Reconfigurable Tasks in Belief-Space Planning

Dirk Ruiken, Tiffany Q. Liu, Takeshi Takahashi, and Roderic A. Grupen

Abstract— **We propose a task representation for use in a belief-space planning framework. The representation is based on specialized object models that enable estimation of an abstract state of a robot with respect to an object. Each manipulation task is represented using a partition over these states defined by the set of known object models. Solutions to such tasks are constructed in a belief-space planner using visual and/or manual interactions with objects that condense belief in a target subset of the task partition. This partition integrates belief over states into a task belief without altering the original belief representation. As a result, sequences of tasks can be addressed that inherit the complete estimate of state over the entire history of observations. Demonstrations of the technique are presented in simulation and on a real robot. Results show that using this task representation and the belief-space planner, the robot is able to recognize objects, find target objects, and manipulate a set of objects to obtain a desired state.**

I. INTRODUCTION

Robotic planners have to deal with uncertainty and partial observability. Belief-space planners are often employed to address these issues but can make it difficult to express generic tasks. A uniform framework for addressing a full range of manipulation tasks with these powerful techniques remains a challenging problem.

This paper describes a task representation for belief-space planning. The approach uses a planning framework called the Active Belief Planner (ABP), which is based on object models that enable estimation of an abstract state of a robot with respect to an object or the environment. This planning framework was used in previous work to perform object recognition based on belief over the abstract state using information theoretic measures to select the most informative visual and manual interactions [1]. In this paper, we generalize the planner to solve any task that can be supported by the known object models. Tasks are defined as goal subsets of a state partition. The planner then tries to enhance the certainty that state estimates reside within these subsets. By defining such a task partition, belief over individual states can be aggregated into a task belief with no changes to the underlying belief state. This supports a more general task planner that preserves the state estimate derived from the total history of actions and observations over multiple tasks and continuous interaction.

II. RELATED WORK

This paper uses a generalization of seminal contributions from the active vision community [2, 3] that can be applied to multi-modal perceptual information and to general-purpose problem solving in an active belief planning framework [1].

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 [4]. Loeb and Fishel discuss how Bayesian Exploration can be used to construct queries to associative memory structures of previous sensorimotor experiences [5]. Browatzki *et al.* use a similar action selection metric and transition probabilities on a view sphere with a set of actions that execute in-hand rotations [6]. 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 [7]. Ruiken *et al.* extend the ATGs by adding geometric information and cost estimates to improve forward modeling capabilities [1].

A popular approach for handling partial observability and uncertainty in robotics is the use of partially observable Markov decision processes (POMDPs). For example, Hsiao *et al.* use a decision theoretic solution to a POMDP to determine relative pose of a known object [8]. Optimal solutions to POMDPs are provided by offline solvers that compute an optimal policy but are generally intractable for real robot problems. Online planners for POMDPs address this problem by planning up to a finite horizon and then choosing the best action at that plan depth [9, 10]. The size of the state space required can still be prohibitive, however. To scale these approaches, Castanon uses a hierarchical POMDP to recognize many objects in a scene [11]. Sridharan *et al.* introduce a hierarchical POMDP to plan visual operators to recognize multiple objects in the scene [12]. Araya *et al.* noted that the reward structure of POMDPs can be prohibitive when the distribution of belief itself is critical for the task [13]. In previous work, we used a belief-space MDP with online planning to plan over the belief distribution itself to perform recognition tasks [1, 7, 14]. These planners can work with large model sets, however, they did not handle planning over multiple objects at the same time. This work uses the same planning framework, but employs a hierarchical structure to overcome this deficiency.

Often the robotics community works on when to switch between tasks [15, 16] rather than how to solve different active perception tasks using a single planner. Grabner *et al.* propose a single framework to solve both object identification and object categorization in object recognition

problems [17]. Lai *et al.* propose a scalable tree-based approach to solve category recognition, instance recognition, and pose estimation [18]. These methods, however, are not active recognition algorithms, and therefore, they do not interact with the environment to reduce the uncertainty. We combine active perception with the ability to switch between tasks.

