny, G. J. Pappas, and K. Daniilidis, "Nonmyopic view planning for active object classification and pose estimation," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1078–1090, 2014.

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

[10] M. Sridharan, J. L. Wyatt, and R. Dearden, "Hippo: Hierarchical pomdps for planning information processing and sensing actions on a robot." in *Int. Conf. on Automated Planning and Scheduling*, 2008, pp. 346–354.

[11] M. Hanheide, C. Gretton, R. Dearden, N. Hawes, J. Wyatt, A. Pronobis, A. Aydemir, M. Göbelbecker, and H. Zender, "Exploiting probabilistic knowledge under uncertain sensing for efficient robot behaviour," in *Proc. Int. Joint Conf. on Artificial Intelligence*, vol. 22, no. 3, 2011, p. 2442.

[12] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *IEEE International Conference on Robotics and Automation*, Shanghai, China, 2011.

[13] M. Spaan, T. Veiga, and P. Lima, "Decision-theoretic planning under uncertainty with information rewards for active cooperative perception," *Autonomous Agents and Multi-Agent Systems*, vol. 29, no. 6, pp. 1157–1185, 2015.