Semantic Robot Programming for Taskable Goal-Directed Manipulation

by

Zhen Zeng

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical and Computer Engineering)
in the University of Michigan
2020

Doctoral Committee:

Associate Professor Odest Chadwicke Jenkins, Chair
Professor Michael Beetz, University of Bremen
Associate Professor Dmitry Berenson
Professor Jessy Grizzle
Professor Benjamin Kuipers

Zhen Zeng

zengzhen@umich.edu

ORCID iD: 0000-0003-4383-3590

DEDICATION

In memory of my father
&
To my mother

ACKNOWLEDGMENTS

This dissertation will not be possible without the tremendous help I get from people surrounding me. First and foremost, I would like to express my deepest gratitude to my advisor, Professor Odest Chadwicke Jenkins, who has helped and supported me to develop research visions and become an independent researcher, and provided a collaborative environment for me to explore and learn. I am very grateful for Chad being more than an advisor, providing incredible amount of support during the toughest time for me and my family as a friend. I would like to pay my special regards to Professor Benjamin Kuipers, whose invaluable support has assisted me to conquer difficulties as I paved my ways towards my research goals. His wisdom has guided me to become a critical thinker. I am also thankful to Professor Michael Beetz, Professor Dmitry Berenson, and Professor Jessy Grizzle for being supportive and serving on my committee. They have broadened my perspectives with constructive feedback and suggestions.

I am fortunate to be surrounded by an inclusive group of friendly, supportive and caring colleagues during my PhD. I enjoyed sharing and discussing research ideas with my colleagues. Many thanks to the outstanding work by Zhiqiang Sui and Zheming Zhou, who provided a firm base for the early work of this dissertation to build upon. I enjoyed collaboratively developing research projects with Yunwen Zhou, Adrian Rfer, Kevin French, Xiaotong Chen, Karthik Desingh, Jana Pavlasek and Emily Sheetz. Our brainstorms always led to sparked innovations and inspiring thoughts.

I would like to thank my significant other, Yuefeng Zhu, for him always being there for me through many moments of happiness, bitterness and crisis. I would also like to recognize my pet bunny, Ubuntu, for healing me and giving me strength in nonverbal ways. At last, I would like to thank my parents with my whole heart for their endless love, support and belief in me.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF ALGORITHMS

ALGORITHM

ABSTRACT

Autonomous robots have the potential to assist people to be more productive in factories, homes, hospitals, and similar environments. Unlike traditional industrial robots that are pre-programmed for particular tasks in controlled environments, modern autonomous robots should be able to perform arbitrary user-desired tasks. It is infeasible to endow robots with programs that can accomplish all possible tasks that users would desire. Thus, it is beneficial to provide pathways to enable users to program an arbitrary robot to perform an arbitrary task in an arbitrary world. Advances in robot *Programming by Demonstration* (PbD) has made it possible for end users to program robot behavior for performing desired tasks through demonstrations. However, it still remains a challenge for users to program robot behavior in a generalizable, performant, scalable, and intuitive manner.

In this dissertation, we address the problem of robot programming by demonstration in a declarative manner by introducing the concept of *Semantic Robot Programming* (*SRP*). In *SRP*, we focus on addressing the following challenges for robot PbD: 1) generalization across robots, tasks, and worlds, 2) robustness under partial observations of cluttered scenes, 3) efficiency in task performance as the workspace scales up from the tabletop to building scale, and 4) feasibly intuitive modalities of interaction for end users to demonstrate tasks to robots.

Through *SRP*, our objective is to enable an end user to intuitively program a mobile manipulator by providing a workspace demonstration of the desired goal scene. We use a scene graph to semantically represent conditions on the current and goal states of the world, where each node denotes an object, and each edge denotes an inter-object spatial relation. To estimate the scene graph given raw sensor observations, we bring together discriminative object detection and generative state estimation for the inference of object classes and poses. By representing the scene graphs with Planning Domain Definition Language, the robot can reason and plan actions to transit the world from current to goal state. The proposed scene estimation method outperformed state of the art in cluttered scenes. With *SRP*, we successfully enabled users to program a Fetch robot to set up a kitchen tray on a cluttered tabletop in 10 different start and goal settings.

In order to scale up *SRP* from tabletop to large scale, we propose Contextual-Temporal Mapping (*CT-Map*) for semantic mapping of large scale scenes given streaming sensor observations. We model the semantic mapping problem via a Conditional Random Field (CRF), which accounts for spatial dependencies between objects in the scene. Over time, object poses and inter-object spatial relations can vary due to human activities in the environment. To deal with such dynamics, *CT-Map* maintains the belief over object classes and poses across an observed environment. We present *CT-Map* semantically mapping cluttered rooms with robustness to perceptual ambiguities, demonstrating higher accuracy on object detection and 6 DoF pose estimation compared to state of the art neural network based object detector and commonly adopted 3D registration methods.