III. TECHNICAL APPROACH

Our planning framework extends a previous version that uses a belief-space planner and a population of forward models to track the belief over the state of the interaction between the robot and the world. We propose a hierarchical planning structure to overcome complexity of environments with multiple objects. A task interpreter is introduced to generalize task definitions in belief-space. The following sections provide details on the forward models from [1], the hierarchical organization, the task interpreter, examples of task types and their partitions, and the resulting belief-space planner.

A. Aspect Transition Models

To model the state with respect to objects in the environment, we use aspect transition graphs (ATGs). These models are centered around the concept of aspects. In general, only a subset of the features attributed to an object can be detected from any given sensor geometry. These subsets of features define the aspect of the object. Aspects are used to specify nodes, called *aspect nodes*, in a multi-graph where edges represent actions that cause probabilistic transitions between the aspect nodes. Actions are implemented as controllers with parameters and estimates of the cost of the action. Multiple aspect nodes in an ATG can share identical aspects that can only be differentiated by the outcome of informative actions. Additionally, geometric information in the models can be used to predict sensor geometries for new observations and support pose estimation. ATG models can be hand-built or autonomously learned by the robot [14, 19–21]. The models used in this work are hybrids, wherein nodes and edges associated with visual actions were learned and those for manual actions were hand-built.

A Dynamic Bayes Net (DBN) is used as a recursive, hierarchical inference engine in which objects o generate aspect nodes x that then generate observations z that can be viewed from a single sensor geometry. The DBN fuses the history of observations and actions a into a maximum likelihood distribution over aspect nodes. The belief $bel(x)$ over the aspect nodes of all known ATG models is used as state for the belief-space MDP. The ATG provides forward models $p(x_{t+1}|x_t, a_t)$ and information for observation models $p(z_t|x_t)$ that are used for the belief update.

B. Hierarchical Planning

In general, tasks can involve multiple objects. Planning over all objects at once can become computationally expensive due to the combinatorial nature of the decision space [11]. Therefore, we cluster features 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 complexity of the planning algorithm is $O(|K||A||X|^2)$, 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 (Alg. 1). The number of hypotheses is expected to be roughly the number of objects in the scene.

C. Belief-Space Planning with Task Interpreter

Based on observations made in the environment, the robot tracks the distributions of belief over each of the k hypotheses. Using segmentation techniques, observations are evaluated to match to existing hypotheses.

The distribution over objects, and thus ATGs, is used to propagate belief forward over multiple actions. For each of the k hypotheses, given a belief 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 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. Our framework uses a Hough transform-based approach described in [1]. Incorporating new observations yields the posterior belief

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

where η is a normalizer.

Task Partitions: In previous work, we used the entropy of posterior belief distributions over objects to select optimal actions for object recognition [1]. The technique presented in Section III-D generalizes the ABP to any task that can be expressed as a partition over the set of states (aspect nodes) using a task interpreter.

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:

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

We use notation $c(x)$ to denote the specific subset of C an aspect node x belongs to. This mapping from an aspect node x to the corresponding subset c is done in constant time and allows the whole belief aggregation over the subsets of C to be calculated in linear time. Example task types are found in Sections III-D.1–III-D.4.

Information Gain: 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,

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

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.

Extending the ABP for Reconfigurable Tasks: To evaluate actions, the belief over aspect nodes is rolled out based on the forward model provided by the ATGs. 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 simplicity, the resulting algorithm is shown for a 1-ply search in Algorithm 1.

