Action-Based Models for Belief-Space Planning

Li Yang Ku, Shiraj Sen, Erik G. Learned-Miller, and Roderic A. Grupen
School of Computer Science
University of Massachusetts Amherst
Amherst, Massachusetts 01003
Email: lku, shiraj, elm, grupen@cs.umass.edu

***Abstract*—Autonomy requires robots to learn models of the environment or objects while simultaneously searching for solutions in the partially observable state space. A flexible representation that supports incremental acquisition of models has several advantages for solving such tasks. In this paper, we investigate the use of graphs to capture the interaction statistics of an agent with aspects of the environment. We present a planner that employs a set of incomplete models for action selection. The approach is evaluated using the Robonaut 2 simulation in the context of object modeling and planning.**

I. Introduction

An intelligent agent must reason about its own skills, and about the relationship between these skills and goals under run-time conditions. This requires the agent to represent knowledge about its interactions with the world in a manner that supports reasoning [2]. Since the early 1970s, the AI and robotics communities have been concerned with the design of efficient representations that support modeling and reasoning. However, most of these representations tend to tackle only one part of the problem—making either the modeling or the reasoning problem easier.

This paper addresses these dual problems of modeling and reasoning by employing a representation grounded in the robot's own actions and perceptions [1]. Our description of state is domain general, as it is computed directly from the status of executable actions and not hand built for a specific task. The relationship between state and action is captured using probabilistic data structures that model objects in the environment [4]. We present a planner that exploits the uniform description of state and the probabilistic models to plan efficiently in partially observed environments.

II. Model Learning

An object in our framework is represented using a directed graph $G = (\mathcal{X},\mathcal{U})$, composed of a set of aspect nodes $\mathcal{X}$ connected by a set of action edges $\mathcal{U}$ that capture the probabilistic transition between the aspect nodes. Each aspect $x \in \mathcal{X}$ represents the properties of an object that are measurable given a set of sensor parameters. We call this graph that summarizes empirical observations of the aspect transitions in the course of interaction an Aspect Transition Graph (ATG).

The ATG of an object is complete if it contains all possible aspect nodes and node transitions. However, in practice, when ATGs are learned through exploration they are often incomplete. In addition, an object might be represented by multiple (incomplete) ATGs. A complete model is more informative but harder to learn autonomously. In this paper, we will focus on handling incomplete object models.

An ATG is added to the robot's memory $\mathcal{M}$ only if the presented object is inferred to be novel. Let $\mathcal{S}_{T-1}$ denote the set of objects that have been presented to the robot in the first $T-1$ trials. Given a sequence of observations $z_{1:t}$ and actions $a_{1:t}$ during trial T, the probability that the presented object O_T during trial T is novel can be calculated;

$$\begin{aligned} &p(O_T \notin \mathcal{S}_{T-1} | z_{1:t}, a_{1:t}, \mathcal{M}) \\ &\quad = \sum_{o_i \notin \mathcal{S}_{T-1}} p(O_T = o_i | z_{1:t}, a_{1:t}, \mathcal{M}) \\ &\quad = \sum_{o_i \notin \mathcal{S}_{T-1}} \sum_{x_t \in \mathcal{X}_i} p(x_t | z_{1:t}, a_{1:t}). \end{aligned} \tag{1}$$

The set $o_i \in \mathcal{O}$ consists of all the objects in the environment. The set $x_t \in \mathcal{X}_i$ consists of all the aspects generated form object o_i. The conditional probability $p(x_t|z_{1:t}, a_{1:t})$ of observing an aspect can be inferred through the Bayes filter algorithm [5]. The presented object O_T is classified as novel if $p(O_T \notin \mathcal{S}_{T-1}|z_{1:t}, a_{1:t}, \mathcal{M}) > 0.5$.

III. Task-Level Planning

The challenge of integrating task-level planners with partial models requires dealing with the partial observability of the state while building plans. Since the true state of the system cannot be observed, it must be inferred from the history of observations and actions. Our planner belongs to a set of approaches [3] that select actions that reduce the uncertainty of the state estimate maximally with respect to the task.

Object recognition can be viewed as a process in which the uncertainty over object identities (as quantified by the object entropy) is reduced with each observation. Our task planner selects the action a_t that minimizes the expected entropy of the random variable O_T representing the object identity;

$$\begin{aligned} &\operatorname*{argmin}_{a_t} \; E(H(O_T | z_{t+1}, a_t, z_{1:t}, a_{1:t-1})) \\ &\quad = \operatorname*{argmin}_{a_t} \sum_{z_{t+1}} H(O_T | z_{t+1}, a_t, z_{1:t}, a_{1:t-1}) \times \\ &\qquad p(z_{t+1} | a_t, z_{1:t}, a_{1:t-1}). \end{aligned} \tag{2}$$

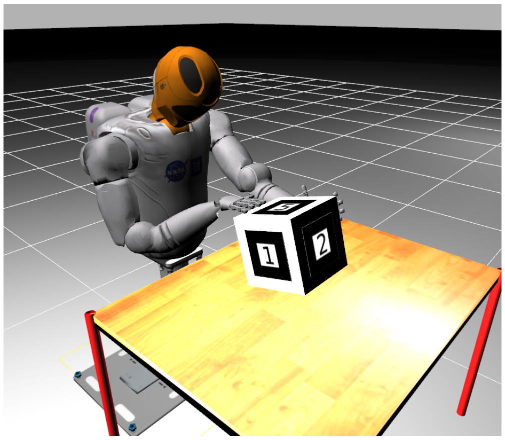

Fig. 1. The simulated Robonaut 2 interacting with a box.

