Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling (ICAPS 2017)

Hybrid Task Planning Grounded in Belief: Constructing Physical Copies of Simple Structures

Takeshi Takahashi,* Michael W. Lanighan,* Roderic A. Grupen*
Laboratory for Perceptual Robotics
College of Computer and Information Sciences
University of Massachusetts Amherst
Amherst, MA 01003, USA
{*ttakahas, lanighan, grupen*}@*cs.umass.edu*

Abstract

Symbolic planning methods have proved to be challenging in robotics due to partial observability and noise as well as unavoidable exceptions to rules that symbol semantics depend on. Often the symbols that a robot considers to support for planning are brittle, making them unsuited for even relatively short term use. Maturing probabilistic methods in robotics, however, are providing a sound basis for symbol grounding that supports using probabilistic distributions over symbolic entities as the basis for planning. In this paper, we describe a belief-space planner that stabilizes the semantics of feedback from the environment by actively interacting with a scene. When distributions over higher-level abstractions stabilize, powerful symbolic planning techniques can provide reliable guidance for problem solving. We present such an approach in a hybrid planning scheme that actively controls uncertainty and yields robust state estimation with bounds on uncertainty that can make effective use of powerful symbolic planning techniques. We illustrate the idea in a hybrid planner for autonomous construction tasks with a real robot system.

Introduction

Planning and executing tasks in partially observable systems is a challenging problem. One common way researchers are addressing this challenge is by using a family of hierarchical planners called task and motion planners (Gravot, Cambon, and Alami 2005; Wolfe, Marthi, and Russell 2010; Srivastava et al. 2013). Generally, these approaches divide the problem into two parts, a high-level task planner that manipulates a symbolic representation of the problem and a low-level motion planner that controls effectors to realize the symbolic plan. High-level planners often rely on symbolic representations of the problem that makes planning tractable by typically ignoring hidden state and/or uncertainty.

Some researchers investigate grounding symbolic representations in actions and experience (Pasula, Zettlemoyer, and Kaelbling 2007; Kulick et al. 2013; Konidaris, Kaelbling, and Lozano-Perez 2014). Maturing probabilistic methods in robotics are providing a sound basis for symbol grounding that supports using probabilistic distributions over symbolic entities as the basis for planning. Our approach grounds symbols describing objects in probabilistic distributions over models called Aspect Transition Graphs (Sen 2013; Ku et al. 2014; Ruiken et al. 2016b). Our system actively interacts with a scene until the distribution over higher-level abstractions stabilizes adequately at which point, powerful symbolic planning techniques can provide reliable guidance to problem solving.

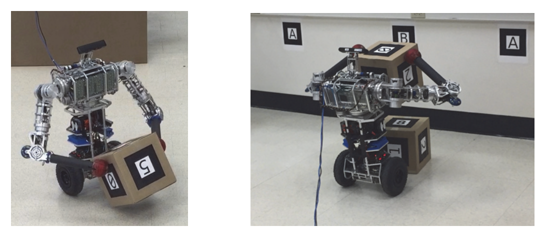

Figure 1: The left figure shows the robot identifying a goal object from a model set through interaction. Once uncertainty over objects is controlled, assemblies can be planned symbolically and executed (right figure).

We investigate a sensorimotor task that constructs a replica of a reference arrangement of objects through actively interacting with partially observed objects. Uncertainty and partial observability are addressed using a Dynamic Bayes Network to fuse information over sequences of actions and to condense belief on symbols (Figure 1 left panel). Plans are then constructed that respect preconditions and postconditions as well as resource constraints using symbolic planning techniques on the vetted symbols and executed (Figure 1 right panel).

Related Work

Several groups are investigating hybrid task planning approaches (Wolfe, Marthi, and Russell 2010; Gravot, Cambon, and Alami 2005; Srivastava et al. 2014). These approaches divide the task into a symbolic task planning problem and a realized motion planning problem. The motion planners used by these approaches provide guarantees on path planning and collision avoidance, handling uncertainty and errors at execution time. These approaches utilize an interface layer between the task level symbolic planner and the motion planner which translates symbols into actions.

