Robot Planning with a Semantic Map

John G. Rogers III and Henrik I. Christensen

Abstract— Context is an important factor for domestic service robots to consider when interpreting their environments to perform tasks. In people’s homes, rooms are laid out in a specific arrangement to enable comfortable and efficient living; for example, the living room is central to the house, and the dining room is adjacent to the kitchen. The identity of the objects in a room are a strong cue for determining that room’s purpose. This paper will present a planner for an autonomous mobile robot system which uses room connectivity topology and object understanding as context for an object search task in a domestic environment.

I. INTRODUCTION

Domestic service robots can enrich our lives by performing chores in the home such as meal preparation, fetch and carry tasks, security, clean-up, and companionship. Not only will robots in the home make people’s lives more convenient, they will also enable the elderly to stay in their homes instead of in an expensive nursing care facility.

Context is an important cue which can be leveraged to improve the performance of robots in a variety of domestic tasks. For example, a service robot is asked to fetch a snack, but it doesn’t know where the snacks are, because the human didn’t put them away last time. Where should the robot search first for the snack? Without understanding of the semantics of context, the robot will waste its time searching the bathroom when it should be focusing its search in the kitchen, office, and living room.

Many domestic service robot tasks will require the ability to recognize and locate a significant fraction of the objects in their user’s home. If the robot is asked to load the dishwasher, then it must first know where to search for the dishes. Some dishes will be found in the kitchen; however, the robot should not ignore the rest of the house or it will risk failing to perform its duties most efficiently. A complete and exhaustive search of every room is too time consuming for the robot’s busy schedule of duties. A robot must understand the context of where objects should be found in the home to perform object retrieval, meal preparation, and clean-up tasks, to enable it to focus its search where task-relevant objects are most likely to be found.

Context can be represented as a graph connecting related places together with objects which can be found in those places. This type of graph can be used in a bottom up fashion to infer room labels from object recognition, as in our previous paper [11]. Conversely, this graph can also be used in a top-down fashion to predict the presence (or absence) of objects based upon room categories. This type of reasoning will be combined with a planner to select actions on a mobile robot to perform an object search task in this paper.

This paper will describe a novel technique for combining the probabilistic model described in [11] for context with a high-level robot action planner to accomplish an object search task in a domestic environment. The planner will be shown to outperform an uninformed search strategy in a series of experiments performed in a simulation. Preliminary experimental results will also be presented demonstrating the performance of the context-aware planning algorithm finding objects in a real-world experiment on a prototype mobile robot system.

We present related work in section II, and an outline of the algorithms used in this paper in section III. The experiments will be described in section IV and the results will be explored in section V and section VI. An outline of future work will be described in section VII.

II. RELATED WORK

Recent approaches to indoor scene recognition based upon semantic understanding of objects and places have emerged. In [4], the authors demonstrate a technique where category level visual object detection is used with a hierarchical probabilistic model to perform visual scene categorization. The authors use a boosted cascade of several feature types for category level object detection. They use a naïve Bayes model to give a distribution over scene label given the output of the object categorizers.

The use of context in predicting where to find objects was shown in [8]. In this work, a co-occurrence database of object names is built from WordNet and Flikr. This paper demonstrates an algorithm where a query object can be predicted based upon the knowledge of other objects in the scene. The authors also describe a planning algorithm which selects paths for a robot which search for the query object using their model.

Active visual search is the problem of using context to improve the performance of an object search task. Recent work in [1], present a technique were an object search is performed on a mobile robot using probabilistic reasoning about object/place co-occurrence. This technique uses a planner which hypothesizes virtual objects and rooms which potentially help the robot accomplish its visual search task.

In this paper, we extend the idea of using virtual objects and rooms of [1] through the use of our probabilistic model. We leverage the context of places seen in a topological map when considering which unknown region to explore, to be
III. ALGORITHM

Robots operating in the real world must deal with uncertainty and errors in measurement and actuation. A probabilistic representation can be used to acknowledge this uncertainty and reason about it. The probabilistic representation which we have developed is called the probabilistic cognitive model (PCM); it is described in section III-A. The PCM is used to plan robot actions to perform an object search task, described in section III-B. The PCM is used to represent the state space in a partially-observable Markov decision process (POMDP) framework.

A. Probabilistic Cognitive Model

We introduced a probabilistic cognitive model (PCM) for place and object classification using conditional random fields in [11]. Objects and rooms are each represented as nodes in an undirected graphical model; their labels are given by multinomial or categorical distributions. There are two types of edges in this graph: between room nodes which indicate adjacency in a topological map, and between objects and rooms which indicate that object is within that room. Each room as well as each object node which has not yet been recognized or classified is given a uniform prior measurement.

