Action-Based Models for Belief-Space Planning

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

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

I. Introduction

An intelligent agent must reason about its own skills, and about the relationship between these skills and goals under run-time conditions. This requires the agent to represent knowledge about its interactions with the world in a manner that supports reasoning [4]. Since the early 1970s, the AI and robotics communities have been concerned with the design of efficient representations that support modeling and reasoning. However, most of these representations tend to tackle only one part of the problem—making either the modeling or the reasoning problem easier.

This paper addresses these dual problems of modeling and reasoning by employing a representation grounded in the robot's own actions and perceptions [3]. Our description of state is domain general, as it is computed directly from the status of executable actions and not hand built for a specific task. The relationship between state and action is captured using probabilistic data structures that model objects in the environment [14]. We present a planner that exploits the uniform description of state and the probabilistic models to plan efficiently in partially observed environments.

II. Related Work

Planning based on belief was introduced by Sondik [17] [16] in solving the optimal control problem characterized by the partially observable Markov decision processes (POMDPs). The value iteration algorithm for solving POMDP was further improved by many authors [5] [11] to solve larger problems. However, the maximum number of states these algorithms can handle is still largely restricted since a POMDP planner must reason in the continuous belief space that has a dimension equal to the number of states minus one.

In this paper, we select a recognition task similar to the simultaneous localization and mapping (SLAM) problem introduced in [8] to demonstrate the capability of our model. Instead of building a map while localizing the robot, our task requires simultaneous object modeling and recognition (SOMAR). The SOMAR problem we formalized in this paper has the number of states proportional to the number of aspects the robot has observed. Unlike most POMDP problems, the number of states increases as more measurement data is gathered in the SOMAR problem; therefore estimating the optimal value function for all current belief intervals does not solve the problem. In our work, a greedy planner is introduced that leads to the lowest expected entropy based on the current belief.

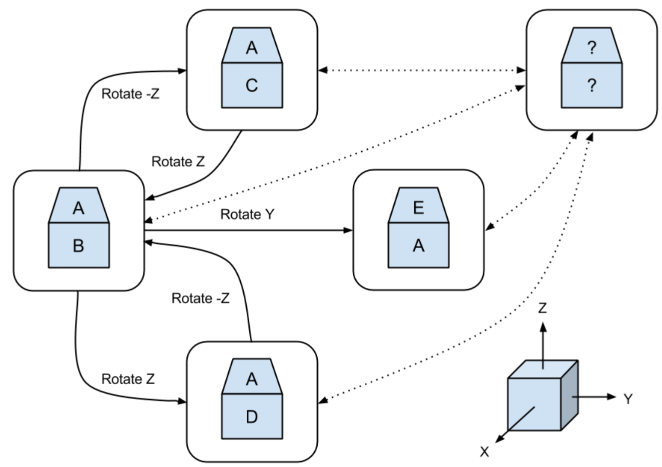

Fig. 1. An example of an incomplete aspect transition graph (ATG) of a cube. Each aspect consists of an observation of two faces of the cube. The lower right figure shows the coordinate frame of the actions and the aspect in the upper right is the "collection node" representing all unknown aspects of the object that may be present. Each solid edge represents a transition between aspects associated with a particular action. Each dotted edge is a transition that may not yet have been observed.

Some recent work [12] [10] [9] in computing optimal solutions for the POMDP problem have focused on solving this problem in Gaussian belief spaces where beliefs are modeled as Gaussian distributions. However, Gaussian distributions are not suitable for modeling beliefs in the SOMAR problem where states are defined as aspects of multiple objects. In [13] a sample-based approach to belief space planning is introduced to handle non-Gaussian belief state. In our work, the non-Gaussian beliefs for all states are propagated through a history of observation. The belief update is simplified by using one collection state that represents all unknown states for each object model.

One of the difficulties in belief space planning is to estimate future observations. In [12], future observation is modeled based on Gaussian distribution and under the assumption that the maximum likelihood observation is always obtained. In our work, the probability of observing a measurement data among a subset of possible future observations is estimated based on past observations stored in object models. The estimation gets more accurate as more information is memorized in object models.

