Affordance-Based Active Belief: Recognition using Visual and Manual Actions

Dirk Ruiken, Jay Ming Wong, Tiffany Q. Liu, Mitchell Hebert,
Takeshi Takahashi, Michael W. Lanighan, and Roderic A. Grupen

***Abstract*— This paper presents an active, model-based recognition system. It applies information theoretic measures in a belief-driven planning framework to recognize objects using the history of visual and manual interactions and to select the most informative actions. A generalization of the *aspect graph* is used to construct forward models of objects that account for visual transitions. We use populations of these models to define the *belief state* of the recognition problem. This paper focuses on the impact of the belief-space and object model representations on recognition efficiency and performance. A benchmarking system is introduced to execute controlled experiments in a challenging mobile manipulation domain. It offers a large population of objects that remain ambiguous from single sensor geometry or from visual or manual actions alone. Results are presented for recognition performance on this dataset using locomotive, pushing, and lifting controllers as the basis for active information gathering on single objects. An information theoretic approach that is greedy over the expected information gain is used to select informative actions, and its performance is compared to a sequence of random actions.**

I. Introduction

The combination of representation, planning, and action for use in robots that is applicable to open and unstructured domains is an important and challenging milestone. In general, state is only partially observable and the outcome of actions is stochastic. Furthermore, states and actions at the low-level are incompatible with the representations required for robust and efficient planning, error detection, and recovery in the presence of uncertainty.

In this paper, we address the aforementioned issues by developing probabilistic action-based forward models and belief-space planning mechanisms. We employ forward models defined by *aspects*, a geometric constellation of features present in a particular field-of-view. A Dynamic Bayes Net (DBN) is used to fuse the history of observations into a maximum likelihood distribution over the population of such models. We consider the posterior distribution at time t over a history of such observations to be the *belief state* of a recognition problem.

Recognition is posed as the unique identification of an object model that takes the form of a probabilistic transition $f : x_t, a_t \mapsto x_{t+1}$ summarized in an *aspect transition graph* (ATG) for each known object where a_t is action and x_t is state. A belief state is projected forward through a population of ATGs to select actions that will likely decrease the uncertainty of future belief states. The result is the Active Belief Planner (ABP).

The authors of this manuscript are affiliated with the Laboratory for Perceptual Robotics, College of Information and Computer Sciences at the University of Massachusetts Amherst, USA. Corresponding author: ruiken@cs.umass.edu The authors would like to thank 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.

II. Background and Related Literature

There is substantial evidence that human visual learning is facilitated by coupling perception and action [1]. The active vision community advocates exploiting actions that change sensor geometries in order to actively improve confidence in perceptual feedback and information gain [2]. Aloimonos *et al.* introduce the first general framework for active vision in order to improve the quality of tracking results [3]. More recently, Denzler *et al.* demonstrate how altering sensor geometry can be used effectively to deal with limited field-of-view and occlusions in the scene [4].

The partial observability and uncertain nature of these types of problems are handled by the use of partially observable Markov decision processes (POMDPs). Sridharan *et al.* introduce a hierarchical POMDP to plan visual operators to recognize multiple objects in the scene [5]. Eidenberger and Scharinger formulate an approximate solution as a POMDP with a horizon 1 value function. They demonstrated that this approach generates next viewpoint actions that successfully recognize multiple objects in a cluttered scene [6]. Hsiao *et al.* [7] utilize a decision theoretic solution to a POMDP to determine relative pose of a known object. They show that rolling out belief states by just two plies can lead to a drastic decrease in the number of actions, but that multi-ply planning quickly becomes computationally prohibitive.

While most studies in active perception do not consider the use of manipulation, some results exist in the literature that incorporate both vision and manipulation to recognize objects. Hogman *et al.* use the action-effect relation to categorize and classify objects [8]. Browatzki *et al.* [9] use a similar action selection metric and transition probabilities on a view sphere but only employ in-hand rotate actions that change viewpoints between visual keyframes recorded in a model.

