Summary of Experiments in Belief-Space Planning at the Laboratory for Perceptual Robotics

Scott M. Jordan, Dirk Ruiken, Tiffany Q. Liu,
Takeshi Takahashi, Michael W. Lanighan, Roderic A. Grupen
College of Information and Computer Sciences
University of Massachusetts Amherst
{sjordan, ruiken, tliu, ttakahas, lanighan, grupen}@cs.umass.edu

Abstract

In robotics, observations and actions are rarely certain, making it difficult to plan and take actions to reach goal states. Belief-space planners are often used to overcome the uncertainty and partial observability, but make it difficult to define generic tasks. In recent work, we presented a planner and task representation framework capable of alleviating these problems. This paper is a summary of our past work on belief-space planning and task representations.

Introduction

Robotics tasks involve planning to reach a goal state. In partially observable environments and in the presence of uncertainty, the task requires accumulating certainty of the goal state. Planning needs to be done over the probability distribution of the state space. Belief-space planners deal directly with the uncertainty by using a Bayes filter to update belief of the state space. Rolling out just a few actions results in a large number of probabilistic future states. Belief-space planners with fixed horizons are able to plan for this problem, but planning time is still an issue due to exponential state expansion. Applying belief-space planners to robotics is challenging because there is a lack of a unifying framework for representing tasks that are usable by belief-space planners.

The belief-space planner in this work belongs to a class of information theoretic planners that uses information gain to reduce uncertainty and trim unnecessary rollouts. We address the task representation problem by partitioning the set of states, and the task is defined as condensing the belief on a desired subset. This task framework automatically balances taking actions to reduce uncertainty and making progress to complete the task. We demonstrate that the same framework is capable of handling several task types such as finding, manipulating, and arranging objects required for a copy task based on visual perception and observation of transition dynamics.

Related Work

Due to partial observability and uncertainty, it is often necessary to have multiple viewpoints or interactions with an object to recognize it. Deinzer *et al.* use reinforcement learning to learn a policy to select viewpoints for recognition (Deinzer et al. 2009). Eidenberger and Scharinger formulate an approximate solution as a partially observable Markov decision process (POMDP). They demonstrated that this approach generates next viewpoint actions that successfully recognize multiple objects in a cluttered scene (Eidenberger and Scharinger 2010).

A necessary component of a belief-space planner is a means of propagating belief distributions through candidate actions using a forward model. For example, Hogman *et al.* use the action-effect relation to categorize and classify objects (Hogman et al. 2015). Sen introduces affordance-based object models called aspect transition graphs (ATGs) that combine bag-of-features feature matching with a graph to model action effects (Sen and Grupen 2014). Ruiken *et al.* extend the ATGs by adding geometric information and cost estimates to improve forward modeling capabilities (Ruiken et al. 2016b).

Often the robotics community works on when to switch between tasks rather than how to solve different active perception tasks using a single planner (Wawerla and Vaughan 2009; Capi 2007). Some methods have been proposed to solve multiple tasks with a single planner, but are not active recognition algorithms, and therefore, they do not interact with the environment to reduce the uncertainty (Grabner, Grabner, and Bischof 2005), (Lai et al. 2011). We combine active perception with the ability to switch between tasks (Ruiken et al. 2016a).

Technical Approach

In this section we describe the three main components of our architecture: object models, the Active Belief Planner (ABP), and the task framework.

Object Model

A Dynamic Bayes Net (DBN) is used as a recursive, hierarchical inference engine in which objects o generate abstract states x that then generate observations of aspects z. Each unique set of features of an object that can be seen from a single viewpoint, together with its geometric constellation, form an aspect z. Features in our system have a type and Cartesian attribute. Pairwise distances between features are used with Hough voting (Ballard 1981) to determine which aspects are present.

Each object o is modeled by a multi-graph called an aspect transition graph (ATG). Graph nodes in the ATG are called aspect nodes, and edges are transitions between aspect nodes. An aspect node is associated with an aspect and represents the abstract state x of the robot with respect to an object. Each transition has an associated, parametrized action with cost estimates. An edge defines a forward model $p(x_{t+1}|x_t, a_t)$ as the probability of transitioning from one aspect node to another for a given action. For example if a box has identical front and back sides, there is an aspect node for each front and back of the object where both nodes expect to see the same aspect.