For each object hypothesis h_k and available action a_t, the algorithm performs a control update to calculate the expected belief $\overline{bel}(x_{t+1}^k)$ after taking action a_t (Line 8). Transition probabilities for the process update $p(x_{t+1}|x_t, a_t)$ are stored in the edges of the ATG. We use threshold $\alpha \max_{x_{t+1}^k}(\overline{bel}(x_{t+1}^k))$ (relative to the highest current belief) to exclude beliefs that come from expected aspect nodes with low probability (Line 10). The α term is a single value from range $(0, 1]$ set by the user at the beginning (Line 1). For all experiments, we used $\alpha = 0.1$. The aspect geometry inside the ATG provides an expected observation z_{t+1} for each expected future aspect node x_{t+1} (Line 11). After performing an observation update following Equation 1 (Line 13), the belief over the corresponding subset of the task partition $c_{t+1}^k(x)$ is updated (Line 14). The expected information gain IG is calculated for each object hypothesis and action combination (Lines 15–19) with

---

Algorithm 1 Active Belief Planner (shown for 1-ply)

1: $\alpha = $ Future observation update threshold
2: $\tau_{h_k, a_t} = 0$ for all h_k, a_t
3: $\boldsymbol{IG} = \{\}$
4: **for all** h_k **do**
5:   **for all** a_t available in ATG **do**
6:    $bel(c_{t+1}^k) = 0$ for all c_t^k
7:    **for all** x_{t+1}^k **do**
8:     $\overline{bel}(x_{t+1}^k) = \sum_{x_t^k} p(x_{t+1}^k|x_t^k, a_t) bel(x_t^k)$
9:    **for all** x_{t+1}^k **do**
10:    **if** $\overline{bel}(x_{t+1}^k) > \alpha \max_{x_{t+1}^k}(\overline{bel}(x_{t+1}^k))$ **then**
11:     $z_{t+1}^k \leftarrow ATG(x_{t+1}^k)$
12:     **for all** x_{t+1}^k **do**
13:      $bel(x_{t+1}^k) = \eta\, p(z_{t+1}^k|x_{t+1}^k)\, \overline{bel}(x_{t+1}^k)$
14:      $bel(c_{t+1}^k(x)) \mathrel{+}= bel(x_{t+1}^k)$
15:     $m = M(c_{t+1}^k, T(c))$
16:    **else**
17:     $m = M(c_t^k, T(c))$
18:    $\tau_{h_k, a_t} = \tau_{h_k, a_t} + \overline{bel}(x_{t+1}^k)m$
19:   $IG_{h_k}(c_t^k, a_t) = M(c_t^k, T(c)) - \tau_{h_k, a_t}$
20:   $\boldsymbol{IG} = \boldsymbol{IG} \cup IG_{h_k}(c_t^k, a_t)$
21: **while** $\arg\max_{h_k, a_t} \boldsymbol{IG}$ is not feasible **do**
22: $h_k^*, a_t^* = \arg\max_{h_k, a_t} \boldsymbol{IG}$
23: $\boldsymbol{IG} = \boldsymbol{IG} \setminus IG_{h_k^*}(c_t^k, a_t^*)$
24: **return** $\arg\max_{h_k, a_t} \boldsymbol{IG}$

$M(c_t, T(c))$ denoting the place holder for the information-based metric employed (e.g. entropy or KL divergence). The action with the highest expected information gain is chosen. Further details on the planner and the pruning methods used can be found in [1] but are not necessary for understanding the system.

D. Task Types

The proposed approach can represent a large number of tasks and task types. In this section, we define four basic task types commonly found in robotics problems. 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. Demonstrations of these tasks are shown in Section IV-A.