Many of the software components shown in the system diagram in figure 1 were described in detail in our previous work [11]; however, a brief description is provided here for clarity. The locations of objects are mapped with the OmniMapper [12], [17], which is based upon GTSAM [3]. The topological arrangement and extent of rooms are determined from 2D laser scans via Gaussian regional analysis [9]. Objects are segmented from Asus Xtion sensor data (similar to Kinect) by segmenting planar table (or floor) surfaces via a RANSAC [5] algorithm implemented in PCL [13], followed by clustering points lying above the tables. Segments are projected into high resolution DSLR images which are used to extract tight region-of-interest for recognition via a SURF [2] matching algorithm. If the object is not represented in the recognition database, then a bag-of-visual-words [15] representation is classified by a relevance vector machine [16]. Unexplored rooms are detected via a frontier-based exploration algorithm [18], and inserted into the topological map as hypothetical rooms.

The room adjacency model and object-in-room affinity model were trained separately using offline data. In our prior work [11], model parameters were selected by hand. The training data for the room adjacency model was generated by manually extracting the topology of a set of single-story house floor plans. The floor-plan topology was represented as a matrix with a 1 in entry \((i, j)\) if room \(i\) is topologically connected (adjacent) to room \(j\). Additionally, ground truth labels are provided for each room. Since architects do not usually specify which bedroom or space is to be used as the office, we labeled one of the bedrooms as an office in larger 3 and 4 bedroom plans.

The BFGS model parameter training procedure from UGM [14] was used to select adjacency affinity parameters. This training procedure is typically run on a fixed graph topology with many data instances; however, in our case we have heterogenous graph topologies which share model parameters. The training procedure was modified to work with heterogenous graph structures by computing the sum of the negative log likelihood and the gradient across all graph topologies. Additionally, a regularization term was added to the optimization to favor small parameter values. The resulting room adjacency model is shown in table I. In this table, common adjacency relationships are given a higher affinity. For example, bedroom and bathroom have the
highest affinity. Additionally, *living room* and *kitchen* have a large affinity and are likely to be found adjacent to one another.

The object-in-room affinity model was developed by analyzing sentences in the Open Mind Indoor Commonsense Database [7]. This database consists of sentences describing relationships between entities which are indoors, in an office or domestic environment. A series of queries was performed to find the number of sentences in the database which contained both a place label from the set (*kitchen*, *living room*, *bathroom*, *bedroom*, *office*) together with the object class labels and a group of instance examples for each class label. The relative frequency of sentences for an object class is used to construct a model with respect to room label. These models are shown in figure 3.

### Table I

<table>
<thead>
<tr>
<th>Room</th>
<th>Kitchen</th>
<th>Hall</th>
<th>LivingRoom</th>
<th>Bathroom</th>
<th>Bedroom</th>
<th>Office</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kitchen</td>
<td>0.82</td>
<td>0.84</td>
<td>2.14</td>
<td>0.67</td>
<td>0.84</td>
<td>0.77</td>
</tr>
<tr>
<td>Hall</td>
<td>0.84</td>
<td>0.65</td>
<td>1.87</td>
<td>1.55</td>
<td>3.25</td>
<td>1.58</td>
</tr>
<tr>
<td>LivingRoom</td>
<td>2.14</td>
<td>1.87</td>
<td>0.76</td>
<td>0.63</td>
<td>0.57</td>
<td>0.72</td>
</tr>
<tr>
<td>Bathroom</td>
<td>0.67</td>
<td>1.55</td>
<td>0.63</td>
<td>0.75</td>
<td>3.30</td>
<td>0.72</td>
</tr>
<tr>
<td>Bedroom</td>
<td>0.84</td>
<td>3.25</td>
<td>0.57</td>
<td>3.30</td>
<td>0.65</td>
<td>0.66</td>
</tr>
<tr>
<td>Office</td>
<td>0.77</td>
<td>1.58</td>
<td>0.72</td>
<td>0.72</td>
<td>0.66</td>
<td>0.89</td>
</tr>
</tbody>
</table>

### Table II

<table>
<thead>
<tr>
<th>Action</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>Move</td>
<td>-3</td>
</tr>
<tr>
<td>Search</td>
<td>-5</td>
</tr>
<tr>
<td>Examine</td>
<td>-4</td>
</tr>
<tr>
<td>Fetch(correct)</td>
<td>20</td>
</tr>
<tr>
<td>Fetch(miss)</td>
<td>-100</td>
</tr>
</tbody>
</table>

The rewards assigned for performing each action.

---

### B. PCM Planner

Our contribution for this paper is a planning module which uses the PCM described in section III-A to select robot actions to perform an object search task. Briefly described, the PCM Planner hypothesizes the effects of its selected actions on the PCM and chooses the action sequence which is most likely to find the object of interest.