Towards *SRP* at the building scale, we explore notions of Generalized Object Permanence (GOP) for robots to efficiently search for objects. We state the GOP problem as the prediction of where an object can be located when it is not being directly observed by a robot. We model object permanence via a factor graph inference model, with factors representing long-term memory, short-term memory, and common sense knowledge over inter-object spatial relations. We propose the Semantic Linking Maps (*SLiM*) model to maintain the belief over object locations while accounting for object permanence through a CRF. Based on the belief maintained by *SLiM*, we present a hybrid object search strategy that enables the Fetch robot to actively search for objects on a large scale, with higher search success rate and less search time compared to state-of-the-art search methods.

CHAPTER 1

Introduction

1.1 Motivation

Programming robots to perform complex tasks is challenging for users who are non-experts in robotics. There are various tasks that emerge in our daily activities across environments like homes, hospitals, warehouses, and similar environments. Users should be able to deploy autonomous robots and easily customize their behavior to accomplish these tasks. Rather than relying on experts to program robots, as in traditional industrial settings, non-expert users should be able to intuitively program robots to perform high-level tasks such as setting up a table and tidying up a living room.

In this dissertation, we focus on users programming robots to perform *object arrangement* tasks. The *Object arrangement* domain involves tasks that are mainly concerned about object poses and spatial relations between objects. The application scenarios of our work include (but are not limited to) common task scenarios as shown in Figure 1.1. These domains include those where

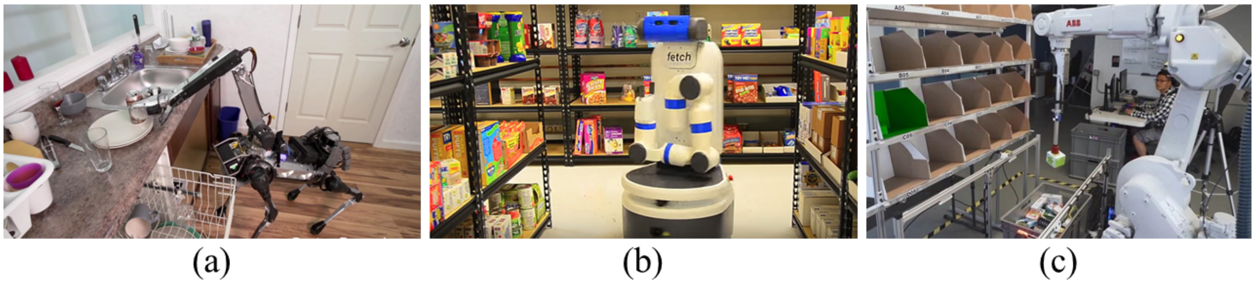


Figure 1.1: Common *object arrangement* tasks in different context. (a) Organizing objects in household environments (Boston Dynamics). (b) Arranging groceries on shelves in a supermarket (Fetch Robotics). (c) Storing items into bins in a warehouse (XYZ Robotics).

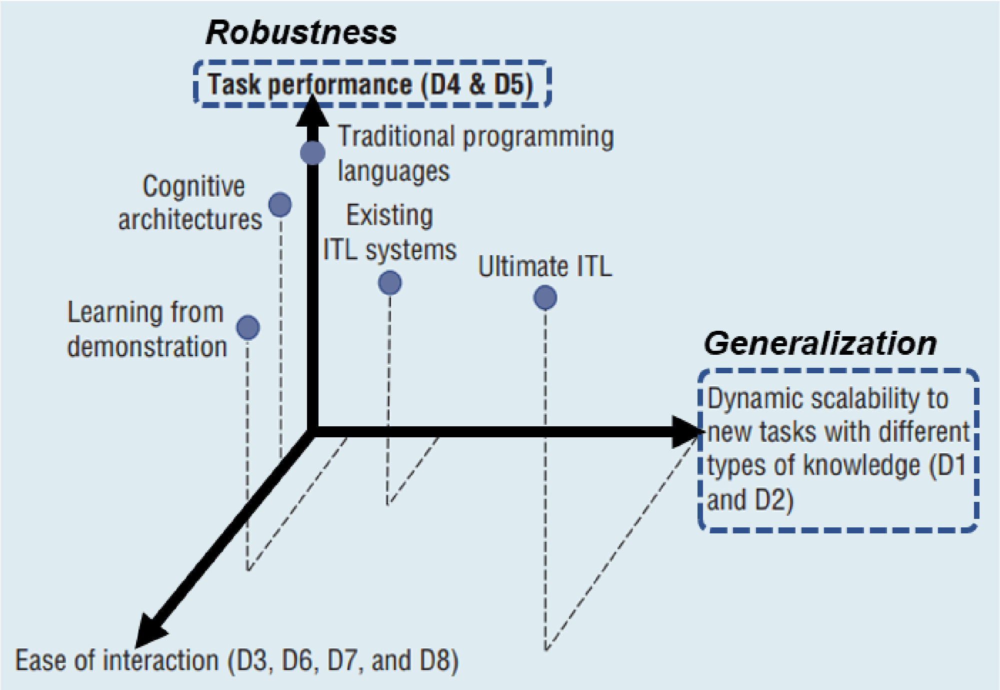

Robustness
Task performance (D4 & D5)
Traditional programming languages
Cognitive architectures
Existing ITL systems
Ultimate ITL
Learning from demonstration
Generalization
Dynamic scalability to new tasks with different types of knowledge (D1 and D2)
Ease of interaction (D3, D6, D7, and D8)

Figure 1.2: Interactive Task Learning (ITL) defines a three dimensional space of methods for robot programming [91]. This dissertation focuses robot programming along the axes of *Generalization* and *Robustness* that will improve future ease of interaction.

the user desires to arrange items in an household environment, arrange groceries on shelves in a supermarket, or store items into bins in a warehouse.