Information theoretic planning approaches usually consist of a belief update function, a sensor model that computes the probability of observations given states and actions, and a planner that maximizes the information gain for a task and expected future belief states. A Bayes filter is often used for the belief update function, and the sensor model is either modeled as a Gaussian distribution [10] or learned [11].

In the belief-space framework, reinforcement learning can be applied to learn a policy that maps state to action[12, 13].

Malmir *et al.* propose a method that learns a policy to recognize an object in the robot hand using deep Q network (DQN) [14]. These techniques produce a significant advantage due to the efficiency of action selection. However, they require a large amount of training to learn the policy.

Our approach uses a model-based strategy similar to that of Sen[15], where distributions over populations of affordance-based object models are used to represent belief dynamics that become more accurate as the entropy over object models decreases. Sen showed that by pruning models with insufficient support, it can scale to large numbers of models (up to $10,000$).

We extend Sen's framework by incorporating geometric information into object models. The additional information results in the use of forward models that can predict features in future observations, allow pose estimation, and encode robot-specific interaction options for each model. Moreover, the move away from bags of features techniques results in reduced false positives. In addition, we relax the pruning conditions by terminating rollouts of the belief that are below a set threshold instead of purging the corresponding belief entirely. This way, the system has the potential to recover from errors.

III. Technical Approach

This paper belongs to a class of information theoretic planning approaches to solving belief-space MDPs. We use inexpensive affordance-based models for objects which are the basis of the belief states. Moreover, we employ the *aspect transition graph* (ATG) to propagate belief states forward through candidate actions in the *Active Belief Planner* (Fig. 1). Furthermore, we propose a means for thresholding expansion of belief states with insufficient support so that belief condensation quickly leads to significantly better performance. The terminology used in this manuscript is summarized in Table I.

TABLE I: Relevant notations and symbols used throughout this manuscript.

Notation	Definition
a_t	the **action** selected at time t
s_t	a 3D scene of objects in the world at time t
f_i	the i^{th} **feature** defined in terms of $type \in \mathbb{R}^{3\times3}$ representing Cartesian mean and covariance
$\boldsymbol{f}_t$	an array of environmental features detected at time t in the scene s_t from the current sensor configuration
z_t	**observations** supported directly by features $\boldsymbol{f}_t \subseteq \boldsymbol{f}_t$
x_t	aspect nodes supported by $\boldsymbol{f}_{0:t} \subseteq \boldsymbol{f}_{0:t}$
o_t	the **object** at time t defined in terms of a unique $id \in \mathbb{Z}$, $\mu \in \mathbb{R}^6, \Sigma \in \mathbb{R}^{6\times6}$ representing a unique pose in $SE(3)$ recovered from observation, and a set of tuples $\langle x, a\rangle$ representing the nodes and edges in the ATG.

A. Object Representation

1) **Aspect Geometry:** In general, only a subset of the features attributed to an object (called *aspect*) can be detected from the current sensor geometry. The characteristic geometry of the features comprising an aspect is used to parse sensor data.

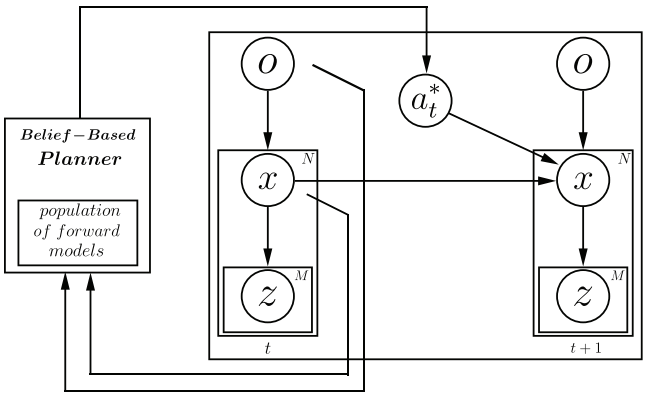

Fig. 1: The recursive filter for condensing belief in the probabilistic object model. Predictions on how actions cause aspect changes (and by inference, changes in the distribution over objects) are used in the belief-based planner to reduce uncertainty in the model space. The belief-based planner is summarized in Algorithm 1.