In [1], psychologist J.J. Gibson introduced the term affordance as the properties the environment affords the animal; affordance can be used to explain how "value" or "meaning" of things in the environment is being perceived. Our action-based models are based on this interactionist view of perception and action that focus on learning relationships between objects and actions specific to the robot. An approach to bind affordances of objects with the robot was also introduced by Stoytchev [18]. In his work, the robot learns sequences of actions that will lead to invariant features on objects through random exploration. Learned behavior sequences and invariant features are then stored in the Affordance Table. In our work, we use a graph representation that memorizes all perception feedbacks caused by actions instead. Our approach learns a model that is not restricted to finding invariants, and has the capability of performing belief space planning.

A model similar to the action-based model employed in this work was first introduced in Sen's work [15]. In this paper, we introduce a mechanism for learning these models without supervision and a method for applying belief space planning on these models. In our previous work [14], a planner that executes actions based on mutual information is proposed. In this paper, we modified this planner to handle incomplete models and conducted an experiment to compare different planning algorithms.

III. Model Learning

A. Aspect Transition Graph

Aspect Graphs were first introduced to represent shape [7] [2] in the field of computer vision. An Aspect Graph contains distinctive views of an object captured from a viewing sphere centered on the object. The Aspect Transition Graph (ATG) introduced in this paper is an extension of this concept. In addition to distinctive views, the object model summarizes how actions change viewpoints or the state of the object and thus, the observation. This limits the model to a specific robot, but allows the model to present object properties other than viewpoint changes. Extensions to tactile, auditory and other sensors also become possible with this representation.

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

TABLE I
Notation

Notation	Definition
x_t	the aspect at time t
z_t	the measurement data at time t
a_t	the control data at time t
$bel(x_t)$	$p(x_t \| z_{1:t}, a_{1:t})$
$\overline{bel}(x_t)$	$p(x_t \| z_{1:t-1}, a_{1:t})$
$\mathcal{M}$	the current robot memory
G_i	an ATG in memory, $G_i \in \mathcal{M}$
$\|G_i\|$	the number of total aspects in an ATG G_i
O_i	the object given to the robot at the ith trial
o_j	the object labeled id j
$\mathcal{O}$	the set of objects in the world, $\forall j \ o_j \in \mathcal{O}$
$\|\mathcal{O}\|$	the total number of objects in the world
$\mathcal{S}_T$	the set of objects given to the robot up to the Tth trial, $O_i \in \mathcal{S}_T \quad i = 1 \ldots T$
$\mathcal{X}_j$	the set of robot states that represents o_j
$\mathcal{U}_j$	the set of action edges in o_j
$\|\mathcal{F}\|$	the number of possible features

The robot memory $\mathcal{M}$ is defined as a set of ATGs that the robot created through past interactions. Each ATG in the robot memory represents a single object presented to the robot in the past.

B. Incomplete Models

The ATG of an object is complete if it contains all possible aspect nodes and node transitions. However, in practice, when ATGs are learned through exploration they are almost always incomplete. In addition, an object might be represented by multiple (incomplete) ATGs. A complete model is more informative but harder to learn autonomously. In this paper, we will focus on handling incomplete object models. Figure 1 shows an example of an incomplete ATG of a cube object.

Assuming that an object has a total of $|G|$ aspects, if the robot has already observed $|\mathcal{X}|$ aspects on this object, a naive way to build an incomplete object model is to add $|G| - |\mathcal{X}|$ unknown aspects to the model and connect them with possible action edges. To make the calculation more efficient, each of our ATG models have a single collection node representing all unobserved aspects. The belief of a collection node is defined as the probability that the robot is currently viewing an unobserved aspect of the object this ATG model represents. By specifying the transition probability between an observed and unobserved aspect, the belief of each state can be updated using the Bayes Filter Algorithm (Figure 2).

C. Conditional Update

For each ATG in the robot memory $\mathcal{M}$ we do a conditional update after observing each new measurement z_t. If the new observation tuple (z_{t-1}, a_{t-1}, z_t) cannot be generated by the current ATG, we augment the ATG to keep track of what the

1: **procedure** BAYES FILTER($bel(x_{t-1}), a_t, z_t$)
2: **for** $\underline{all}\ x_t$ **do**
3: $\overline{bel}(x_t) = \sum_{x_{t-1}} p(x_t|a_t, x_{t-1}) \cdot bel(x_{t-1})$
4: $bel(x_t) = p(z_t|x_t) \cdot \overline{bel}(x_t)$
5: **end for**
6: $normalize(bel(x_t))$
7: **end procedure**

Fig. 2. Bayes Filter Algorithm

ATG would be if it matches the observation. If a new aspect node is created during the conditional update to match the new observation, the belief associated with the collection node representing all unobserved aspects that will be transitioned to this newly created node. If a new observation tuple is in conflict with existing nodes or edges in the ATG, the new observation is discarded and the belief of the collection node is reset to zero.