*Authors are listed in random order to denote equal contributions.

Object identity is rarely considered in task and motion planning work, as is the question of where the symbols originated from.

To address the "where did the symbol come from" problem, several researchers are exploring methods to ground symbols. In order to account for noise and uncertainty in action Pasula et al. learn domain specific probabilistic models that can be used to realize symbols for planning in a simulated environment (2007). Rather than use predefined symbols, Konidaris et al. learn symbol representations from low-level features in simulation (2014). Kulick et al. demonstrate how relational symbols that are grounded through supervised learning can be acquired and utilized for planning on a robot (2013). These methods demonstrate that symbols can be grounded in robotics to facilitate planning. However, the models require some predefined structure or are domain specific.

Belief space approaches to the task planning problem have been investigated previously, notably in Kaelbling and Lozano-Pérez (2011) and in Hadfield-Menell et al. (2015). Kaelbling and Lozano-Pérez recursively refine an abstract task level plan through planning and acting in belief space. The key insight in this work is to construct finite horizon plans based on current belief and re-plan when necessary. Hadfield-Menell et al. utilizes a belief space planner based on geometric principles with a pre-defined domain and problem. In our approach, a domain for symbolic planning is derived from a model set with additional geometric constraints. Planning problems are generated at run-time.

Recognizing objects in a scene has been extensively investigated in the computer vision community. Recent results in deep learning have shown great success in this area (Krizhevsky, Sutskever, and Hinton 2012; Long, Shelhamer, and Darrell 2015). These methods use a single image to recognize objects. Some objects are ambiguous from a single viewpoint, so these approaches may not work. A robot can actively interact with the object to obtain multiple perspectives to resolve these ambiguities. Actively selecting the next viewpoint is known as active perception. Active perception has been investigated in both computer vision and robotics (Aydemir et al. 2013; Shubina and Tsotsos 2010; Denzler and Brown 2000; Ruiken et al. 2016b; Sridharan, Wyatt, and Dearden 2010). The goal of this area is to find or recognize an object. They do not consider how to conduct higher-level tasks with this information.

Hybrid Task Planner

Our hybrid task planner consists of two planning algorithms: a model-based belief space planner and a symbolic planner. The model-based belief space planner is used to overcome uncertainty in lower-level interactions with objects in the environment and to ground symbols. It is used to recognize the target and to orient the target to specific configurations relative to the robot. The symbolic planner is used at the high-level to handle resource and geometric constraints. In this paper, we focus on a task where a robot copies a demonstrated arrangement of features. We define an assembly to consist of the observable features from a pre-specified viewpoint. The copy task involves the following sub tasks: (1) find the targets for assembly, (2) orient the targets, and (3) *pick&place* the target in the arrangement.

A pseudo-code of our proposed algorithm is shown in Algorithm 1. If the symbolic planner provides a solution, the robot executes the solution. Otherwise, the robot searches for building blocks using the belief space planner until the symbolic planner finds a solution. The details are explained in the following sections.

Algorithm 1 Hybrid Task Planner

while assembly is not done **do**
  $solution$=SymbolicPlanner(PDDL $problem$)
  if $solution$ does not exist **then**
    Compute c_{target} using Eqn (3)
    Select $a(h_k) \in A_{find}$ from ABP with *find*(c_{target})
    Execute $a(h_k)$
    if $bel(c^k_{target}) > \beta_{symbol}$ in Eqn (1) **then**
      Symbolize c^k_{target} and generate PDDL $problem$
  else
    Select $a(h_k, c^k_{target}) \in A_{pick\&place}$ from $solution$
    Select $a_{orient}(h_k) \in A_{orient}$ from ABP with *orient*(c^k_{target})
    Execute $a_{orient}(h_k)$
    if $bel(c^k_{target}) > \beta_{orient}$ in Eqn (4) **then**
      Execute $a(h_k, c^k_{target})$
      Observe and generate PDDL $problem$