Algorithm 1 Overview of Belief Planner

1: $t = 0$
2: $\overline{bel}(\boldsymbol{x}, \boldsymbol{o})_0 =$ initialize_model_priors()
3: **while** TRUE **do**
4: $\quad z_t =$ make_observation()
5: $\quad bel(\boldsymbol{x}, \boldsymbol{o})_t =$ update_model_posteriors()
6: $\quad \boldsymbol{a}^*_t =$ plan_action()
7: $\quad$ execute($\boldsymbol{a}^*_t$)
8: $\quad t = t + 1$
9: $\quad \overline{bel}(\boldsymbol{x}, \boldsymbol{o})_t =$ update_model_priors()

To match a model aspect, the features must be of the same *type* (e.g. "visual edge" or "tactile surface normal"), and the geometry must match. There are many ways to match geometric templates. If it is possible to solve the feature correspondence problem easily, then a number of least squares techniques could be used. Alternatively, voting algorithms like RANSAC [16] or Hough transforms [17] can be applied. This paper uses the latter (Sect. III-B).

2) **Aspect Transition Graph (ATG):** Aspect transition models are constructed from extensive, cumulative experience with the object under controlled conditions [18–21]. However, we use these representations as forward models during problem solving behavior in non-ideal contexts that include sensor noise, suboptimal lighting, missing information (occlusion), and extraneous information (distraction) arising from scenes that can contain multiple objects in (initially) unknown arrangements.

The ATG model is defined as a directed multi-graph $G = (\mathcal{X}, \mathcal{U})$ where $\mathcal{X}$ denotes a set of aspect nodes connected by action edges $\mathcal{U}$. Each edge $u \in \mathcal{U}$ is a parametrized skill describing the actions a robot may perform from belief state x_t to change the sensor geometry. These models encode the transitions (along edges in the graph) as learned search distributions built from an agent's previous interactions with the object. Search distributions are defined to be (multivariate) Gaussian distributions $\mathcal{N}(\mu, \Sigma)$ that describe the change in the sensor geometry relative to the object where the robot

has successfully detected the outcome aspect in the past.

Search distributions may be learned autonomously using *structure learning* techniques where transition functions corresponding to re-useable structures in the world are obtained and stored in memory. The models used in this paper are the results of learned search distributions for edges corresponding to the ORBIT action [21]. The remaining search distributions for PUSH, LIFT, and FLIP are hand generated in the current implementation. However, the same approach should be able to learn these distributions as well.

B. Model-Based Aspect Detection

The aspects detected in the features $\boldsymbol{f}_t$ are computed by matching model aspects (feature types and geometry) to the current observation (`make_observation()` in Fig. 1). We would like to estimate the distribution of support for model aspects $p(z_t|\boldsymbol{f}_t)$ as well as the pose at time t of all of the objects that could have generated these observations.

The proposed recognition system extends the generalized Hough transform [17]. Given a random variable representing the distance between features f_i and f_j, $\delta_{ij} = (\mu \in \mathbb{R}, \sigma \in \mathbb{R})$, the area of intersection between the model distance distribution and the observed distance distribution provides a matching score that is used to cast votes for the location of the object frame. The tally over all feature pairs is stored in an accumulator array in $\mathbb{R}^3$ and produces a dominant peak. Using this representation for the aspect geometry, the fully connected graph with all model distance distributions δ_{ij} and a Hough voting table for each pair are stored in the aspect model. Algorithm 2 summarizes the Generalized Hough Transform for classifying model aspects in the scene. In lines (2) and (3), all pairs of features detected in the scene are considered to define possible geometric (sub)structures. In line (4), samples of the distance between features f_i and f_j are used to approximate a Gaussian inter-feature distance distribution. In the loop defined by lines (5)–(8), this quantity is compared to all of the inter-feature distance distributions stored in the aspect models for these features using the Gaussian correlation.