Ideally, if we have high certainty that the given object is identical to the object an ATG in $\mathcal{M}$ represents, saving the augmented ATG representing this object might be beneficial. However, it is unlikely that we can be 100% sure that the two objects are identical with a finite number of observations. For the purpose of this paper, we simplify the problem by not saving the augmented nodes and edges to avoid a false match that might contaminate the robot memory.

D. Identify Novel Objects and Recognize Memorized Objects

An ATG is added to the robot memory $\mathcal{M}$ only if the presented object is judged to be novel. A novel object is defined as an object that has not been presented to the robot in the past. Although the robot might not have seen all the objects or all the aspects of each object, to simplify this problem we make this very limiting assumption that the robot knows that $|\mathcal{O}|$ objects exist in the environment and each object has $|G|$ aspects. If the robot assumes that there are more objects in the environment or more aspects of an object then there actually are, it will bias the judgment toward novelty.

Let $\mathcal{S}_{T-1}$ denote the set of objects that have been presented to the robot in the first $T-1$ trials. Given a sequence of observations $z_{1:t}$ and actions $a_{1:t}$ during trial T, the probability that the object presented during trial T, O_T, is novel can be calculated;

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

Where o_i is an element of set $\mathcal{O}$ designating all of the objects in the environment. Element x_t of set $\mathcal{X}_i$ describes all the aspects comprising object o_i. The conditional probability $p(x_t|z_{1:t}, a_{1:t})$ of observing an aspect is inferred using a Bayes filter. Object O_T is classified as novel if $p(O_T \notin \mathcal{S}_{T-1}|z_{1:t}, a_{1:t}, \mathcal{M}) > 0.5$.

If a particular object is judged to be a previously observed object, we associate it with the ATG that is most likely to generate the same set of observations. The posterior probability of object o_i is calculated by summing the conditional probability of observing aspect x_t over all aspects comprising object o_i,

$$p(O_T = o_i|z_{1:t}, a_{1:t}, \mathcal{M}) = \sum_{x_t \in \mathcal{X}_i} p(x_t|z_{1:t}, a_{1:t}). \tag{2}$$

E. Bayes Filter Algorithm

The posterior probability of an aspect is calculated after each measurement and control update using the Bayes Filter Algorithm [19]. The algorithm is stated in Figure 2, where $\overline{bel}$ is the posterior probability $p(x_t|z_{1:t-1}, a_{1:t})$ after executing a new action a_t and bel is the posterior probability $p(x_t|z_{1:t}, a_{1:t})$ after observing a new measurement z_t. Line 3 is the control update step and line 4 is the measurement update step.

The initial belief over aspects is determined based on the number of aspect nodes and ATGs in the memory. Assuming that there are $|\mathcal{M}|$ ATGs in the memory and $|\mathcal{X}_i|$ aspect nodes are observed in G_i, the initial belief is given by

$$bel(x_0) = \frac{1}{|G_i| \cdot |\mathcal{M}|} \tag{3}$$

$$bel(x_0^u) = \frac{|G_i| - |\mathcal{X}_i|}{|G_i| \cdot |\mathcal{M}|} \quad , \quad x_0^u \in G_i, \tag{4}$$

where x^u is the collection node representing all unobserved aspects in G_i and $|G_i|$ is the number of total aspects in G_i. In this paper we assume all ATGs have the same $|G_i|$.

We assume that transitions between aspects are deterministic; given the current aspect, the same action always leads to the same next aspect. Therefore, each aspect only has one outward action edge of the same type. The transition probability $p(x_t|a_t, x_{t-1})$ in the control update step for each aspect can be calculated by counting how many possible aspect nodes (including the collection nodes) the current aspect can transition to.