The PCM Planner performs a depth-first-search on a PCM by trying potential actions at each level and hypothesizing new PCM states based on the result of these actions. For these experiments, the set of robot actions is: *Move*, *Search*, *Examine*, and *Fetch*. The *Move* action can be performed to transition to another room which is topologically adjacent to the one in which the robot currently inhabits. The PCM hypothesized by the planner for this action is the same as the current PCM, but with the robot in the desired adjacent room. The *Search* action is used to try to find objects in the current room. The result of the *search* action in the hypothesized next state is that a new object is found by the robot. A uniform prior is placed on this new object; after the posterior is computed with loopy belief propagation (LBP), the context of the current room will affect the label of the hypothesized object. The *Examine* action is selected by the planner to look at a previously segmented or categorized object and try to identify it directly with the SURF-based recognition module. The hypothesized next PCM state upgrades the object categorized label distribution to a clamped, known recognition. The final action, *Fetch*, is terminal and represents the robot making its final choice; choosing this object as the solution to the object search task. It should be noted that the robot doesn’t actually perform the *Fetch* action at this time; the robot currently reports that the requested object has been found.

Reward values are specified by hand at this time and can be seen in table II. The reward values for the *Fetch* action are tuned relative to one another; the penalty for fetching the wrong object is much larger than the reward for fetching the correct object. Because of this disparity, the robot will need to be quite sure that an object is the correct one before it retrieves it. The rewards for the other actions are small negative values which are roughly proportional to the amount of effort and time necessary to execute these actions. Other values of rewards could be specified to favor certain actions. An optimal set of reward values could also be learned with respect to a certain task; however, this is reserved for future work.

The PCM planner chooses actions in a partially-observable Markov decision process (POMDP). In a POMDP, the system state is represented as a belief state, which is a probability distribution over states. In our application, the belief state is represented by the PCM. The specification of a POMDP consists of a representation for the system state, an action
model which describes how the belief state is updated as a result of actions, an observation function, and a reward function. In this application, the reward function provides a positive reward for performing the Fetch action on the desired object, and a strong negative reward for performing the Fetch action on the wrong object, as seen in table II. The robot also receives a small negative reward for performing each of the non-terminal actions to prevent loops or overly conservative behavior. The state transition function is given by the rules for adding hypothesized elements to the PCM as described above. In our implementation, observation actions are implemented as state transition functions.

The PCM Planner expands the search tree of actions until it reaches a terminal state, where expected rewards are computed. The sequence of actions which results in the highest expected reward is then returned, and the first action on that sequence is performed. Consider a sequence of actions where the robot has a starting state for the PCM where it has just observed an object which indicates that it is in the Kitchen, but it is looking for the TV remote, which is most likely to be found in the Living Room, which is probably adjacent to the Kitchen.

The specific function which is optimized by the PCM-Planner is:

\[
E[R(s_0, \{a_i\})] = \sum_s p(a_i, s) E[R(s_i, a_i)]
\]

(1)

\[
s_i = T(s_{i-1}, a_{i-1})
\]

(2)

In this equation, the robot gets a reward for performing an action in a state given by function \( R(s_i, a_i) \), and shown in table II. In some cases, such as fetching an object, the reward function takes multiple values based upon the state. In this case, the expectation of the reward can be computed by analyzing the marginals on the relevant object being fetched. The reward value can be computed for this action as:

\[
E[R(s, FETCH(k))] = p(k = t) F^+ + (1 - p(k = t)) F^-
\]

(3)

Here, \( F^+ \) is the reward for fetching the correct object, which is 20, and \( F^- \) is the reward for fetching the wrong object, which is -100. The planner searches over all sequences of actions and selects the sequence which maximizes equation 1.

Computing optimal policies for POMDPs is intractable in general; but we have exploited sparsity and action dependency to simplify the policy search process, as in [10]. The sparsity property allows the search procedure to visit a relatively small number of possible belief states when selecting the next action. In addition, the action sequence features a hierarchical dependency which further reduces the complexity of the search space.

IV. EXPERIMENT

The experiments in this paper were designed to establish the effectiveness of the PCM Planner algorithm to leverage semantic information to perform an object search task in a domestic environment.

The PCM Planner algorithm was first tested in a simulation environment which abstracted away the performance of the perception components to demonstrate effectiveness. In the simulation scenario, object segmentation, classification, and recognition are replaced with an object simulator which responds to control commands from the PCM Planner to mimic ideal robot performance. The object simulator provides the correct label for recognition if the Examine action is executed within 2 meters of an object, and it provides a distribution on the object which is uniform across classes at 4 meters and becomes more peaked at the correct label between 4 and 2 meters from the target. This is an idealized model of robot performance designed to demonstrate the performance advantage of the PCM planner over an uninformed search strategy.