Object model—Aspect Transition Graph

We use the Aspect Transition Graph (ATG) (Sen 2013)(Ku et al. 2014)(Ruiken et al. 2016b) to model the objects used as the building blocks for assembly. The term "aspect" has roots in the computer vision community (originally introduced in the 70s (Koenderink and Doorn 1979)) to represent an object using multiple viewpoints. We adopted this term in our work to represent salient features observed from a specific, relative sensor geometry and use it as the latent state in our system. A subset of observable features f provides support for aspect nodes x that can be generated by objects. Actions $a \in A$ cause transitions between the aspects nodes. The transition probabilities are defined as $p(x_i|x_j, a(\theta))$, where θ are parameters of the action stored in the ATG.

Belief Space Planning

To address the first two sub tasks (finding and orienting the object), we employed the Active Belief Planner (ABP) of (Ruiken et al. 2016a). Their algorithm computes the belief over the classes of models C by summing the belief of aspect nodes $x \in C$ that satisfy a given specification. Actions are then selected to condense belief over these partitions.

Finding an object with the target aspect In our task, only a subset of an object's properties such as visual appearance and haptic response are needed to achieve the task. A robot finds c_{target}, a suitable class of object that contains the aspect necessary for the task. Following (Ruiken et al. 2016a), we define the *find*(c_{target}) task as follows,

$$\exists h_k \, [bel(c^k_{target}) > \beta_{symbol}], \quad (1)$$

where h represents an object hypothesis in the scene, k is the ID of the hypothesis, and β_{symbol} is a threshold. A "hypothesis" is a spatially constrained volume in which we maintain distributions of belief over multiple object models. c_{target} is computed as follows,

$$c_{target} = \{x_i | \exists x_j \exists o_k \, [\, p(o_k|x_i) = 1 \wedge \quad p(o_k|x_j) = 1 \wedge \quad \mathbb{1}(x_j) = 1 \,]\}, \quad (2)$$

where o is an object, x is an aspect node of the object, $p(o|x)$ is the probability that aspect node x belongs to object o, and C is a set of aspects that satisfy the task. $\mathbb{1}(x) = 1$ if $x \in C$ and 0 otherwise. We use mutual information as a metric to find the best next action $a \in A_{find}$ to reduce the uncertainty of the aspect effectively.

In our task, the robot needs to find multiple target goals $c_{target_1}, c_{target_2}, \cdots, c_{target_M}$ where M is the number of goals necessary for the task. The robot selects c_{target} to investigate by choosing the goal whose entropy is the highest among the target goals c_{target_j},

$$c_{target} = \arg\max_{c_{target_j}} E[-\log_2(p(c_{target_j}))]. \quad (3)$$

Orienting an object The robot will be required to orient the target block to specific configurations relative to the robot in order to prepare for *pick*&*place*. Following (Ruiken et al. 2016a), the *orient*(c^k_{target}) task is defined in ABP as follows to obtain $a \in A_{orient}$,

$$bel(c^k_{target}) > \beta_{orient} | c_{target} = \{x_j | \mathbb{1}(x_j) = 1\}. \quad (4)$$

Symbolic Planning and Symbol Grounding

We leverage the power of symbolic planning to resolve the preconditions of actions and resource constraints. In order for such a system to be reliable we ground the symbols by condensing belief. Symbols are derived from aspects in ATGs. By utilizing aspects as symbols, we can aggregate all the aspects (regardless of the object that they belong to) that support the task. When the belief in an aspect is sufficiently high, we can *symbolize* the aspect for use in planning.

Currently, we assume a task domain has been specified in PDDL[1]. Inspired by (Srivastava et al. 2014), we use an interface layer to translate between the symbolic and belief-based physical representations in order to generate PDDL problems at run-time (Lines 8, 15 in Algorithm 1)(See Figure 2 for an example).

The main symbols used in our system are directly derived from the ATG model set provided to the robot and the specified goal. Additional constraints (such as supporting/on relations) are determined using additional geometrical constraints. After a problem has been generated, it is submitted to a symbolic planner (Helmert 2006) that quickly finds a satisficing plan if one is available. In the problems we considered, satisficing was a more practical choice over optimality due to the inherent partial observability present in the task. If a plan does not exist the robot continues to symbolize its environment until one can be found.