To simplify the problem, we also assume that there is no noise in the measurement data. Therefore, the observation probability $p(z_t|x_t)$ would be either 1 for a match or 0 for a mismatch. Note that, although there is no uncertainty in the measurement data, we still have uncertainty over aspects since different objects could generate the same observation z_t.

IV. TASK-LEVEL PLANNING

A. Minimizing Uncertainty

The challenge of integrating task-level planners with noisy and incomplete models requires that we confront the partial observability of the state while building plans. Since the true state of the system cannot be observed, it must be inferred from the history of observations and actions. Our planner belongs to a set of approaches (for example [12]) that select actions to reduce the uncertainty of the state estimate maximally with respect to the task.

Object recognition can be viewed as one such task in which the uncertainty over object identities (as quantified by the object entropy) is reduced with each observation. Selecting the action a_t that minimizes the expected entropy of the distribution over elements of set O_T representing the object identity reduces the uncertainty over object identities the most after the next observation z_{t+1};

$$\begin{aligned}
&\operatorname*{argmin}_{a_t} E(H(O_T|z_{t+1}, a_t, z_{1:t}, a_{1:t-1})) \\
&= \operatorname*{argmin}_{a_t} \sum_{z_{t+1}} H(O_T|z_{t+1}, a_t, z_{1:t}, a_{1:t-1}) \times \\
&\quad p(z_{t+1}|a_t, z_{1:t}, a_{1:t-1}). \qquad (5)
\end{aligned}$$

Where H is the entropy associated with the random variable. The entropy is zero if the state is uniquely determined; it reaches its maximum if all states are equally likely;

$$\begin{aligned}
&H(O_T|z_{t+1}, a_t, z_{1:t}, a_{1:t-1}) \\
&= \sum_{o_i \in \mathcal{O}} p(o_i|z_{1:t+1}, a_{1:t}) \log p(o_i|z_{1:t+1}, a_{1:t}). \qquad (6)
\end{aligned}$$

The posterior probability $p(o_i|z_{1:t+1}, a_{1:t})$ can be calculated by updating the existing belief using the Bayes Filter Algorithm. The prior probability $p(z_{t+1}|a_t, z_{1:t}, a_{1:t-1})$ of observing z_{t+1} given past observations can be calculated by summing the probability of all aspects generating observation z_{t+1},

$$\begin{aligned}
&p(z_{t+1}|a_t, z_{1:t}, a_{1:t-1}) \\
&= \sum_{o_i \in \mathcal{O}} \sum_{x_{t+1} \in \mathcal{X}_i} p(x_{t+1}|a_t, z_{1:t}, a_{1:t-1}) \cdot p(z_{t+1}|x_{t+1}). \qquad (7)
\end{aligned}$$

Where the posterior probability of an aspect $p(x_{t+1}|a_t, z_{1:t}, a_{1:t-1})$ is updated using the Bayes Filter Algorithm.

B. Approximation

The runtime for calculating the expected entropy given an action is $O(|\mathcal{F}| \cdot |\mathcal{O}|^2 \cdot |\mathcal{X}|)$. To speed up the calculation, an approximate expected entropy for each action is calculated instead:

$$\begin{aligned}
&E(H(O_T|z_{t+1}, a_t, z_{1:t}, a_{1:t-1})) \\
&\simeq \sum_{z_{t+1}} H(O_T|z_{t+1}, a_t, z_{1:t}, a_{1:t-1}) \cdot p(z_{t+1}|a_t, z_{1:t}, a_{1:t-1}) \times \\
&\mathbb{1}_{(threshold, \infty)}(p(z_{t+1}|a_t, z_{1:t}, a_{1:t-1})). \qquad (8)
\end{aligned}$$

Here $\mathbb{1}(\cdot)$ is the indicator function and the $threshold$ is set to $1/|\mathcal{F}|$ in this experiment. The value of the indicator function is 1 if the input is greater than $threshold$ and 0 otherwise.

If an observation z_{t+1} is unlikely to be observed, the entropy $H(O_T|z_{t+1}, a_t, z_{1:t}, a_{1:t-1})$ will not be calculated.