Using aspect nodes of an ATG as an abstract state x greatly reduces the state space of the problem. The ATG also contains all relevant (known) actions to interact with the object. Therefore, out of all possible action parameterizations, only useful ones provided by the ATG need to be considered. Additionally, the ATG provides forward and observation models for belief update and planning.

Active Belief Planner (ABP)

The purpose of the planner is to select actions that provide the most information by rolling out possible future states. The forward models from all ATGs are used to propagate belief over multiple actions. To handle multiple objects, features are clustered into spatial hypotheses h_k based on the compatibility of their spatial distributions with known object models. The planner can then probabilistically reason over one object hypothesis at a time. The number of hypotheses is expected to be roughly the number of objects in the scene. For each of the k hypotheses, given a belief at time t over aspect nodes $bel(x_t^k)$ and the executed action a_t, the belief is updated by

$$\overline{bel}(x_{t+1}^k) = \sum_{x_t^k} p(x_{t+1}^k|x_t^k, a_t) bel(x_t^k),$$

where $\overline{bel}$ denotes that the posterior is due solely to action a_t. The planner then performs an observation update based on the expected observations provided by the ATG, yielding the posterior belief with η as a normalizer:

$$bel(x_{t+1}^k) = \eta\, p(z_{t+1}^k|x_{t+1}^k)\, \overline{bel}(x_{t+1}^k).$$

From this posterior belief, the planner evaluates all candidate actions and predicts the most informative next action. After this action is executed, new observations are matched to aspect nodes to calculate $p(z_{t+1}^k|x_{t+1}^k)$ based on the geometric constellation of features and observation covariances.

The algorithm has a complexity of $O(|K||A||X|^2)$, where K is the set of independent hypotheses, A is the set of eligible actions for each hypothesis, and X is the set of aspect nodes. The time required to expand all belief nodes is dependent on the distribution of belief and quickly decreases when the belief condenses on fewer aspect nodes. The search depth of the algorithm is variable and is automatically increased as belief condenses, and forward planning becomes less expensive. For more detailed information on the algorithm, we refer the reader to (Ruiken et al. 2016b; 2016a).

Task Framework

The ABP can plan over any level of the hierarchical DBN (objects, aspect nodes, or features). Assuming a “complete” ATG for all objects in the model space, any task that can be performed using actions comprising the edges in the ATG can be specified by defining a partition C over aspect nodes of the ATG. This partition aggregates belief on the aspect nodes into targeted subsets for the task. Most tasks result in a partition with two subsets: all aspect nodes that *do* and that *do not* satisfy the task specifications. For other tasks, the aspect nodes may be split into n different subsets to, for example, recognize an object within a model space of n objects.

The belief over the partition C can be calculated by summing the belief over aspect nodes contained in each subset c:

$$bel(c) = \sum_{x \in c} bel(x).$$

Standard information-based metrics can be applied in a belief-space planner to choose the next best action. The choice of the metric changes the behavior of the robot. For example, minimizing the entropy,

$$H(c_t) = -\sum_{c_t} bel(c_t) \log\left(bel(c_t)\right),$$

causes the belief-space planner to pick actions that efficiently condense belief into the subset c that best represents the history of observations. If the model space contains the correct object, this corresponds to a recognition task. Alternatively, a target distribution $T(c)$ can be specified over all c. In this case, minimizing the Kullback-Leibler (KL) divergence between $T(c)$ and the current belief $bel(c_t)$,

$$D_{KL}(T(c)||bel(c_t)) = \sum_{c_t} T(c) \log\left(\frac{T(c)}{bel(c_t)}\right),$$

results in actions that steer the robot toward the target state(s) while automatically balancing information gathering actions and actions towards the task goal. Tasks defined this way are most general and can include recognition at the object and aspect node levels.

Task Types

In this section, we define four basic task types commonly found in robotics. These are only samples of possible tasks that can be represented in this framework; tasks are only limited by the expressiveness of the known ATG model. Each type can be differentiated by the way the task partition defines the task for the planner. A graphical example of task partitions for the four task types is shown in Figure 1.