[1]PDDL files are available at:
https://www-robotics.cs.umass.edu/index.php/Main/Publications

```
(define (problem auto-generated)
  (:domain  arcube-assembly )
  (:objects  floor - aspect h2 - obj
    h0 - obj h1 - obj
    1-x - aspect 3-0 - aspect )
  (:init  (clear floor) (valid_object h2)
     (valid_object h0) (valid_object h1)
     (has_aspect h0 1-x)(has_aspect h1 1-x)
     (has_aspect h2 1-x)(has_aspect h2 3-0) )
  (:goal  (and (on 1-x floor) (on 3-0 1-x)) ))
```

Figure 2: Example of an ARcube assembly problem generated during the task in Figure 3. :goal statement is generated when the robot registers the demonstrated assembly. The :init and :object statements are updated after each symbolization.

Demonstrations

To demonstrate our proposed mechanism, we conducted a simple copy task with resource constraints using the uBot-6 mobile manipulator (Ruiken, Lanighan, and Grupen 2013) with ROS (Quigley et al. 2009). We construct assemblies out of objects in the ARcube model-set of (Ruiken et al. 2016b). ARcubes are partially observable objects that can be used to precisely control planning complexity. Each ARcube has 48 aspect nodes. Twenty different ARcubes are in the model set in our demonstration, yielding $n * 20 * 48$ states for the belief space planner to consider where n is the number of building blocks present. Motion constraints are considered by the belief-space planner and solved using a harmonic function motion planner (Connolly and Grupen 1993). The ABP uses an adaptive search depth akin to (Ruiken et al. 2016a) where belief is propagated further into the future if the next action is planned in 0.6 seconds. This better informs action selection.

The robot constructs a copy of a stack that is presented *a priori*. The copy task requires the proper arrangement of two ARcubes. The robot first needs to register the goal of the task by observing a target stack. There will be two target aspects in the stack: one aspect for the bottom of the stack and one for the top. The robot parses these aspects based on its model-set to register the goal. The robot then goes to the copy area to observe the current copy state. The robot determines what aspects are missing for the copy, then generates sub-tasks to search for these aspects. There are three ARcubes to consider as building blocks for the replica. Each of the three affords aspects that can be used for the bottom block, but only one of them satisfies the top (see Figure 3). The planner must resolve resource constraints before arranging the objects.

Initially, the robot is unaware of the number or identity of building blocks available for the copy task. The number and identity are exposed through the belief space planner. Actions used by the symbolic planner are limited to pick and place actions. These symbolic actions are realized through an interface layer of actions A := {*pick*&*place, orient*}, which resolves motion constraints by orienting the objects if needed. Actions used for *orient* and *find* tasks are $A_{orient} = A_{find}$:= {*orbit, flip, lift, push*}. *pick*&*place* action $a \in A_{pick\&place}$ takes two parameters: which hypothesis to pick

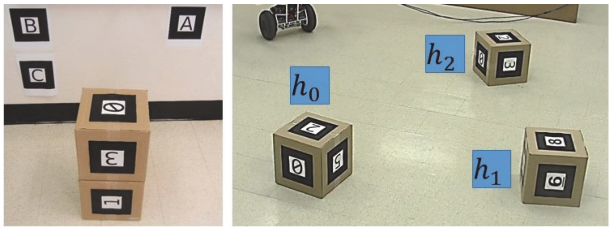

Figure 3: The left figure shows the goal of the copy task. The building blocks that can be used to copy the structure are displayed on the right. Although not visible from this view angle, hypothesis h_2 has both aspects '1-x' (an aspect with '1' in front) and '3-0' (an aspect with '3' in front and '0' on top), while hypotheses h_0 and h_1 only have aspect '1-x'. This introduces a resource constraint as only hypothesis h_2 can be used for the top position.

Figure 4: On the left is the demonstrated stack, while the right stack is the approximate copy constructed by the robot.