There has been significant progress in robot programming [16], especially robot *Programming by Demonstration* (PbD) [17] [9] in recent years, enabling non-expert users to program robot behaviors. In robot PbD, users program robot behavior through interactively demonstrating the task to the robot. As shown in Figure 1.2, Laird et al. [91] casts robot PbD as one approach to Interactive Task Learning (ITL). ITL defines three important metrics for evaluating different approaches (such as PbD) to programming robot behavior: 1) *Generalization* for dynamic scalability to new tasks with different types of knowledge; 2) *Robustness* in performance of tasks; 3) *Ease of interaction* by end users. We focus our consideration of the state of the art on two of these three dimensions, *Generalization* and *Robustness*.

We layout the literature of robot programming with respect to axes *Generalization* and *Robustness*, as visualized in Figure 1.3. With traditional programming languages such as C++ and

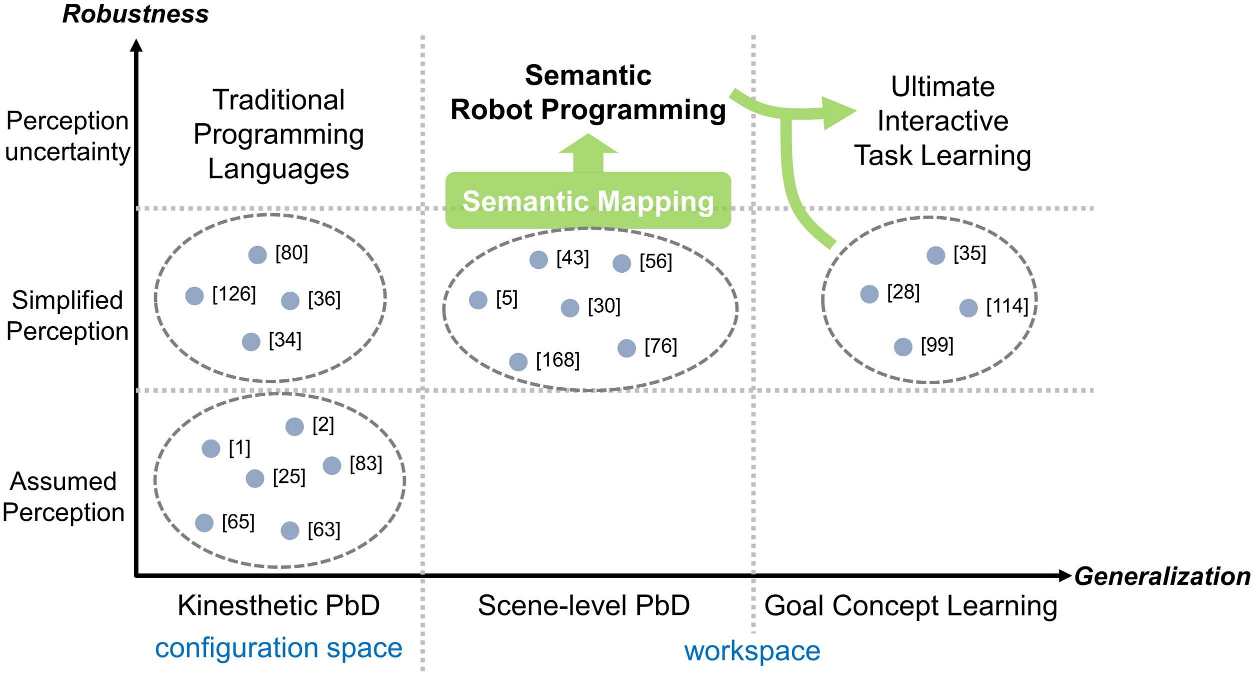


Figure 1.3: A visualization of existing works for robot programming along the axes of *Generalization* and *Robustness*.

Python, one can develop computer programs with high task performance and efficient execution on a robot. However, it can be difficult to generalize such programs to different robots, tasks and other environments. Furthermore, instead of directly interacting with robots to program their behavior, end users have to get over the barriers of traditional programming languages to program robot behavior. In the future, end users should be able to program robot behavior in an intuitive, generalizable and robust way with ultimate interactive task learning.

Programming by Demonstration (or interchangeably Learning from Demonstration) provides the pathway towards ultimate interactive task learning. PbD systems enable end users to program robots directly through demonstrations. End users can directly demonstrate to robots desired trajectories for performing tasks through kinesthetic teaching. End users can also demonstrate steps and/or goal of tasks to robots just as how one teaches kids about various tasks. PbD systems provide more intuitive modality of interaction and better generalization to new tasks.