In a *recognition* task, the robot is presented with one or more object(s) of unknown identity. The robot has ATG models for n different objects and has to identify the probability that the data supports each of the known ATG models. The robot can use any action present in all of the ATGs to investigate and manipulate the object(s). The aspect nodes are

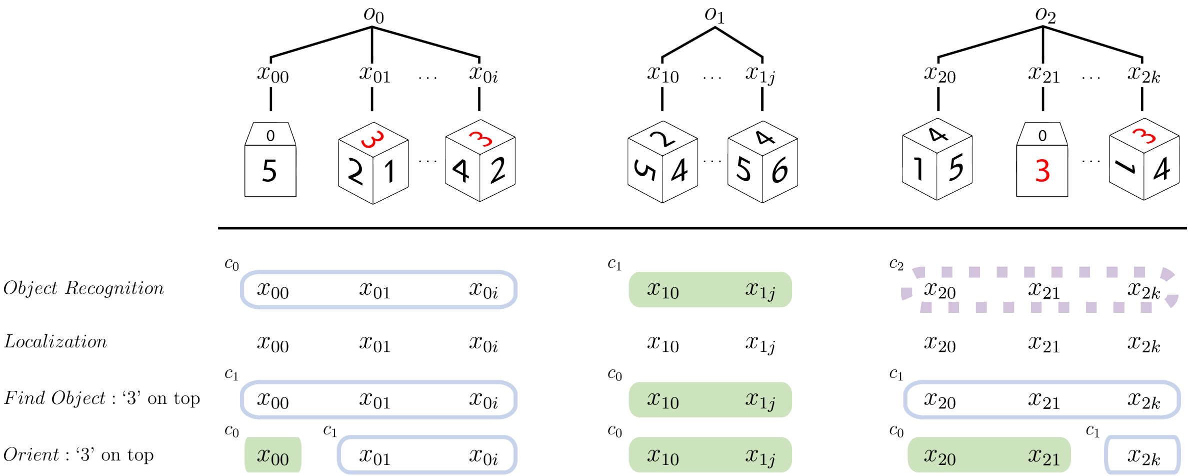

Figure 1: Simplified DBNs for three objects (top) are illustrated with task partitions for sample task types (bottom). Different subsets of the task partition are shown with different outlines.

partitioned into n subsets based on which object they belong to.

A *localization* task establishes the pose with respect to features of one or more object(s) encoded in the aspect nodes. The robot is presented with a single sensor view of either known or unknown identity. For each hypothesis, the robot has access to $|X|$ aspect nodes for all n ATG models and has to identify which known aspect node x_i, $0 \leq i < |X|$, corresponds to the constellation of features detected in this single view. The partition for *localizations* has a subset for each aspect node.

We define the *find* task as follows: the robot is presented with one or more object(s) of either known or unknown identity and has access to n known ATG models. The robot interacts with the object(s) until it is certain that at least one object satisfies the task specifications. The partition for the example in Figure 1 has two subsets, one for aspect nodes belonging to objects with a 3 on top and another for ones that do not.

The *orient* task is a *find* task with the added specification of the configuration that the object should have with respect to the robot. Only matching aspect nodes x are considered as task success (as opposed to all aspect nodes of objects with at least one matching aspect node). The partition for the example in Figure 1 forms two subsets, one for aspect nodes with a 3 on top and another for ones that do not.

Experiments and Results

In order to demonstrate the capabilities of this belief-space planning framework, we use a model set specifically designed to stress planners. Objects in this set are *ARcubes*, which are rigid cubes with a single ARtag centered on each of the six faces. ARtags can be easily identified and localized and are used as a proxy for more general purpose visual processing (Kato and Billinghurst 1999). Only their id and position are used. A large number of possible visually similar cubes and the natural sparseness of visual features leads to a large degree of ambiguity, Additionally, each visually unique cube can have up to six eccentrically weighted counterparts, which are visually identical and can only be differentiated through the transition dynamics of manual actions. The partial observable nature of the ARcubes together with the high ambiguity require long sequences of actions to recognize objects or manipulate them. More realistic objects are often easier to differentiate from even a single view. By using ARcube objects, we can generate a domain that is difficult for the planner instead of the perception.