IV. Experiments

We evaluated the capabilities of our model and planner using a Robonaut 2 simulator as shown in Figure 1. The simulation contains 100 unique objects called ARcubes that consist of a $28cm$ cube with unique combinations of ARtags on the six faces; 12 different ARtag patterns are used in this experiment. In an ATG for an ARcube, an aspect consists of ARtag features observed on 2 faces. Each ATG has 24 unique aspects and each aspect has 132 different pattern combinations. For the sake of simplicity, we assume that an object does not have two faces with the same ARtag. The robot can perform 3 different manipulation actions on the object: 1) flip the top face of the cube to the front, 2) rotate the left face of the cube to the front, and 3) rotate the right face of the cube to the front.

Table I shows the result of using the planner to recognize the object presented. Each test involves 100 trials and starts with an empty robot memory $\mathcal{M}$. In each trial, the task is to decide which ATG in memory the experiment corresponds to or to declare it as a new object. For each trial, an object is chosen at random and presented to the robot. The robot observes the object and executes an action. This process is repeated 20 times. At the end of each trial the robot determines the likelihood that the presented object is novel and the most likely existing object in memory is identified.

The last row in Table I presents the results averaged over all the tests. The success rate is the percentage of objects correctly classified, that is, correctly identified in memory or declared as a novel object. The system correctly recognizes the object 100% of the time, and correctly determines if the presented object is novel or not 98.8% of the time.

We also tested the efficiency of the planner against a random policy. The number of actions executed per trial were varied from 4 to 20. Figure 2 shows how the success rate of a test varies with the number of actions executed per trial. As is evident from the plots, the information theoretic planner outperforms a random exploration policy for all cases except when the number of actions per trial is low. Both algorithms perform equally poor when not enough information is provided.

TABLE I
THE SUCCESS RATE OF AN INFORMATION THEORETIC PLANNER IN RECOGNIZING THE OBJECT (20 ACTIONS PER TRIAL)

Test	Correct Identification	Correct Recognition	Success Rate
1	100/100	34/34	100%
2	98/100	32/32	98%
3	98/100	40/40	98%
4	99/100	37/37	99%
5	99/100	32/32	99%
average	98.8%	100%	98.8%

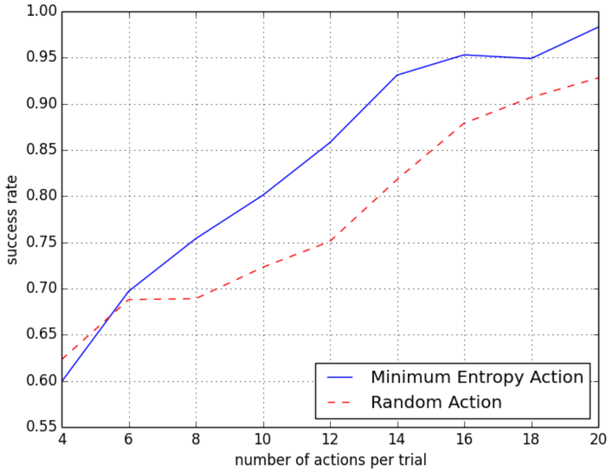

Fig. 2. The plot shows the average success rate of 10 tests as the number of actions per trial are increased. Selecting actions that minimize entropy leads to a higher success rate then selecting actions at random.

V. Conclusion

This paper describes an incremental learning framework for building a memory of objects through interaction. We presented a Bayes framework that performs inference over incomplete object models. We then showed the strengths of combining this representation with a Belief-space planner. For future work, we are planning to test our algorithm on objects without fiducial markers and are interested in studying when to merge incomplete object models from different trials. We are also exploring how a belief space planner can be used to solve tasks involving multiple objects in the scene and extensions of the idea that can incorporate multi-modal sensory features like tactile data and temporally extended actions.

Acknowledgments

This material is based upon work supported under Grant NASA-GCT-NNH11ZUA001K.

References

[1] S. Hart and R. Grupen. Learning generalizable control programs. In *Transactions on Autonomous Mental Development*, pages 1–16, Zaragoza, Spain, 2010.

[2] Leslie Pack Kaelbling and Tomás Lozano-Pérez. Hierarchical task and motion planning in the now. In *IEEE Conference on Robotics and Automation Workshop on Mobile Manipulation*, 2010.

[3] R. Platt, R. Tedrake, L. Kaelbling, and T. Lozano-Perez. Belief space planning assuming maximum likelihood observations. In *Proceedings of Robotics: Science and Systems*, Zaragoza, Spain, June 2010.

[4] S. Sen and R. Grupen. Manipulation planning using model-based belief dynamics. In *Proceedings of the 13th IEEE-RAS International Conference on Humanoid Robots*, Atlanta, Georgia, October, 2013.

[5] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.