Kinesthetic PbD [63, 65, 25, 34, 83, 126] involves users demonstrating low-level motion trajectories and/or a sequence of actions in the configuration space to complete a task. Users need to procedurally demonstrate the entire task for a later replay of the demonstrated behaviors on robots. We refer to these works as forms of *procedural robot PbD*. The focus of kinesthetic PbD is to learn the procedural behavior, thus, perception is usually assumed or simplified. Furthermore, kinesthetic PbD requires a large amount of demonstrations for robots to generalize or interpolate motion trajectories and action sequences to arbitrary world states. Consequently, the generalization and robustness of kinesthetic PbD are limited to proprioceptive perception, configuration space demonstrations, procedural programming, and replay of robot behavior.

We propose the next generation of PbD will enable users to program robot behavior through declarative demonstrations in the workspace rather than the configuration space. In contrast to *procedural robot PbD*, we focus on understanding the goal rather than the procedures of a task. Once the goal is learned, the robot can reason how to then achieve the user's intended outcomes. We refer to works for learning goals from users demonstrations as *declarative robot PbD*.

For many tasks that users are faced with, we argue that it is the goal of the task, that matters the most, rather than the motion trajectories or the sequence of actions. There are existing works [43, 5, 168, 30] on scene-level PbD where users demonstrate goal scenes, and the robot should reproduce the goal scene. These works simplify the problem of perceiving the current and goal scenes by using virtual environments, limiting objects to 2D domain, or assuming isolated objects and clean background. As a result, these works have yet to generalize to real-world unstructured environments. New methods are needed for handling perceptual uncertainty in cluttered scenes, such that the robot can robustly infer the goal of a task from user's demonstrations in workspace, as well as infer the current state of the world to reason and plan to perform the demonstrated task.

Thus, we propose to bridge semantic mapping and declarative robot PbD for robustness under perceptual uncertainty, as *Semantic Robot Programming*. In this dissertation, we use semantic concepts such as objects and inter-object spatial relations to describe the goal of a tasks. In the future, robots should be able to learn other types of semantic concepts such as cold and warm, different styles of fried eggs, etc to enrich its own vocabulary to describe task goals. To move towards ultimate interactive task learning, we can further build semantic robot programming on ideas from works in goal concept learning [28, 35, 99, 114].

1.2 Problem Statement: Semantic Robot Programming

This dissertation introduces *SRP* as a declarative approach to the problem of robot programming from workspace demonstrations. Through *SRP*, we focus on understanding the goal of a user demonstrated task in robot workspace and the current state of the world for robots to perform the task. In *SRP*, robots observe the goal scene demonstrated by the user, as well as the initial scene of the environment at a later time. This dissertation handles the perceptual uncertainty in the observations, and grounds these observations with high-level representations of the goal and initial state of the world. Based on the inferred goal and initial states, robots can plan and execute goal-directed actions towards completing the task.

In a formal description of *SRP*, a user demonstrates the desired task to the robot and the observation of the demonstrated goal scene is Z_G. At a later time, when the robot is asked to reproduce the demonstrated task, the robot observes the initial scene Z_I. *SRP* methods focus on estimating the goal state s_G and the initial world state s_I, from which a sequence of actions $\{a_1, a_2, \cdots, a_N\}$ can be planned and executed to transit the world from state s_I to s_G. We assume that the robot is equipped with primitive actions for picking and placing objects with known pre- and post- conditions.

To robustly infer the goal and initial state from robot observations, we focus on dealing with perception challenges caused by partial observations and perceptual ambiguity due to clutter. Furthermore, in order to plan actions to interact with objects spread across the environment, robot should model the uncertainty of objects that are out of its current field of view. To deal with this challenge, we provide ways to maintain belief of objects that are not being directly observed.

1.3 Contributions

To address the problem of *SRP*, this dissertation introduces methods for semantic mapping suitable for declarative robot PbD. We posit semantic maps provide a generic abstraction layer for robot programming. Such semantic abstractions can enable both declarative robot PbD and new forms of procedural robot PbD. In *object arrangement* scenarios, a user can program a robot to perform tasks in an unstructured environment through demonstration of goal scenes. By grounding the demonstrated goal scene and current scene into semantic maps composed by objects along with their axiomatic spatial relations, the robot is able to adapt to arbitrary start states of the world when performing the demonstrated task. Furthermore, the robot can effectively perform the task in a large workspace by maintaining probabilistic believes over objects locations when objects are not being directly observed.

In particular, this dissertation makes the following contributions:

1. **SRP: Semantic Robot Programming for Goal-Directed Manipulation** (Chapter 3). We introduce a new form of robot programming, Semantic Robot Programming, that enables users to declaratively program robots to perform tasks through demonstrations of goal scenes. With *SRP*, robots generalize to arbitrary start states of world when performing the demonstrated task. To achieve this generalization, *SRP* abstracts scene graphs that represent the goal and start state of the world from the observations of the demonstrated goal scene and the current scene. A scene graph is composed by objects present in the scene along with their poses, as well as the axiomatic spatial relations between objects. Abstracted scene graphs for the current and goal scene can be further expressed in Planning Domain Definition Language (PDDL). Thus, robots can plan high-level actions such as pick and place actions to manipulate objects to transit the world from the start to the goal state, achieving the desired inter-object spatial relations as demonstrated by the user. In order to robustly abstract the scene graphs from observations under perceptual uncertainty, we combine discriminative object detection and generative Bayesian state estimation for estimation of object classes

and poses. With estimated object poses, we can derive axiomatic spatial relations between objects based on heuristic geometric assertions.