Algorithm 2 Aspect-Based Generalized Hough Transform

```
1: initialize aspect models & Cartesian accumulator array
2: for all 0 ≤ i < NUMFEATURES do
3:   for all i + 1 ≤ j < NUMFEATURES do
4:     (μ_ij, σ_ij)=SAMPLE_SCENE_3D_DISTANCE(i, j)
5:     for all 0 ≤ k <NUMASPECTS do
6:       if ⟨f_i, f_j⟩ ∈ x_k then
7:         score ← score_ij(μ_ij, σ_ij, x_k.δ_ij)
8:         CAST_HOUGH_VOTES(i, j, k, score)
```

C. Belief Update

Belief over aspect nodes $bel(x_t)$ is updated based on the executed action a_t and the new observation z_{t+1} (`update_model_prior()/posteriors()` in Fig. 1). Given transition probability $p(x_{t+1}|x_t, a_t)$, the belief is updated by

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

where $\overline{bel}$ denotes that the posterior is due solely to action a_t. Incorporating the new observation yields

$$bel(x_t) = \eta \; p(z_t|x_t) \; \overline{bel}(x_t), \tag{1}$$

where η is a normalizer. Based on the belief over aspect nodes x_t we can calculate the belief over objects o_t:

$$bel(o_t) = \sum_{x_t} p(o_t|x_t) bel(x_t).$$

D. Affordance Model-Based Planning

The model-based Active Belief Planner (ABP) uses the forward model to expand a search tree through several candidate actions and score future belief states using information theoretic measures (`plan_action()` in Fig. 1). In principle, the search tree can be expanded to any depth, however, without an auxiliary policy structure, the size of the tree and the planning time is prohibitive. We propose that the search tree should be expanded to a fixed depth. Algorithm 3 outlines how the ABP calculates 1-ply greedy information gain for each action based on the available action choices in the ATG. All required information for predicting expected future belief is contained in the object models. Transition probabilities for the process update $p(x_{t+1}|x_t, a_t)$ are stored in the edges of the ATG. The aspect geometry inside the ATG provides an expected observation z_{t+1} for each expected future aspect node x_{t+1}. After performing an observation update using Equation 1, the entropy $H(o_{t+1})$ can be computed. Finally, we calculate the expected information gain for each action a_t over objects as

$$IG(o_t, a_t) = H(o_t) - H(o_{t+1}|a_t).$$

Then, the planner selects the best action based on the highest information gain. The motion planner of the robot examines the feasibility of the selected action, and if the action is not feasible, the planner selects the next best action.

The complexity of the planning algorithm for a greedy 1-ply plan is $O(|A||X|^2)$ where A is the set of eligible actions and X is the set of aspect nodes. The most expensive step in the planner is the measurement update of all aspect nodes x_{t+1} with each expected observation z_{t+1} inside the prediction calculation. But since z_{t+1} is only dependent on the model, all values for $p(z_{t+1}|x_{t+1})$ can be precomputed to greatly reduce computation cost.

Additionally, the influence of expected observations z_{t+1} on the entropy and information gain for an action a_t is minimal if they are generated from expected aspect nodes x_{t+1} with low probability. Excluding such x_{t+1} from the prediction calculation can reduce computational load while having negligible impact on the prediction result. We use threshold $\alpha \max(\overline{bel}(x_{t+1}))$ (relative to the highest current belief) to exclude such belief from the prediction calculation. As a result, the complexity of the problem is reduced to

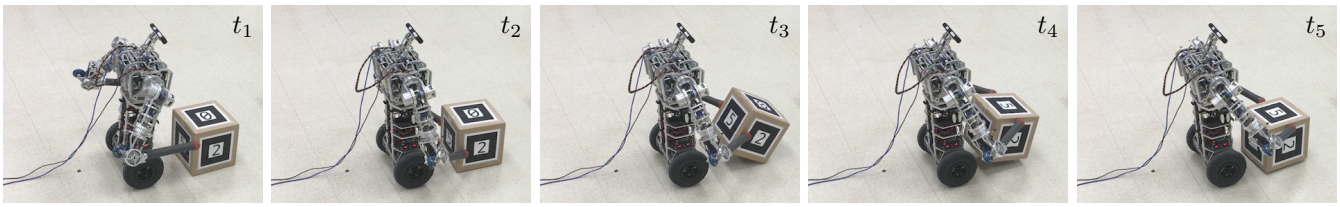

Fig. 2: The uBot-6 performing an information gathering FLIP action on a target ARcube with uniform mass distribution. The action causes a 90° rotation.