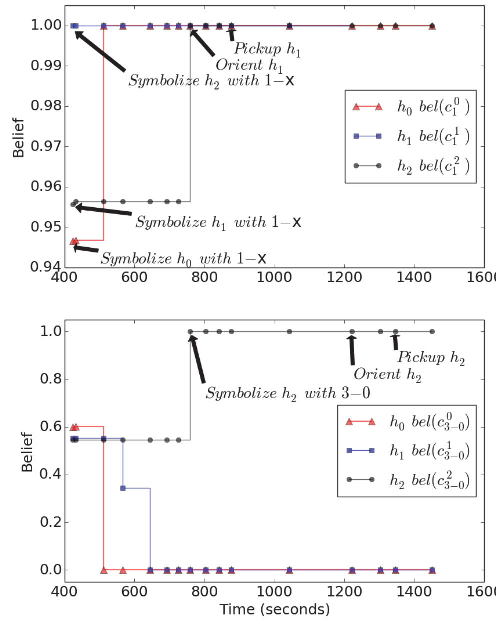

Figure 5: The figures show the task beliefs for each hypothesis in the scene for finding aspect '1-x' (top figure) and aspect '3-0' (bottom figure). The robot *symbolized* all of the hypothesis with aspect '1-x' after the first observation at 423 seconds, and symbolized hypothesis h_2 with aspect '3-0' at 759 seconds.

and the place goal. We use $\beta_{symbol} = 0.9$ and $\beta_{orient} = 0.7$ in this demonstration.

Results and Discussion

The robot successfully constructed the copy despite the resource constraint situation as shown in Figure 4. Belief over the aspects used in the plan is shown in Figure 5. The figure shows when hypotheses were *symbolized*, when targets were oriented, and when *pick&place* actions were started. Executing physical actions dominates planning time. Physical actions took 38.8 seconds on average. On average, ABP with 1-ply took 0.26 seconds, and 3.7 seconds for 2-ply. The symbolic planner took 0.23 seconds on average. The efficient computation is partially due to the simple domain and problem we are addressing at the symbolic level.

In our demonstration the robot correctly symbolized the objects needed to complete the task as seen in Figure 5 while not falsely symbolizing improper objects even though symbolizing objects in partially observable problems (such as those with ARcubes) is difficult. Symbols should be generated only when the semantics of actions are vetted by the belief space planner. To illustrate, the result would have been different if we had used a maximum likelihood observation to symbolize after the first action at 423 seconds (Figure 5). The plan formed from this belief would have led to a sub-optimal solution only containing one block—failing the copy as it would not be able to resolve the resource constraint.

We performed ten experiments changing the number of building blocks used in the copy task from two to four for the two object stack problem. In these preliminary experiments, eight successfully symbolized and planned a solution for the copy with the current implementation. The two failures were due to stochastic action outcomes. In a fully implemented closed loop belief-space approach a robot would never fail, continuing to take actions to increase belief unless some insurmountable error occurred. The only effect such failures would have is to increase the length of execution.

Conclusion and Future Work

We proposed a hybrid planning architecture that utilizes a model-based belief space planner and a symbolic planner. With this approach, we demonstrated how a belief space planner stabilizes the semantics of feedback from the environment through interaction to enable reliable *symbolization* of higher level abstractions. We detailed a preliminary result of our architecture on a real robot system performing a two-block stacking copy with resource constraints. Thanks to the *symbolization* grounded in belief, the system was able to overcome the resource constraint in this example that would have prevented maximum likelihood approaches from succeeding. Future work will focus on implementing additional actions and planning structures to enable the system to recover from failures in this domain.

Acknowledgments

The authors would like to thank Tiffany Liu and Dirk Ruiken for informative and insightful discussions. This material is based upon work supported under Grant NASA-GCT-NNX12AR16A. 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.

References

Aydemir, A.; Pronobis, A.; Göbelbecker, M.; and Jensfelt, P. 2013. Active Visual Object Search in Unknown Environments using Uncertain Semantics. *IEEE Transactions on Robotics* 29(4):986–1002.