2. **CT-Map: Robust Contextual Temporal Semantic Mapping for Cluttered Scene at Scale** (Chapter 4). In order to capture the goal scene and current scene of the world at a larger scale, we deal with a sequence of perception sensor data observed across the environment. We develop an on line robust inference method for semantic mapping under perceptual uncertainty present in cluttered scenes. *CT-Map* encodes the contextual dependencies between objects and temporal dependencies of each object across consecutive frames in a Conditional Random Field. Scene state is inferred in terms of objects classes and poses by maximizing the overall posterior probability. Through *CT-Map*, robot can maintain and update the scene estimates as new observations become available. In unstructured cluttered environments, *CT-Map* is able to deal with perceptual uncertainty introduced by 1) partial observations due to objects occluding each other, and 2) perceptual aliasing, e.g., ambiguity introduced by objects that share similar appearance characteristics. We demonstrate that *CT-Map* improved object detection and pose estimation beyond baseline methods that treat observations as independent samples of a scene.

3. **GOP/SLiM: Generalized Object Permanence with Semantic Linking Maps for Active Visual Object Search** (Chapter 5). As we scale up the robot programming framework to deal with tasks in a large scale workspace (e.g. floor level), a critical issue is that visual sensors usually have a limited field of view. When an object that is required for a task is not directly observed in the current field of view, the robot should be able to predict where that object can be located and efficiently search for it. We introduce *Generalized Object Permanence* (GOP) as an understanding of an environment that drives predictions of object locations. We computationally model GOP through a factor graph, by incorporating long-term, short-term, and common-sense knowledge on inter-object spatial relations. We propose a semantic mapping technique, *Semantic Linking Maps* (*SLiM*), that maintains the belief over objects

locations while accounting for inter-object spatial relations modeled in GOP. Based on the maintained belief over object locations in *SLiM*, robots can efficiently search for objects required for a task through our proposed active visual search strategy.

CHAPTER 2

Related Work

This dissertation aims at enabling users to program robots to complete tasks. This chapter discusses related works in the context of robot programming. The major field that is related to this dissertation is Programming by Demonstration (PbD) [9, 18]. In PbD, users program robot behavior through demonstrations. As pointed out by Laird et al. [91], *Generalization*, *Robustness* and *Ease of interaction* are three important metrics for evaluating different approaches (such as PbD) for interactive task learning. In this dissertation, we focus on laying out the related works in PbD with respect two out the three important metrics, *Generalization* and *Robustness*. *Ease of interaction* is not within the scope of evaluation in this dissertation.

We layout the related works in PbD with respect to the *Generalization* and *Robustness* axes, as shown in Figure 1.3. The top right corner is the ultimate goal of the field, i.e., ultimate interactive task learning. Taking the axis of *Generalization* as a reference, existing works mainly divides into three categories:

- **Procedural robot PbD** focus on conveying the procedures or steps to complete a task to the robot, and the robot should replay the programmed behavior at execution time. Existing works in this category is concerned about teaching the robot *how* to complete a task.

- **Declarative robot PbD** focus on informing the robot about the goal of the task, and the robot should accomplish the task through goal-directed behavior at execution time. Existing works in this category is concerned about teaching the robot *what* is the goal of the task.

- **Goal Concept Learning** focus on teaching robot certain concepts (e.g. color, location) to represent goals. In contrast, the previous two categories assume known concepts that can be used to properly represent goals.

Existing works in procedural robot PbD require procedural demonstrations from users for a later replay of the procedural behavior on the robot. The primary focus is to learn to reproduce the demonstrated behavior through trajectory learning. Instead, dealing with perception uncertainty if not the core of these works, thus they rely on assumed or simplified perception.

For many tasks being studied in PbD, it is not necessarily the demonstrated procedures that matters, instead, it is the goal of the task that matters. There have been some works in declarative robot PbD focusing on inferring the goal of the task from demonstrations, and rely on planning method to generate goal-directed behavior on the robot. However, existing works in declarative robot PbD usually rely on assumed perception such as visual aids, or simplified perception with singulated objects and clean background. This dissertation takes the approach of declarative robot PbD, and is mainly concerned about inferring the goal of a a task from unstructured demonstrations, and estimate the current state of the world under uncertainty, with which the robot can use a planning method to generate goal-directed behavior to complete the task.

In the following sections, we will discuss related works in each of the three major categories: procedural robot PbD, declarative robot PbD, and goal concept learning.