$O(|A||X||X'_{t+1}|)$ where $|X'_{t+1}|$ is the number of aspect nodes that have belief above threshold. Typically, belief is initially distributed over a large number of aspect nodes and even greedy search can take a long time after the very first observation. We observed that, even for a model set of highly similar objects, the number of aspect nodes decreases very quickly with every action.

Algorithm 3 Active Belief Planner

$\alpha =$ Future observation update threshold
$\tau_{a_t} = 0$ for all a_t
$\boldsymbol{IG} = \{\}$
for all a_t available in ATG **do**
  for all x_{t+1} **do**
    $\overline{bel}(x_{t+1}) = \sum_{x_t} p(x_{t+1}|x_t, a_t) bel(x_t)$
  for all x_{t+1} **do**
    if $\overline{bel}(x_{t+1}) > \alpha \max(\overline{bel}(x_{t+1}))$ **then**
      $z_{t+1} \leftarrow ATG(x_{t+1})$
      for all x_{t+1} **do**
        $bel(x_{t+1}) = \eta\, p(z_{t+1}|x_{t+1})\, \overline{bel}(x_{t+1})$
      for all o_{t+1} **do**
        $bel(o_{t+1}) = \sum_{x_{t+1}} p(o_{t+1}|x_{t+1}) bel(x_{t+1})$
        $H(o_{t+1}|a_t) = -\sum_{o_t} bel(o_{t+1}) log(bel(o_{t+1}))$
    else
      $H(o_{t+1}|a_t) = H(o_t)$
    $\tau_{a_t} = \tau_{a_t} + \overline{bel}(x_{t+1}) H(o_{t+1}|a_t)$
  $IG(o_t, a_t) = H(o_t) - \tau_{a_t}$
  $\boldsymbol{IG} = \boldsymbol{IG} \cup IG(o_t, a_t)$
while arg max$_{a_t}$ $\boldsymbol{IG}$ is not feasible **do**
  $a^* =$ arg max$_{a_t}$ $\boldsymbol{IG}$
  $\boldsymbol{IG} = \boldsymbol{IG} \setminus IG(o_t, a_t^*)$
return arg max$_{a_t}$ $\boldsymbol{IG}$

IV. The Experimental System

A. ARcube Benchmark

The ARcube domain is designed to control for varying performance in perceptual front ends and varying motor geometry and to focus on the influence of representation and planning for visual and tactile recognition tasks. The nature of the ARcube objects in this domain supports easy generation of large numbers of very similar objects that require combinations of several visual and manual actions to differentiate—an ideal domain for examining planner efficiency.

ARcubes are rigid cubes whose size can be adjusted to meet the requirements of task specifications. Each of the six faces of the cube is marked with a single ARtag. The open-source ARToolKit is available for detecting and localizing the tags as a proxy for more general purpose visual processing. Visual observations of these features establish the location of the center of each tagged face. The permutation of unique tags applied to objects provide a simple means of controlling the complexity of recognition experiments. The number of visually differentiable ARcubes in the model set given n unique ARtags is $|\mathcal{M}| = \binom{n}{6} \cdot 5 \cdot 3!$ enabling easy creation of very large model set sizes.

ARcubes are only partially observable from any single sensor geometry. The natural sparseness of features on any one cube leads to a large degree of ambiguity. Experimental recognition tasks constructed this way are very challenging, requiring multiple sensor geometries to fully disambiguate objects within a set of test objects. Furthermore, the efficiency of the sensing strategy varies based on the composition of the model set.