The ATGs for the ARcube objects include transitions for FLIP, LIFT, PUSH, and ORBIT actions with the uBot-6 mobile manipulator. The uBot-6 and the actions are detailed in (Ruiken, Lanighan, and Grupen 2013; Ruiken et al. 2016b).

In the first setup, we illustrate solving two different *find tasks*. A second setup shows how the aforementioned task types can be sequenced for the robot to solve a structure copying task.

Find Tasks

As described in previous sections, assigning different partitions to a task can change the distribution over which the ABP plans, and thus, reconfigure the planner for different tasks. We define task specifications for two different *find* tasks and run the planner using their respective task partitions in simulation. We use 30 ATG models for ARcube objects. This model set contains 16 visually unique cubes. Some of these cubes have up to six eccentrically weighted counterparts. For each case, we provide rollouts of the belief over the subsets c_i of partition C. Figures 2(a) and 2(b) include the belief over objects to illustrate how the subsets of C are composed. Each o_i is textured based on the subset of C to which it belongs.

We demonstrate two examples of the *find* task to showcase two common scenarios. The first task is to find an object matching ATG model o_{24}. The robot is presented with an unknown object and needs to determine if this object is indeed object o_{24}. The rollout of the beliefs over subsets c and the object identity o can be seen in Figure 2(a). The planner chooses a sequence of actions to focus on o_{24} without having to worry about telling it apart from all the other object models.

The second task is to find an object that could be oriented such that a set of features is in the correct relative position to the robot. In this example, ARtag '1' should face the robot, '4' should be on top, and '2' should be on the bottom of the cube facing the floor. The identity of the object is not important. The subsets c defining the task can be seen in Figure 2(b) together with rollouts of the beliefs over c and o. The task succeeds without the belief condensing over the identity of the object at hand; the robot can focus on what matters for the task.

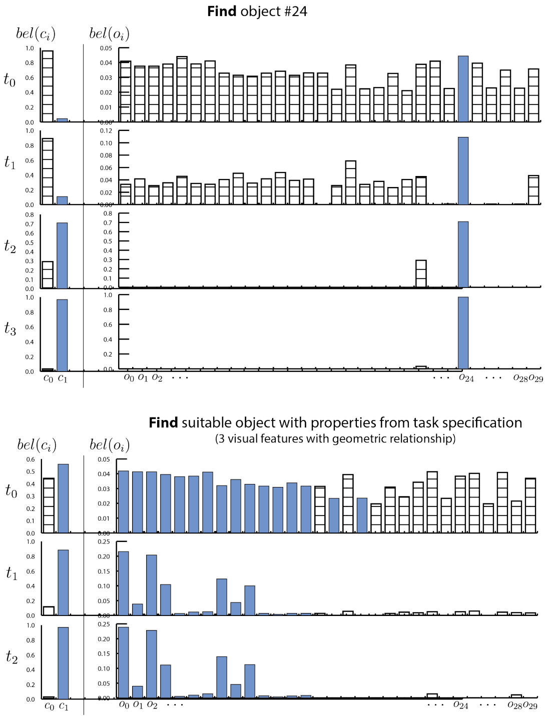

Figure 2: Top: Example task to find an object matching a specific ATG model (o_{24}). Target subset c_1 contains all aspect nodes of object o_{24}; those of all other objects are in c_0. Bottom: Example task to find an object with matching features. Target subset c_1 contains all aspect nodes of objects that contain the necessary features; those of all other objects are in c_0. Task specification: belief over subsets c is shown on the left. To visualize how subsets c are composed, the belief over objects o is shown on the right

Structure Copying

In this setup uBot-6 is presented an assembly consisting of two ARcubes. The robot is required to observe the target objects and reproduce the structure in a staging area. Both the original assembly and staging area for the copied structure are known to the robot and contain visual markers on the wall as pose guidance fiducials. For simplicity, the task specification is only based on observations from a single vantage point (one aspect). In general, the task can be based on constraints from a history of observations. For example, the robot could take observations from different vantage points and interact with the objects in the target assembly to gather more information in order to replicate it more precisely. Figure 3 (top) shows a side-by-side comparison of the target assembly and the assembly reproduced by the robot. For this experiment, the robot needs to pick-and-place two ARcubes in the designated staging area. We use our proposed algorithm to perform pick-and-place actions by sequencing task types that were presented above. The ATG model set used for this demonstration contains 14 object models.