2.1 Procedural Robot Programming by Demonstration

Procedural robot Programming by Demonstration (PbD) methods teach robots *how* to accomplish a task by communicating the steps or process required to complete the task. There have been works in PbD mainly focusing on teaching the task procedures at two different levels: trajectory programming via kinesthetic teaching and task structure programming. Existing works on trajectory programming deal with low level procedures for executing an action, while other works on task structure programming consider high level procedures that chain actions into a sequence for accomplishing a task. There are also works that models the world dynamics, and use Reinforcement Learning (RL) techniques to improve programmed robot behavior beyond user demonstrations.

2.1.1 Trajectory Programming via Kinesthetic Teaching

Given user demonstrations of a trajectory of a primitive action, trajectory programming methods aim to learn the low-level skill by encoding the trajectory profile, and regenerate the trajectory at execution time. Dynamical system based methods encode demonstrated trajectories by estimating parameters in a dynamical system, or a distribution. Kormushev et al. [83] use Dynamic Movement Primitive (DMP) [130] to encode full body trajectory on a humanoid for a board cleaning task. Park et al. [128] and Paxton et al. [131] incorporated inverse optimal control and potential fields [75] respectively to adapt learned DMPs to new environments with unseen obstacle configurations. Khansari et al. [74] propose Stable Estimator of Dynamical Systems (SEDS) as a way to model trajecotires via a non-linear, time-independent dynamical system. Instead of modeling trajectories through SEDS, Butterfield et al. [25] learn a single-valued policy function that maps perception to control signals from demonstrated trajectories. This method uses Gaussian process function regressors to learn the policy function. Tanwani and Calinon et al. [161, 26] use Gaussian Mixture Models (GMM) to capture the underlying distribution of demonstrated trajectories. Similarly, Brandl and Peters et al. [22] deploy probabilistic motor primitives (ProMPs) [127] that maintains a distribution over trajectories.

In contrast, Fod and Jenkins et al. [50, 66] automatically derive a library of coupled perceptual and motor routines called *perceptuo-motor primitives* from human movement data, and use this library of primitives for classifying and imitating humen movement demonstrations. In order to discover primitives from demonstrated trajectories, Jenkins et al. [65] propose a spatial-temporal extension to Isomap for automatically clustering time-series data such as human movements into representative clusters, with each corresponding to a primitive action. Lee and Abbeel et al. [93] directly transform a demonstrated trajectory to a new one that adapts to a new start state through non-rigid registration. They register task relevant point cloud in current scene to the observed ones in demonstration, along with the corresponding trajectories. With the advances in deep neural networks, there have been works [95, 49, 185] that start to explore learning end-to-end mapping from perception to motor data given example trajectories. This approach requires enormous amount of data for training towards generalization.

Trajectories represented in the configuration space of the robot are limited to generalization within the configuration space. However, robot actions should be generalized to different states in the workspace, e.g., various object poses in the environment. Vochten et al. [175] propose an object pose invariant trajectory representation and demonstrate on a pouring task. However, this method requires manual specification of the reference frame defined on object parts for each particular task. Dang and Allen [36] consider manipulation actions as a series of sequential rotations and translations, e.g. bottle cap turning, and door sliding. Their approach automatically extract the reference frame of various tasks in terms of translation and rotation axis. Their framework is well suited for articulated object manipulation, but not generalized to other domains such as pick and place tasks. Ureche and Billard et al. [171] extract the reference object at different states of a task execution. However, the reference frame of a task is not necessarily associated with the overall geometry of an object, but a particular part of the object instead.

2.1.2 Task Structure Programming

Often in times more than one primitive action is needed to accomplish a task. For example, in the case of object stacking, one needs to first reach, grasp, then transport, and place. Programming the task structure to the robot is teaching the robot about the high-level task plan or subtasks to complete a task. Given continuous demonstrations of a task, Butterfield et al. [25], Niekum et al. [125], Kulic et al. [88] and Zeng et al. [187] use Hidden Markov Models (HMM) to parse the demonstrated trajectories into segments, where each trajectory segment can be modeled separately as introduced in section 2.1.1. Furthermore, transitions between motion primitives can be modeled through HMM. In contrast, Akgun and Thomaz et al. [4] explore manual specification of segment boundaries, i.e., keyframes, from the human-robot interaction perspective. Nicolescu et al. [124] enabled robots to learn high-level task plan from multiple user demonstrations. The proposed method can take user's feedback during robot practice trials and update the learned task plan. Building on the developments in Neural Program Induction [136] where a latent program representation is learned to generate program outputs, Xu et al. [183] infers a hierarchical task structure from a raw video sequence of a demonstration through one-shot learning. Nevertheless, this approach requires huge amount of meta-learning data towards generalization over adversarial scenes.