In addition to surface markings via ARtags, otherwise identical objects can be eccentrically weighted at each face, increasing the size of the model space six folds. These cubes are visually identical and can only be differentiated through the transition dynamics of manual actions.

B. Experimental Platform – uBot-6

The robot platform on which experiments are conducted is the uBot-6 [22], a toddler-sized, dynamically balancing, 13-DOF mobile manipulator developed at the University of Massachusetts Amherst (Fig. 2). The four actions enumerated in Section IV-C were implemented on this platform and encoded in the ATGs for the objects described in Section IV-A. A separate model was constructed for each object.

C. Information Gathering Actions

The following parametrized skills are implemented as information gathering actions for a given population of models:

1) ORBIT – A locomotive action in which the robot drives and reorients itself toward the target object, thus, changing the viewpoint. An angle θ about the world $\hat{z}$-axis is used to parametrize the action.
2) PUSH – The robot extends its arm and pushes along a particular normal on a face of an ARcube. Pushing causes different outcomes depending on the mass distribution—the ARcube will either move approximately straight ahead causing no visual change or pivot approximately 45° about the world $\hat{z}$-axis, revealing a three-feature visual aspect.
3) LIFT – The robot performs a grasp on the object near the geometric center, raises the object, and then places it

down. For a uniformly weighted object, this sequence of actions causes no reorientation or visual change. Given an object with eccentric mass along the top-most face of the cube, the action results in the object flipping $180°$ around the axis between the hands. As result, the mass is re-positioned at the bottom of the cube.

4) FLIP – Like LIFT, this action consists of a sequence of primitives, however, the grasp goals are closer to the body of the robot. For eccentrically loaded objects, this action causes the mass to be at the base of the object. In the uniform mass distribution case, it results in moving the front face to the top and revealing the bottom face of the object.

Figure 2 shows an example of uBot-6 executing a FLIP action.

V. EXPERIMENTAL RESULTS

A large set of simulated and real robotic platform experiments are conducted to compare random action selection against action selection under a greedy 1-ply planner in solving the object identification task.

Simulated experiments contain three model populations ($\mathcal{M}$) of sizes 30, 60, and 120. Each of the three sets are randomly chosen out of the set of 210 possible distinct cubes described in Section III-A. The uBot-6 experiments used a model set of size $|\mathcal{M}| = 30$ that consisted of 18 visually unique objects. Of those, two are selected, and for each, six of its eccentrically weighted counterparts are appended to the set.

A. Simulated Results

A simulation (of the sensor readings and actions) is used to demonstrate the scalability of action selection under the greedy planner with increasing model size. To validate the simulation, trends present in these results are compared to those obtained from the real system. Outgoing transition probabilities are sampled to simulate potential candidates for action outcomes, which are chosen at random based on the probability of outcomes given in the model from the current state. For each run, a randomly chosen object is selected and simulated for 30 actions. The goal is to identify the object's identity within the model space. We compare the greedy ABP to random action selection by running both of these approaches on 30 different objects for each of the three model sets consisting of $|\mathcal{M}| = 30$, 60, and 120. Table II illustrates the results. The ABP approach condenses belief on the **correct** object by reducing the entropy over the population of models. The entropy $H(o_t)$ listed in the table is the entropy of the population of objects at the end of the trial. With $\alpha = 0.1$, the planner takes significantly **fewer actions** and requires **less time overall** in comparison to random action selection. The average time shown describes the time necessary for the object posterior to exceed 0.95 or for the maximum number of actions (30). It is important to note that action time dominates; the planning time per action for the greedy ABP was around $1.5\,s$ in the worst case and on average less than $0.5\,s$. Both approaches theoretically converge to the correct identification as $t \to \infty$, however, these results show only 30 actions or less.

TABLE II: Simulated comparison of random action vs. action selection by the ABP ($\alpha = 0.1$) for different model set sizes.