To establish a baseline for comparison with an uninformed version of the PCM Planner, we replaced its room adjacency and object/room affinities with uniform models. This is the PCM Planner without any context since all adjacency and object affinities are equivalent. The first series of experiments will compare the time to complete an object search task using the PCM Planner with a contextual model of object-room and room-room adjacency relationships, to the time taken by an uninformed version of the PCM Planner.

We also performed preliminary live robot experiments in the Aware Home facility at Georgia Tech [6] using our mobile robot “Jeeves”, which has been shown in our prior work [11]. In these experiments, we placed one object on a table in each room in a conspicuous and easy to find area, and had many un-modeled objects on other tables in the background. For example, we placed a Food object on the kitchen table, and we left the cups, plates, and vases on the counter for which we had not built recognition models.

V. RESULTS

The simulation described in section IV was used with a set of five rooms, shown in figure 4. Objects were sampled from a distribution similar to the one used in selecting the PCM affinity parameters between objects and rooms. Between one and three objects are placed uniformly within each room. The PCM Planner is compared with the uninformed version to locate each of the target object classes.

In each experiment, the robot was started in the Foyer, as seen in figure 4. The robot was given an object class to search for which can be found in one or more of the rooms in the simulated environment. Each object was tried three times with random variation in the specific objects sampled for each room and their positions for both the uninformed and the context-aware PCM planner. Random seeds were reused between the uninformed and the context aware tests so that each option was tried on exactly the same arrangements of objects.

The results of this experiment can be seen in table III. The time taken to find each object class in each run is shown. The median run time is shown in bold. Certain objects such as the Toys and Electronics can be found in the living room, which is the first room explored by both
Fig. 4. The layout of the world used in the simulation experiments. The robot starts in the Atrium, and is given an object class to find. The robot must discover the topological structure of the environment to leverage contextual information to improve the search procedure.

To illustrate one experiment sequence, the robot starts searching the Living Room until it finds an instance of the Toys class. Once the robot found the Toys object, it was able to determine that it was most likely in the Living Room. The probabilistic model posterior belief state after this object recognition is shown in figure 6. Seeing this object has made the robot believe that it is in the Living Room.

The robot now has two exploration frontier hypothetical room nodes; it chooses one of them to explore. Both of these hypothetical room nodes are likely to be Kitchens since the kitchen is likely to be next to the Living Room. The robot searches in this room and finds the target object class, Food. Once the object is recognized, the posterior belief state is updated, as seen in figure 7. The robot then selected this object.

The video attachment which accompanies this paper follows a similar experiment run. Other experiments were run in these accessible rooms such as searching for Toys when starting in the Kitchen, as well as other starting locations for object searches within these rooms. The key result observed was the planner stopped searching for the target object when it realized that it was unlikely to find it in the room as it became more certain of that room’s label. The planner instead determined that another adjacent room would be more likely to contain the target object and it would continue its search there.

VI. CONCLUSIONS

We have presented a technique for leveraging contextual cues in the form of room adjacency and object in room affinity in a partially-observable Markov decision process framework. The problem addressed in the experiments in this paper is to find an element of an object class in an unknown indoor environment. This technique was evaluated with a simulation which simplified the object recognition and classification components and established the usefulness of the technique over an uninformed search. A live robot experiment was presented in which our robot was able to use the planner to leverage these contextual cues to find the target object class. Contextual cues were learned from training data.
home for a long time and has accumulated information about the environment, and a human labeled model represents a system which was provided with an initial tour of the home by its new user. Both of these scenarios are realistic possibilities for domestic service robots.

VIII. ACKNOWLEDGMENTS

This work was made possible through the ARL MAST CTA project. The authors would also like to thank Carlos Nieto for his assistance in performing the experiments.

REFERENCES


Fig. 6. Jeeves has seen a Toys object on a table in the Living Room. The robot is searching for a Food object, and the PCM planner decides to search an adjacent room instead of searching the Living Room further to find this object (as Kitchens are likely to be next to Living Rooms)

and were applied to improve robot performance in an object search task.

VII. FUTURE WORKS

Currently, all probabilistic cognitive model edge affinity and node prior parameters are trained from static offline data. We would like to augment this form of training with an online component to adapt to particular surroundings.

In many ways, the scenario chosen for this paper is more difficult than a domestic service robot would encounter. In this paper, the robot starts with no information about its surroundings – it must construct a new map and figure out what the room labels are each time. If the robot was able to load a previously built model, or if it were given the labels for the rooms, then the object search task could be performed much more quickly. Both of these alternatives should be considered; a robot loading a previously built model corresponds to a system which has been operating in a home for a long time and has accumulated information about