1) Recognition Task: 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 goal is to condense belief into a single subset of the task partition defined by objects in the model space. In other words, if all objects in the scene should be identified, the belief for each hypothesis h_k must condense on one object
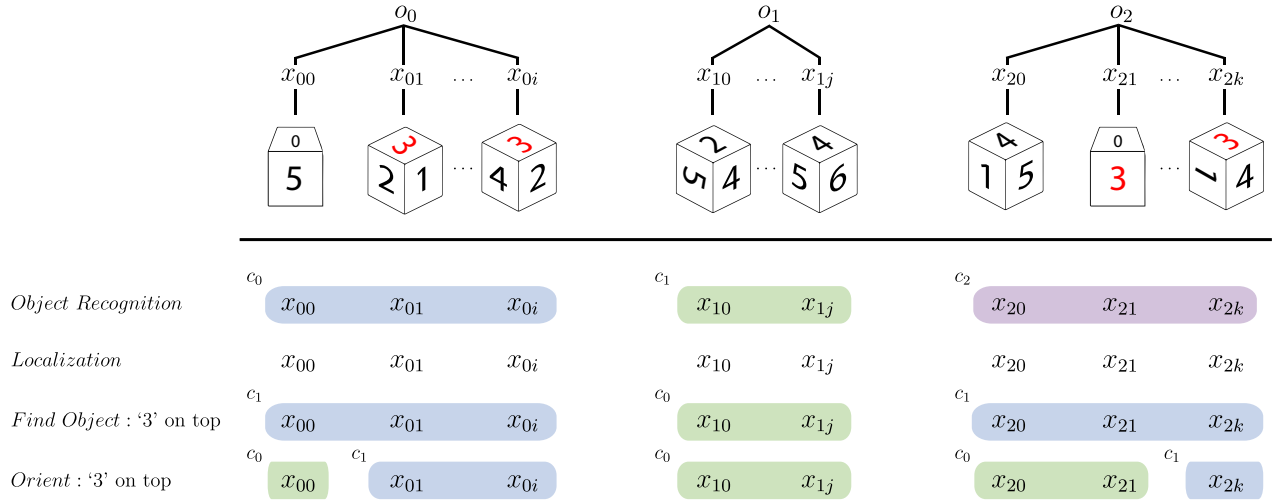
Fig. 1. A simplified DBN for three objects is shown here. Task partitions for examples of each task type are shown below. For the *recognition* task, all aspect nodes for each object are grouped into one subset of the partition, resulting in three subsets in the partition (colored blue, green, and purple). For the *localization* task, the task partition contains single-element subsets, where the element is one aspect node from one of the three objects, resulting in $(i + j + k + 3)$ subsets (uncolored for readability). For the *find* task, we show an example of finding an object with a '3' feature on top. The task partition for this contains two subsets: one has all the aspect nodes of two objects where at least one aspect node has the '3' feature (colored blue) and the second has all the remaining aspect nodes (colored green). Finally, for the *orient* task, we show an example for orienting an object such that the '3' feature is on top. This task partition also contains two subsets. One has all the aspect nodes where the '3' feature is on top (colored blue) and all remaining aspect nodes belong to the second (colored green).

identity. This task type can be expressed mathematically as

$$\forall h_k \; [\max_j bel(c_j^k(x)) > \beta], \quad (2)$$

where β is some threshold for the belief. The task partition C over all aspect nodes from all ATGs splits the aspect nodes of each ATG model into a separate subset, resulting in a partition with n different subsets c_j:

$$c_j = \{x_i | p(o_j | x_i) = 1\} \quad \text{for} \quad 0 \le j < n.$$

The row in Figure 1 labeled 'Object Recognition' illustrates this partition.

2) Localization Task: A *localization* task establishes the pose with respect to features of one or more object(s) encoded in 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, $0 \le i < |X|$, corresponds to the constellation of features detected in this single view. Again, the robot can use any action available in the ATG models to investigate and manipulate the object(s). This task type has the same mathematical formulation as *recognition* (Eq. 2) with a different task partition. The task partition C for *localization* divides each aspect node for all ATG models into separate subsets, resulting in a partition with $|X|$ different subsets c_j:

$$c_j = \{x_i\} \quad \text{for} \quad 0 \le j < |X|.$$

Once belief is condensed on an aspect node, the robot knows which object it is sensing and where it is relative to that object. An example of a resulting partition can be found in the row labeled "Localization" of Figure 1.

3) Find Task: Often the specific identity of object(s) or aspect(s) is not important. Instead, the utility of an object for a task can be based on a subset of its properties such as visual appearance, haptic responses, or interaction possibilities. Thus, a robot can be asked to *find* a suitable object—one that contains at least one aspect node that satisfies the task specifications.