$\lvert\mathcal{M}\rvert$	Method	Obj. Bel.	Entropy	Actions	Time (s)
30	Random	0.89 ± 0.24	0.29 ± 0.49	14.2	531.3
	ABP	1.00 ± 0.00	0.00 ± 0.00	4.5	173.8
60	Random	0.94 ± 0.16	0.17 ± 0.22	20.2	756.5
	ABP	1.00 ± 0.00	0.00 ± 0.00	5.1	205.4
120	Random	0.93 ± 0.16	0.25 ± 0.48	18.8	686.1
	ABP	1.00 ± 0.00	0.00 ± 0.00	5.9	264.4

B. Object Identification Task on uBot-6

The results of 20 trials using uBot-6 are summarized in Table III; these results describe 10 trials each of the greedy ABP and the random action selection approaches. Statistically significant trends indicate that the greedy ABP outperforms random by condensing belief faster (both in time and number of actions) towards the correct model in comparison to random. These trends agree with the simulation results for the same task. In Figure 3 the slight drops in the correct object belief are a result of action failures; the unlikely outcomes in transitions contribute to increasing belief in other object models that encode such outcomes. The greedy approach has little trouble converging to the correct object, which is similar to the results seen in simulation, but random has more trouble converging to the correct model. Discrepancies between the simulation and robot results can be attributed to differences in the hand-built transition probabilities and estimated action costs from the actual ones on a real robot.

TABLE III: Object scene experiment using uBot-6 ($\alpha = 0.1$)

$\lvert\mathcal{M}\rvert$	Method	Obj. Bel	Entropy	Actions	Time (s)
30	Random	0.48 ± 0.42	0.98 ± 0.72	8.7	1149.1
	ABP	0.98 ± 0.04	0.14 ± 0.21	4.1	680.6

Identification Accuracy: Although entropy in most cases is driven to nearly 0.0, the most important metric for any identification task is accuracy. Table IV shows the accuracy results for the simulated and real robot experiments. The greedy ABP achieved 100% in all tasks, but random misclassified four models in the $|\mathcal{M}| = 30$ runs on uBot-6 and three in simulation as well as one in each of $|\mathcal{M}| = 60,\ 120$. We consider a run to have successful classification if the posterior of the correct model has the greatest belief out of all models at the end of the run.

TABLE IV: Identification accuracy computed over all experiments

	$\lvert\mathcal{M}\rvert$	ABP	Random
Simulation	30	100.00%	90.00%
	60	100.00%	96.67%
	120	100.00%	96.67%
uBot-6	30	100.00%	60.00%

VI. CONCLUSION AND FUTURE WORK

We propose an active recognition system that propagates belief through a forward model called the aspect transition

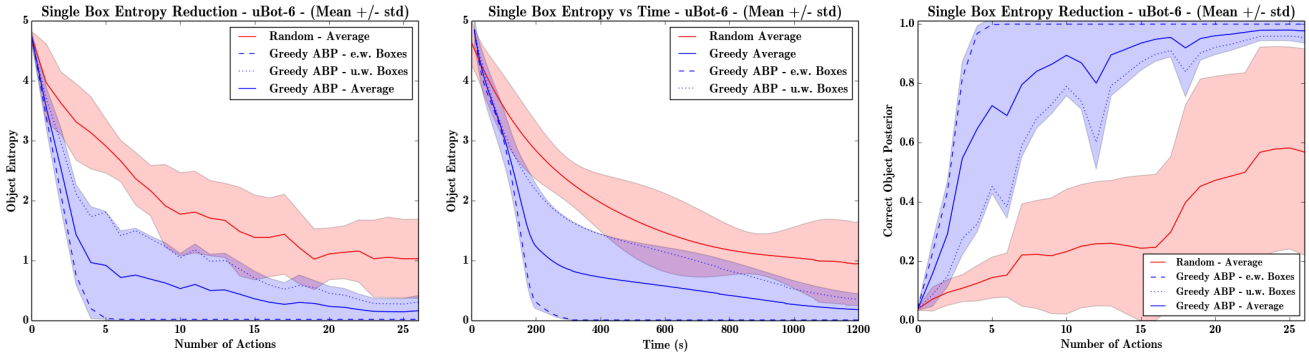

Fig. 3: Single object identification task with uBot-6 using object model set $|\mathcal{M}| = 30$. Plots compare the performance of the greedy ABP (blue) versus random action selection (red). Here, 'e.w. Boxes' = eccentrically weighted boxes and 'u.w. Boxes' = uniformly weighted boxes.