Instead of replaying the demonstrated behavior by following the same order of actions, robots should be able to adapt to different world states. Given large-scale distributed data collected from human demonstration through a web-based interface, Crick et al. [34] learn decision trees to navigate a robot through a maze. Decision trees enable the robot to adapt to different world states, i.e., where the robot is located in the maze. Niekum et al. [126] construct a finite-state representation of a demonstrated task, which enables the robot to make decisions on next motion primitive adaptively, and recover from errors. Konidaris et al. [80] construct a skill tree by grouping segmented trajectories that underlay the same skill, and merging different groups together to form a tree structure. Without explicit trajectory segmentation, Grollman and Jenkins [55] infer the subtasks through maintaining a distribution of possible subtask partitions represented by mixture of experts, where each expert is associated with a mapping from perception to action. Vondrak et al. [176] proposed a state-space controller to learn biped control from human demonstrations. The learned controller can be expressed as a finite state machine, which transits to atomic control actions depending on timing or contact events. They were able to reproduce demonstrated human motions such as walking and gymnastics on a virtual agent in different environments. French et al. [51] learned behavior trees [41] [105] from multiple user demonstrations. End users can easily interpret behavior trees to understand what the robot has learned from demonstrations. Recently, Colledanchise et al. [33] have further shown that behavior trees are generalize other reactive robot controller architectures such as subsumption architecture [23], behavior compositions [24] [106], and decision trees [143].

Other works focus on learning a symbolic representation of demonstrated action, and leave it to a planner for deriving a task plan. Abdo et al. [2] infer preconditions and effects of actions from a few demonstrations through clustering in the observation feature space. Similarly, Ahmadzadeh et al. [3] extract the predicates for pre- and post-conditions of actions through the relative position between objects and a known landmark. In addition to discovering the consistency in pre- and post-action states as a way to ground the symbolic action, Jetchev et al. [67] propose to infer the state abstraction into symbols that provides discriminative power for transition model and reward learning.

There have been works in visual or graphical programming approaches [123, 8, 132] for programming robot behaviors. These approaches assume existing library of pre-defined parameterized actions. Users can interactively move action blocks to build a control flow graph, a finite state machine or behavior trees to create robot behaviors through a graphical user interface. Huang et al. [62] allow creation of customized perceptual landmarks. However, registering the customized perceptual landmark under different view points and scene clutter is challenging. Guadarrama et al. [56] propose a natural language interface for programming the robot to rearrange objects in a 2D domain. They ground the spatial relations between isolated objects on a clean table given visual observation and interpret the target object referred by the user based on the verbal description.

2.1.3 Reinforcement Learning on Programmed Behavior

Robot performance on the programmed behavior is limited by user demonstrations without further learning. Reinforcement learning (RL) can be used to improve robot performance beyond the demonstrations. The goal of a RL method is to derive a policy that maximizes the expected accumulated discounted rewards. A policy is a mapping function from the state space to the robot action space. Kober et al. [77] present a general survey of RL. Traditional RL methods do not scale well with high dimensionality in state and actions space. Policy search methods [37] have been commonly adopted to learn a policy in high dimensional state and action space, as often occurred in robotics. User demonstrations can be used to initialize the policy, from which the robot can continue to improve upon through RL. Using user demonstrations for an initial policy can speed up the policy learning process significantly compared to a random initial policy, especially in high dimensional space. Kormushev et al. [82] enable a robot arm to do a dynamic pancake-flipping task through RL on a motion trajectory encoded by DMP, where the DMP is initialized by user demonstrations. Kroemer et al. [85] build a library of motion primitives encoded by DMP from user demonstrations, and learn to sequence theses motion primitives together to perform different manipulation tasks given a reward function.

2.2 Declarative Robot Programming by Demonstration

In many robotic tasks, it is not the motion trajectories or the sequence of actions that are of central importance, but the goal of the task (e.g. setting up a dining table). Many different motion trajectories can correspond to the same goal, such as in pick and place scenarios. A large amount of demonstrations is required for trajectory-based programming to generalize over different world states, e.g. various object poses in manipulation tasks. In addition, trajectory-based programmed behavior on one robot can hardly generalize to another robot with different kinematics. Preprogramming the sequence of actions for accomplishing a task does not generalize well to different initial states of the world, where the same order of demonstrated actions may not apply. Grounding the pre- and post-conditions of actions with predicates offers the potential to generalize over different initial states when combined with a planner. However, existing works rely on simulation environment, fiducial markers, or assume simple environment without scene clutter to ground sensory data with predicates. On the other hand, as the dimension of the world state space increases, more demonstrations are required for the predicates learning process, which essentially involves identifying consistent patterns across the demonstrations in a unsupervised learning manner.

In contrast to procedural robot PbD that concerns about the trajectory or sequence of actions to accomplish a task, declarative robot PbD focus on inferring the goal of a task from demonstrations. In declarative robot PbD, robot accomplishes the task through goal-directed behavior instead of replaying programmed behavior as in procedural robot PbD. Given the inferred goal, robot can either use planning methods or RL methods to achieve the goal. Thus, declarative robot PbD generalizes well to different world states, i.e., can reproduce the inferred goal in any given situation, without relying on user demonstrations to determine procedures towards a goal. Works presented in this dissertation focus on inferring the goal from user demonstrations in workspace, and enabling the robot to reproduce the inferred goal from arbitrary initial states of the world.

Existing works in declarative robot PbD have researched inferring goal conditions from demonstrations. Given sensor observations of demonstrations of a task, the objective is to ground the observations into symbols or predicates. This dissertation focus on grounding demonstrated goal with symbols, and use symbolic planner to generate goal-directed behavior.