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. This can be expressed mathematically as

$$\exists h_k \; [bel(c_1^k) > \beta]. \quad (3)$$

The task partition C splits all aspect nodes into two subsets— suitable (c_1) and not suitable (c_0):

$$c_1 = \{x_i | \exists x_j \exists o_k \; [\; p(o_k | x_i) = 1 \land \quad (4)$$
$$p(o_k | x_j) = 1 \land$$
$$y(x_j) = 1 \;]\},$$

$$c_0 = X \setminus c_1 \quad (5)$$

with

$$y(x) = \begin{cases} 1 & \text{aspect node } x \text{ satisfies task} \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

Both reduction of entropy or KL divergence over $bel(c(x))$ work as metrics to guide the planner. Given one unknown object, the planner determines the suitability of the object for this task. If presented with more unknown objects, it will investigate the most promising object(s) first in order to find a suitable one.

4) Orient Task: The *orient* task is a *find* task with the added specification of the configuration that the object should have with respect to the robot. It uses the same mathematical formulation as in Equation 3. The same function $y(x)$ from the previous task (Eq. 6) is also used, but 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 as in Equation 4):

$$c_1 = \{x_i | y(x_i) = 1\}, \tag{7}$$

$$c_0 = X \setminus c_1. \tag{8}$$

By selecting actions that condense belief in c_1 using KL divergence as the metric, the robot can manipulate objects into a desired configuration to satisfy task requirements without having to know the precise identity of the object.

IV. DEMONSTRATIONS

In order to demonstrate the capabilities of this belief-space planning framework, we use two different setups involving the uBot-6 mobile manipulator [22]. The first setup demonstrates solving two of the task types described in Section III-D: *recognition* and *find*. The second setup shows how the aforementioned task types can be sequenced for the robot to solve a copying task.

The model set used for these demonstrations consists of ATG models for ARcubes together with FLIP, LIFT, PUSH, and ORBIT actions detailed in [1]. ARcubes are rigid cubes whose size can be adjusted to meet the requirements of the robot geometry. A single ARtag is centered on each of the six faces of the cube. An open-source ARToolKit software is available for detecting and localizing the tags as a proxy for more general purpose visual processing [23]. Visual observations of these features detect the location of the center of each tagged face. When viewing an ARcube square-on, we can refer to the locations of the tags with 'top', 'front', 'right', 'back', 'left', and 'bottom'. ARcubes are only partially observable from any single sensor geometry. The partial observability and the natural sparseness of features on any one cube lead to a large degree of ambiguity.

A. Exhibition of Planning for Three Example 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 examples of a *recognize* task and 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, which are visually identical and can only be differentiated through the transition dynamics of manual actions. For each case, we provide rollouts of the belief over the subsets c_i of partition C. Figures include the belief over objects to illustrate how the subsets of C are composed. Each o_i is colored based on subset of C to which it belongs.

1) Recognition Task – "Identify an object": For the recognition task, we present the robot with an unknown object. After the initial observation, the robot uses the ABP to select next best actions to execute until the object has been identified. In Figure 2, the belief over the object identity can be seen for several time steps until the object is correctly identified as o_{24}. The aspect nodes of each object form a separate subset c_i, and therefore the belief over c is equal to the belief over the corresponding objects o_i. It took five actions for the belief to condense completely on one object.
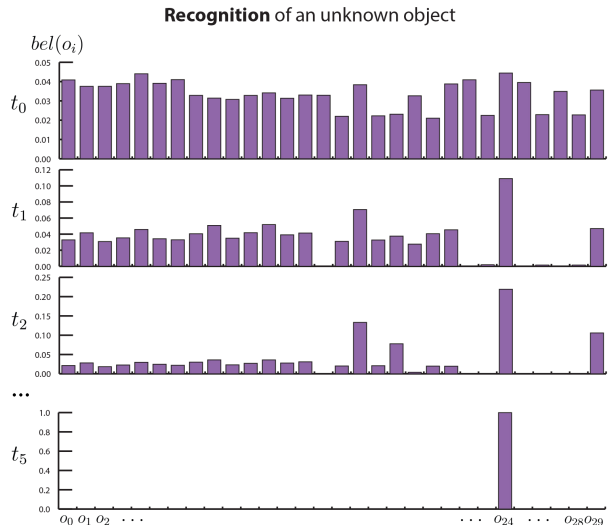

Fig. 2. Example task to recognize an unknown object: a condensed rollout of the belief over object models o_i is shown. The belief over c is equal to the belief over the corresponding objects o_i. Therefore, we refrain from coloring o_i based on their membership in c for readability.