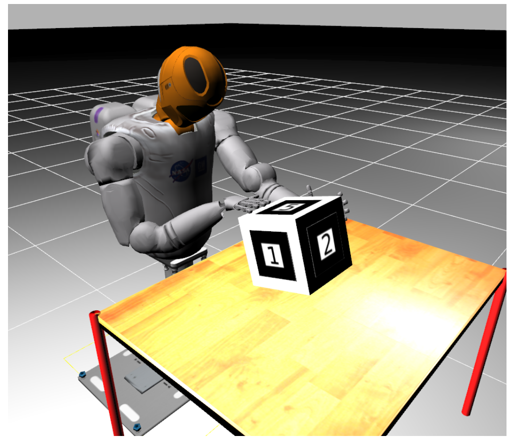

Fig. 3. The simulated Robonaut 2 interacting with a box.

The approximate expected entropy will be lower than the actual entropy, however it effects all estimates in the same way and should allow us to identify the action that leads to the minimum entropy most of the time.

V. Experiments

A. Environment

We evaluated the capabilities of the proposed model and planner using the Robonaut 2 simulator (Figure 3) and an exclusively kinematic simulator. The kinematic simulator runs much faster and is used to collect more data for comparing different planners. The simulation environment contains 100 unique objects called ARcubes that consist of a $28cm$ cube with unique combinations of ARtags on the six faces; 12 different ARtag patterns are used in this experiment. In an ATG for an ARcube, an aspect consists of ARtag features observed on the top face and the front face. As in Figure 1 each ATG has 24 unique aspects and each aspect has 132 different pattern combinations. For the sake of simplicity, we assume that an object does not have two faces with the same ARtag.

In the Robonaut 2 simulator, the simulated Asus Xtion sensor located in Robonaut 2's head is used for visual and depth input. The ARtags located on the ARcubes are detected and recognized using the ARToolkit [6] and are classified as the top or front ARtag based on the 3 dimension position. To simplify the experiment, we are not distinguishing different orientations of the ARtag.

The robot can perform 3 different manipulation actions on the object: 1) flip the top face of the cube to the front, 2) rotate the left face of the cube to the front, and 3) rotate the right face of the cube to the front. The robot will be able to execute any of these actions under the condition that it observes both of the ARtags. If the ARcube is tilted, too far or too close to manipulate, the robot will try to adjust the cube till it is in the right position. These adjustment actions are not stored in the ATGs.

TABLE II
THE SUCCESS RATE OF AN INFORMATION THEORETIC PLANNER IN RECOGNIZING THE OBJECT (10 ACTIONS PER TRIAL)

Test	Correct Identification	Correct Recognition	Success Rate
1	80/100	20/21	79%
2	79/100	25/27	77%
3	87/100	21/25	83%
4	78/100	26/28	76%
5	84/100	24/27	81%
average	81.6%	90.7%	79.2%

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

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

B. Simultaneous Object Modeling and Recognition

To address the dual problem of modeling and reasoning, we formalize the problem of achieving this as simultaneous object modeling and recognition (SOMAR). The goal of SOMAR is to have the robot build up a set of object models through interacting with random objects one at a time. The task is evaluated based on whether the robot can identify novel objects and recognize which object model it corresponds to if it have been observed in the past.

This problem is inspired by the simultaneous localization and mapping (SLAM) problem introduced in [8]. Instead of building a map while localizing the robot, our task requires performing object modeling and recognition at the same time. The SOMAR problem is equivalent to a modified SLAM problem where multiple incomplete maps are given to the robot where the goal is to locate the robot in one of the maps or identify that the robot is in none of these maps and start modeling the current environment.

C. Result

Tables II and III show the result of using the planner to recognize the object presented. Each test involves 100 trials and starts with an empty robot memory $\mathcal{M}$. In each trial, the task is to decide which ATG in memory the given object corresponds to or to declare it as a novel object. For each trial, an object is chosen at random and presented to the robot. The robot observes the object and executes an action. This process is repeated 10 times in the first experiment and 20 times in the second experiment. At the end of each trial the robot determines the likelihood that the presented object is novel and the most likely existing object in memory is identified.

The last row in Table II and Table III presents the results averaged over all the tests. The success rate is the percentage of objects correctly classified, that is, correctly identified in memory or declared as a novel object. When 10 actions are performed per trial, the system correctly recognizes the object 90.7% of the time, and correctly determines if the presented object is novel or not 81.6% of the time. The overall success rate is 79.2% in this experiment. When 20 actions are performed per trial, the overall success rate reaches 98.8%.