Connolly, C. I., and Grupen, R. A. 1993. The Applications of Harmonic Functions to Robotics. *Journal of Robotic Systems* 10(7):931–946.

Denzler, J., and Brown, C. 2000. Optimal Selection of Camera Parameters for State Estimation of Static Systems: An Information Theoretic Approach. *University of Rochester Tech. Rep* 732.

Gravot, F.; Cambon, S.; and Alami, R. 2005. aSyMov: A Planner that Deals with Intricate Symbolic and Geometric Problems. In *Robotics Research. The Eleventh International Symposium*, 100–110. Springer.

Hadfield-Menell, D.; Groshev, E.; Chitnis, R.; and Abbeel, P. 2015. Modular Task and Motion Planning in Belief Space. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4991–4998. IEEE.

Helmert, M. 2006. The Fast Downward Planning System. *Journal of Artificial Intelligence Research (JAIR)* 26:191–246.

Kaelbling, L. P., and Lozano-Pérez, T. 2011. Hierarchical Task and Motion Planning in the Now. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, 1470–1477. IEEE.

Koenderink, J. J., and Doorn, A. v. 1979. The Internal Representation of Solid Shape with Respect to Vision. *Biological cybernetics* 32(4):211–216.

Konidaris, G.; Kaelbling, L. P.; and Lozano-Perez, T. 2014. Constructing Symbolic Representations for High-Level Planning. In *28th AAAI Conference on Artificial Intelligence*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in neural information processing systems*, 1097–1105.

Ku, L. Y.; Sen, S.; Learned-Miller, E. G.; and Grupen, R. A. 2014. Action-based Models for Belief-Space Planning. In *Workshop on Information-Based Grasp and Manipulation Planning, at Robotics: Science and Systems*.

Kulick, J.; Toussaint, M.; Lang, T.; and Lopes, M. 2013. Active Learning for Teaching a Robot Grounded Relational Symbols. In *IJCAI*.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully Convolutional Networks for Semantic Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440. IEEE.

Pasula, H. M.; Zettlemoyer, L. S.; and Kaelbling, L. P. 2007. Learning Symbolic Models of Stochastic Domains. *Journal of Artificial Intelligence Research (JAIR)* 29:309–352.

Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; and Ng, A. Y. 2009. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, volume 3.

Ruiken, D.; Liu, T. Q.; Takahashi, T.; and Grupen, R. A. 2016a. Reconfigurable Tasks in Belief-Space Planning. In *16th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*. IEEE.

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 *IEEE/RSJ the International Conference on Intelligent Robots and Systems (IROS)*. IEEE.

Ruiken, D.; Lanighan, M. W.; and Grupen, R. A. 2013. Postural Modes and Control for Dexterous Mobile Manipulation: the UMass uBot Concept. In *13th IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 721–726. IEEE.

Sen, S. 2013. *Bridging the Gap Between Autonomous Skill Learning and Task-Specific Planning*. Ph.D. Dissertation, University of Massachusetts Amherst.

Shubina, K., and Tsotsos, J. K. 2010. Visual Search for an Object in a 3D Environment using a Mobile Robot. *Computer Vision and Image Understanding* 114(5):535–547.

Sridharan, M.; Wyatt, J.; and Dearden, R. 2010. Planning to See: A Hierarchical Approach to Planning Visual Actions on a Robot using POMDPs. *Artificial Intelligence* 174(11):704–725.

Srivastava, S.; Riano, L.; Russell, S.; and Abbeel, P. 2013. Using Classical Planners for Tasks with Continuous Operators in Robotics. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 27–35.

Srivastava, S.; Fang, E.; Riano, L.; Chitnis, R.; Russell, S.; and Abbeel, P. 2014. Combined Task and Motion Planning Through an Extensible Planner-Independent Interface Layer. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 639–646. IEEE.

Wolfe, J.; Marthi, B.; and Russell, S. J. 2010. Combined Task and Motion Planning for Mobile Manipulation. In *International Conference on Automated Planning and Scheduling (ICAPS)*, 254–258.