2) Find Task – "Find a suitable object": 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 3. The color indicates the subset c to which the aspect nodes of each object belongs. Here, the aspect nodes of o_{24} belong to c_1 (blue), while all other aspect nodes belong to c_0 (green). The robot is presented with the same object as for the recognition example in Figure 2. The planner chooses a different sequence of actions since it can focus on o_{24} without having to worry about telling it apart from all the other object models, resulting in fewer actions to reach task completion.

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 4 together with rollouts of the beliefs over c and o.
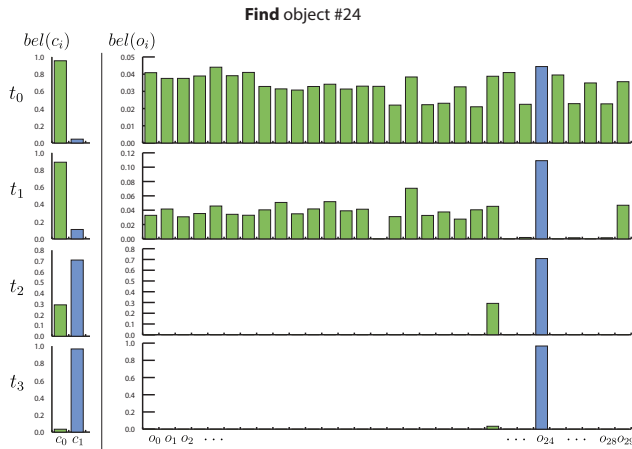
Fig. 3. Example task to find an object matching a specific ATG model (o_{24}) provided as the task specification: a rollout of belief over subsets c is on the left. To visualize how subsets c are composed, the belief over objects o is shown on the right. Target subset c_1 contains all aspect nodes of object o_{24} (blue); those of all other objects are in c_0 (green).
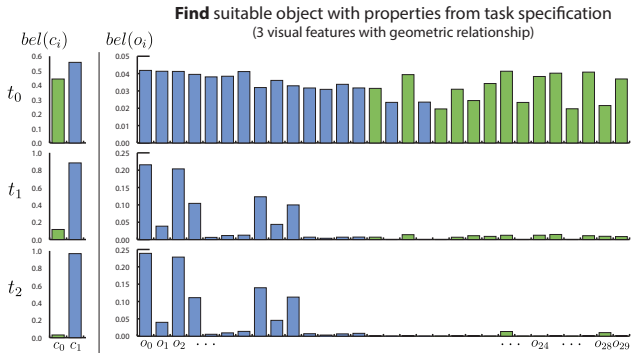

Fig. 4. Example task to find an object with matching features: a rollout of belief over subsets c is on the left. To visualize how subsets c are composed, the belief over objects o is shown on the right. Target subset c_1 contains all aspect nodes of objects that contain the necessary features and can be oriented to expose them (blue); those of all other objects are in c_0 (green).

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.

B. Sequencing Find and Orient Tasks for 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 5 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
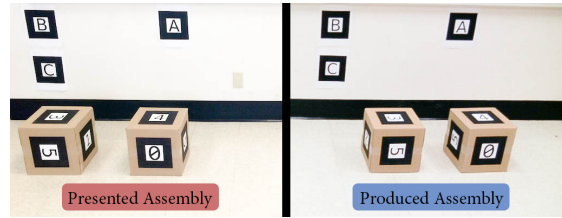

Fig. 5. Side-by-side comparison of the assembly template (left) and the assembly reproduced by the robot (right). The robot observes the assembly template and copies it in the staging area using objects that it determines to be appropriate from the search scene (Fig. 6).