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

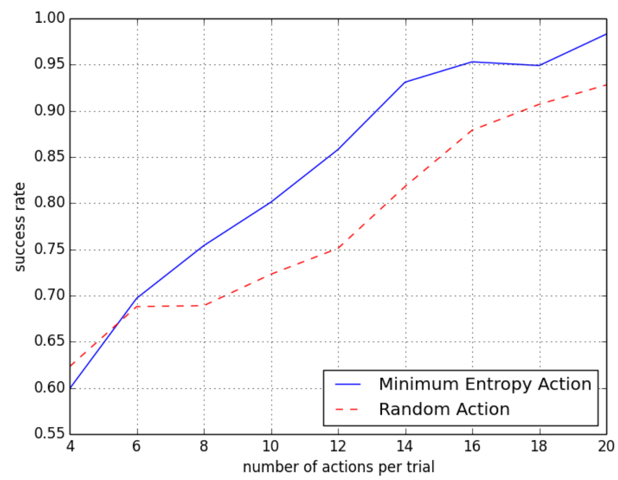

Fig. 4. The plot shows the average success rate of 10 tests as the number of actions per trial are increased. Selecting actions that minimize entropy leads to a higher success rate then selecting actions at random.

VI. CONCLUSION

This paper describes an incremental learning framework for building a memory of objects through interaction. We presented a Bayes framework that performs inference over incomplete object models. We then showed the strengths of combining this representation with a belief-space planner. This information theoretic planner is then compared with a random exploration policy based on a problem we formalized as simultaneous object modeling and recognition. We showed that the belief-space planner leads to a higher success rate then selecting actions at random.

For future work, we are planning to test our algorithm on a greater number of objects with more realistic features and are interested in studying when to merge incomplete object models from different trials. We are also exploring how to represent interactions between multiple objects in the scene and extensions of the idea that can incorporate multi-modal sensory features like tactile data and temporally extended actions.

Acknowledgments

This material is based upon work supported under Grant NASA-GCT-NNX12AR16A.

References

[1] J Gibson. Perceiving, acting, and knowing: Toward an ecological psychology. *chap. The Theory of Affordance). Michigan: Lawrence Erlbaum Associates*, 1977.

[2] Ziv Gigus and Jitendra Malik. Computing the aspect graph for line drawings of polyhedral objects. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(2):113–122, 1990.

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

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

[5] Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1):99–134, 1998.

[6] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *Augmented Reality, 1999.(IWAR'99) Proceedings. 2nd IEEE and ACM International Workshop on*, pages 85–94. IEEE, 1999.

[7] Jan J Koenderink and Andrea J van Doorn. The internal representation of solid shape with respect to vision. *Biological cybernetics*, 32(4):211–216, 1979.

[8] John J Leonard and Hugh F Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Intelligent Robots and Systems' 91.'Intelligence for Mechanical Systems, Proceedings IROS'91. IEEE/RSJ International Workshop on*, pages 1442–1447. Ieee, 1991.

[9] Cindy Leung, Shoudong Huang, Ngai Kwok, and Gamini Dissanayake. Planning under uncertainty using model predictive control for information gathering. *Robotics and Autonomous Systems*, 54(11):898–910, 2006.

[10] Sachin Patil, Yan Duan, John Schulman, Ken Goldberg, and Pieter Abbeel. Gaussian belief space planning with discontinuities in sensing domains. In *Int. Symp. on Robotics Research (ISRR)(in review)*, 2013.

[11] Joelle Pineau, Geoff Gordon, Sebastian Thrun, et al. Point-based value iteration: An anytime algorithm for pomdps. In *IJCAI*, volume 3, pages 1025–1032, 2003.

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

[13] Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Efficient planning in non-gaussian belief spaces and its application to robot grasping. In *International Symposium on Robotics Research*, 2011.

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

[15] Shiraj Sen. *Bridging the gap between autonomous skill learning and task-specific planning*. PhD thesis, University of Massachusetts Amherst, 2013.

[16] Richard D Smallwood and Edward J Sondik. The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

[17] Edward Jay Sondik. The optimal control of partially observable markov processes. Technical report, DTIC Document, 1971.

[18] Alexander Stoytchev. Toward learning the binding affordances of objects: A behavior-grounded approach. In *Proceedings of AAAI Symposium on Developmental Robotics*, pages 17–22, 2005.

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