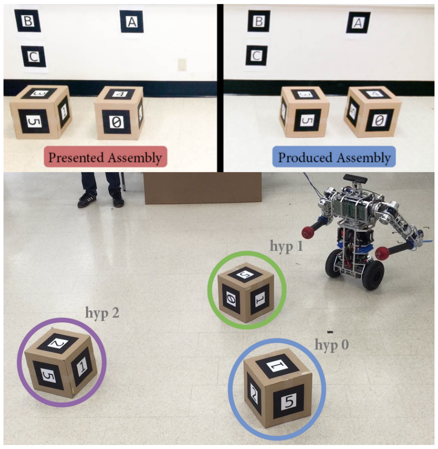

Figure 3: Side-by-side comparison of the assembly template (top left) and the assembly reproduced by the robot (top right). The robot observes the assembly template and copies it in the staging area using objects that it determines to be appropriate from the search scene (bottom).

Conclusion

We have shown a method for solving a variety of tasks in belief-space. The ATG restricts the state space and the set of actions to only useful interactions with an object. To demonstrate the flexibility of our task representation, we applied it to tasks at multiple levels in our hierarchy and showed, the planner selects actions that are relevant for the task. This allows the planner to quickly condense belief and efficiently complete tasks on the robot. Future work will extend the features to include haptic feedback.

Acknowledgements

The authors would like to thank Jay Ming Wong, Mitchell Hebert, Li Ku, and Erik Learned-Miller for their contributions. This material is based upon work supported under Grant NASA-GCT-NNX12AR16A and NSF GRFP Grant 1451512. Any opinions, findings, conclusions, or recommendations expressed in this material are solely those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration or the National Science Foundation.

References

Ballard, D. 1981. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition* 13(2):111–122.

Capi, G. 2007. Robot task switching in complex environments. In *Proc. of IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics (AIM)*.

Deinzer, F.; Derichs, C.; Niemann, H.; and Denzler, J. 2009. A framework for actively selecting viewpoints in object recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)* 23(04):765–799.

Eidenberger, R., and Scharinger, J. 2010. Active perception and scene modeling by planning with probabilistic 6D object poses. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

Grabner, M.; Grabner, H.; and Bischof, H. 2005. Fast visual object identification and categorization. In *Proc. of NIPS Workshop in Interclass Transfer*.

Hogman, V.; Bjorkman, M.; Maki, A.; and Kragic, D. 2015. A sensorimotor learning framework for object categorization. *IEEE Trans. on Autonomous Mental Development (TAMD)* 8(1).

Kato, H., and Billinghurst, M. 1999. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proc. of IEEE/ACM Int. Workshop on Augmented Reality (IWAR)*.

Lai, K.; Bo, L.; Ren, X.; and Fox, D. 2011. A scalable tree-based approach for joint object and pose recognition. In *Proc. of AAAI Conf. on Artificial Intelligence*.

Ruiken, D.; Liu, T. Q.; Takahashi, T.; and Grupen, R. 2016a. Reconfigurable tasks in belief-space planning. In *Proc. of IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*.

Ruiken, D.; Wong, J. M.; Liu, T. Q.; Hebert, M.; Takahashi, T.; Lanighan, M. W.; and Grupen, R. A. 2016b. Affordance-based active belief: Recognition using visual and manual actions. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

Ruiken, D.; Lanighan, M. W.; and Grupen, R. A. 2013. Postural modes and control for dexterous mobile manipulation: the UMass uBot concept. In *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*.

Sen, S., and Grupen, R. 2014. Integrating task level planning with stochastic control. Technical Report UM-CS-2014-005, University of Massachusetts Amherst.

Wawerla, J., and Vaughan, R. T. 2009. Robot task switching under diminishing returns. In *Proc. of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.