algorithm to perform pick-and-place actions by sequencing task types that were presented in Section III-D. The ATG model set used for this demonstration contains 14 object models. The robot randomly chooses the first object to obtain
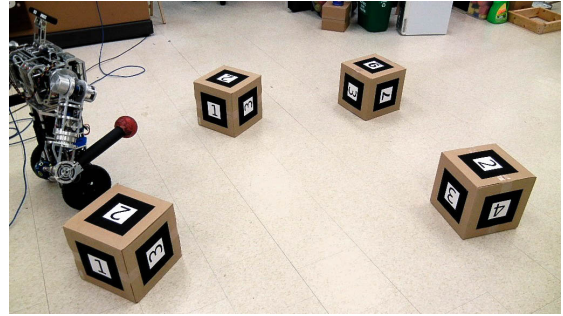

Fig. 6. Four ARcubes are placed in the search scene. The robot uses a model set of 14 ARcubes. It establishes hypotheses for each of the four ARcubes and plans over them according to the task partitions defined. 11 object models afford the '4-0' aspect and 11 afford the '5-3' aspect.

for pick-and-place. For the situation presented in Figure 5, the robot chooses to pick-and-place the right most object first. To do this, a hand-built finite state machine runs a *find* task to locate an ARcube, from the search scene of four ARcubes (Fig. 6), that affords an aspect with ARtag '0' in front and ARtag '4' on top ('0-4' aspect). Based on the observation of the assembly template, the robot assigns the partition C following Equations 4–6: $y(x_i) = 1$ if x_i is a '0-4' aspect. Once the robot is certain that it has an ARcube with those feature specifications (which takes a single action), it executes an *orient* task to manipulate the cube from the *find* task such that ARtag '4' is on top and ARtag '0' is in front (six actions). The robot assigns the partition C for this *orient* task using Equations 6–8, where $y(x_j) = 1$ if x_j is a '0-4' aspect. After the robot accomplishes the *orient* task, it uses a pick-and-place controller to grasp, transport, and drop off the cube at the designated location in the staging area.

After placing the first object, the robot goes through the same sequence of tasks for the second object. For the situation presented in Figure 5, the robot executes a *find* task for an ARcube that affords a '5-3' aspect (one action), an *orient* task to reveal the '5-3' aspect (two actions), and a pick-and-place to drop off the cube in the staging area. Figure 7 shows the belief over time in the subsets that contain the '4-0' aspect (left) and '5-3' aspect (right). For this demonstration, the execution time heavily dominated the
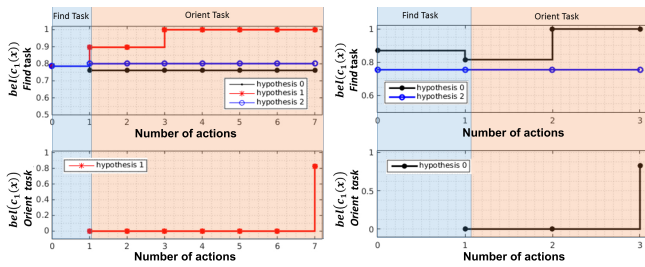
Fig. 7. These plots show the evolution of belief over the goal subset of the task partition for the objects that were copied (*left*: first object ('4-0'); *right*: second object ('5-3')). The top and bottom plots show the task beliefs for each hypothesis in the scene for the *find* and *orient* tasks, respectively. For the first object, the robot did not register one of the objects in the scene, so it only establishes three hypotheses. The belief for the *orient* task exceeds the threshold after action #7. For the second object, since the robot has already interrogated the scene, it starts with a higher belief prior for the remaining hypotheses. The belief for the *orient* task exceeds the threshold after action #3.

planning time, which was less than $1\,sec$ for each action on average. Following the sequence of tasks as presented, the robot was able to successfully reproduce the target assembly as shown in Figure 5.

V. CONCLUSION

We proposed and demonstrated a novel task representation for performing configurable information-gathering tasks with a single belief-space planner. The planner handles multiple objects in a hierarchical manner, allowing it to probabilistically reason over one object at a time. Any task that can be modeled by the underlying representation of belief dynamics (ATGs) can be expressed in the form of partitions over belief states. This enables the robot to switch between tasks while preserving state information. The choice in the type of information metric driving the planner towards task completion changes the behavior of the robot.