graph (ATG). An ATG has action transitions between aspect nodes that contain sparse features with information regarding their geometric relationships. These aspect nodes represent the belief states. A recursive, hierarchical inference engine, called the Dynamic Bayes Net (DBN), is used to fuse the history of observations and actions into a maximum likelihood distribution over aspect nodes. Using information theoretic measures, the Active Belief Planner (ABP) successfully chooses actions to efficiently identify objects.

Like most information theoretic planners, the system can suffer from intractability as the model space grows. To combat this issue, the ABP terminates belief rollouts when the belief falls below a set threshold. This is a relaxation of pruning conditions set by a previous planning framework, where corresponding beliefs were purged. The relaxed pruning condition allows the system to have the potential to recover from errors. Thresholding the belief rollouts reduces the number of beliefs that need to be considered within the first few actions, which enables multi-ply planning to be run in real-time on robotic systems. In fact, continued work that follows this paper has improved the efficiency of the planner, allowing the robot to plan up to 2 to 3-plies. Additionally, it has been extended to handle recognition of multiple objects in a scene.

Future research is aimed at investigating methods to further reduce planning time. Currently, all actions are chosen exclusively with regard to belief condensation regardless of action cost; a key insight is that actions have varying costs in terms of power consumption and time. Information gain per unit time and energy and the effects on planning and execution remain to be studied in future work.

References

[1] E. Gibson and E. Spelke, *The Development of Perception*, 4th ed. Wiley, 1983.

[2] R. Bajcsy, "Active perception," *IEEE Trans. on Robotics and Automation*, vol. 76, no. 8, pp. 996–1005, August 1988.

[3] Y. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *Int. Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356, 1988.

[4] J. Denzler, M. Zobel, and H. Niemann, "Information theoretic focal length selection for real-time active 3D object tracking," in *Proc. of the Int. Conf. on Computer Vision*. IEEE, 2003, pp. 400–407.

[5] 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, pp. 704–725, 2010.

[6] R. Eidenberger and J. Scharinger, "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)*, 2010.

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

[8] V. Hogman, M. Bjorkman, A. Maki, and D. Kragic, "A sensorimotor learning framework for object categorization," *Trans. on Autonomous Mental Development*, 2015.

[9] 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.

[10] J. Denzler and C. Brown, "Optimal selection of camera parameters for state estimation of static systems: An information theoretic approach," *University of Rochester Tech. Rep*, vol. 732, 2000.

[11] M. Huber, T. Dencker, M. Roschani, and J. Beyerer, "Bayesian active object recognition via Gaussian process regression," in *Proc. of Int. Conf. on Information Fusion (FUSION)*. IEEE, 2012.

[12] F. Deinzer, C. Derichs, H. Niemann, and J. Denzler, "A framework for actively selecting viewpoints in object recognition," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 04, pp. 765–799, 2009.

[13] L. Paletta, M. Prantl, and A. Pinz, "Reinforcement learning for autonomous three-dimensional object recognition," in *Proc. of 6th Symposium on Intelligent Robotic Systems*, 1998, pp. 63–72.

[14] M. Malmir, K. Sikka, D. Forster, I. Fasel, J. R. Movellan, and G. W. Cottrell, "Deep active object recognition by joint label and action prediction," *arXiv preprint arXiv:1512.05484*, 2015.

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

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[17] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.

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

[19] ——, "Aspect transition graph: An affordance-based model," in *European Conference on Computer Vision, 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 the Int. Symposium on Robotics Research*, August 2015.

[21] J. M. Wong and R. Grupen, "Intrinsically motivated multimodal structure learning," in *Proc. of Int. Conf. on Development and Learning and on 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 *Proc. of IEEE-RAS Int. Conf. on Humanoid Robots (Humanoids)*, Oct 2013.