Currently, ATGs only contain aspect nodes for single objects, therefore any task partition over aspect nodes can only express tasks with respect to a single object. In order to handle tasks involving several objects, e.g. assemblies, they have to be decomposed into a sequence of single object tasks. This can be accomplished with a hand-built finite state machine (as we used in the presented structure copying task) or a higher-level planner. In the future, we would like to investigate how this framework can be used to solve multi-object tasks directly. We also plan to extend the task representation to combine multiple task specifications that can simultaneously influence the ABP. This can enable a robot to, for example, find several different parts simultaneously.

REFERENCES

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

[2] R. Bajcsy, "Active perception," *IEEE Trans. on Robotics and Automation*, vol. 76, no. 8, 1988.

[3] Y. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. Journal of Computer Vision*, vol. 1, no. 4, 1988.

[4] V. Hogman, M. Bjorkman, A. Maki, and D. Kragic, "A sensorimotor learning framework for object categorization," *IEEE Trans. on Autonomous Mental Development*, vol. 8, no. 1, 2015.

[5] G. E. Loeb and J. A. Fishel, "Bayesian action & perception: Representing the world in the brain." *Frontiers in Neuroscience*, vol. 8, 2013.

[6] B. Browatzki, V. Tikhanoff, G. Metta, H. H. Bülthoff, and C. Wallraven, "Active object recognition on a humanoid robot," in *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012.

[7] S. Sen and R. Grupen, "Integrating task level planning with stochastic control," University of Massachusetts Amherst, Tech. Rep. UM-CS-2014-005, 2014.

[8] K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Task-driven tactile exploration," in *Proc. of Robotics: Science and Systems (RSS)*, 2010.

[9] S. Ross, J. Pineau, S. Paquet, and B. Chaib-Draa, "Online planning algorithms for POMDPs," *Journal of Artificial Intelligence Research*, vol. 32, 2008.

[10] A. Somani, N. Ye, D. Hsu, and W. S. Lee, "DESPOT: Online POMDP planning with regularization," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2013.

[11] D. A. Castanon, "Approximate dynamic programming for sensor management," in *Proc. of IEEE Conf. on Decision and Control (CDC)*, 1997.

[12] M. Sridharan, J. Wyatt, and R. Dearden, "Planning to see: A hierarchical approach to planning visual actions on a robot using POMDPs," *Artificial Intelligence*, vol. 174, no. 11, 2010.

[13] M. Araya, O. Buffet, V. Thomas, and F. Charpillet, "A POMDP extension with belief-dependent rewards," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2010.

[14] L. Ku, S. Sen, E. Learned-Miller, and R. Grupen, "Action-based models for belief-space planning," in *RSS Workshop on Information-Based Grasp and Manipulation Planning*, 2014.

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

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

[17] M. Grabner, H. Grabner, and H. Bischof, "Fast visual object identification and categorization," in *Proc. of NIPS Workshop in Interclass Transfer*, 2005.

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

[19] L. Ku, S. Sen, E. Learned-Miller, and R. Grupen, "Aspect Transition Graph: An affordance-based model," in *ECCV Workshop on Affordances: Visual Perception of Affordances and Functional Visual Primitives for Scene Analysis*, 2014.

[20] L. Ku, E. Learned-Miller, and R. Grupen, "Modeling objects as aspect transition graphs to support manipulation," in *Proc. of Int. Symp. on Robotics Research (ISRR)*, 2015.

[21] J. M. Wong and R. Grupen, "Intrinsically motivated multi-modal structure learning," in *Proc. of Int. Conf. on Developmental Learning and Epigenetic Robotics (ICDL-EPIROB)*, 2016.

[22] D. Ruiken, M. W. Lanighan, and R. A. Grupen, "Postural modes and control for dexterous mobile manipulation: The UMass uBot concept," in *IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, 2013.

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