There have been works that ground the perceptual observations of demonstrated goals with symbols, and then apply a symbolic planner to generate a sequence of actions starting from a new initial state of world to the goal. The most related work to this dissertation in declarative robot PbD is by Ekvall and Kragic [43]. They propose to learn the goal of a demonstrated task and use a task planner to reach the goal from different initial states of the world. They demonstrated their approach in a table setting example where the goal is represented by object poses in a absolute world reference frame or relative object reference frame. However, their perceptual system is only capable of dealing with isolated cuboid objects following a 2D layout on a clean tabletop, and they assume all objects are fully visible. The limitation of their perceptual system makes it difficult to generalize to unstructured demonstrations in cluttered scenes, as well as tasks that involves 3D spatial relations between objects. Akgun et al. [5] propose to simultaneously learn actions and goals from demonstrations. They use HMMs to represent both the actions and goals observed at manually specified key frames. The emissions are modeled as multivariate Gaussian distributions, therefore goals are probabilistically represented with the emission mean and covariance matrix. Their work assumes that demonstrations have a single object of attention. However, varying attentions across multiple objects are usually involved a complex robotic task that requires multiple steps. More importantly, their goal representation does not directly lead to a plan of the task execution. Consequently, their method does not generalize to new initial states.

As a large amount of user goal demonstrations in workspace becomes available through crowdsourcing, a goal template can be learned. Chung and Cakmak et al. [30] utilize crowdsourcing to collect a rich set of goal demonstrations. The robotic task is to build a 2D object model on a table with basic building lego blocks. There can be many different 2D configurations of the building blocks that correspond to the same object model class. They model the goal for each object model class through a generative graphical model where the distribution of local block patterns are captured. The goal of a user demonstration can be inferred by maximizing a posterior probability over the object model classes. The robot can then pick a user demonstration from the inferred goal class and reproduce the same 2D configuration. Nevertheless, their approach is limited to 2D and discrete domain. Similarly, Toris et al. [168] learn a goal template for object arrangement tasks such as setting up a table. The goal template is represented by GMM that express the absolute or relative poses of objects. The frame of reference of placing each object is autonomously extracted through ranking GMM clusters based on the variances. On the other hand, their approach does not allow user customization of goal from the learned goal template. Moreover, their framework does not handle perceptual understanding because the goal states are learned in a virtual environment. The work presented in this dissertation is evaluated on the same domain of tasks, i.e., wide area pick and place tasks, where robots need to move around the world to pick and place multiple objects. In addition, this dissertation addresses inferring goal states from real sensor data under unstructured demonstrations.

Similar to the related works mentioned in this section, this dissertation focus on inferring the goal from user demonstrations in workspace. Rather than simplifying the perception problems by simulation, virtual environment, limiting the objects to 2D domain, or isolated objects with full visibility on a clean background, this dissertation handles raw sensory data from unstructured demonstration on tasks that require 3D spatial understanding. Proposed approaches in this dissertation handle perceptual uncertainty from observations of cluttered scenes, and maintain a belief over current states of the world from which robot can generate a goal-directed behavior.

There are also works in developing dialogue systems for human to instruct robot to perform tasks. Scheutz et al. [145, 146, 144] develop a framework for natural language dialogue interaction called DIARC (short for Distributed Integrated Affect, Reflection, and Cognition). DIARC can automatically convert natural language instructions into goals formally represented in computational tree logic. Mohan et al. [113] learn goal-oriented hierarchical tasks from situated interactive instructions. They frame the task learning problem as an explanation-based learning problem, and the robot agent is able to learn the structure of the task through interactively quering the instructor for desciptions of the goals and choices of actions. Kirk et al. [76] learn the goal of tasks from multi-modality including visual demonstrations and linguistic instructions. They show less words are required to teach a task when visual demonstrations are also used for learning.

To push the boundaries of generalization of performing tasks across robots, Tenorth and Beetz et al. [166, 15] propose KnowRob and Open-EASE with cross-platform formats for representing knowledge and knowledge processing for autonomous personal robots. They propose to describe knowledge in Description Logic with Web Ontology Language (OWL). Knowledge is represented hierarchically with OWL in terms of classes, instances and properties. With robot perception and control grounded in the knowledge base, their general knowledge processing mechanism can reason about uncertainty and plan action efficiently.

2.3 Goal Concept Learning

Among the works in either procedural robot PbD or declarative robot PbD, it is assumed that the robot has known concepts that can be used to properly represent the goals to be reproduced. For example, in trajectory programming via kinesthetic teaching, the goal to be reproduced is the robot arm trajectory in configuration space. In declarative robot PbD, the goal to be reproduced is represented by a set of predicates defined on a set of pre-defined symbols.

There are cases where the robot is not given the proper state abstraction to represent the goals. For example, the robot has a RGB camera but does not know the concept of “green”. In order to teach the robot tasks such as clean out all green blocks on table, one needs to teach the robot the concept of "green" such that the robot can build on the learned concept to represent goals. There are works focus on learning the proper state abstraction as a way to emerge new concepts, and there are also works that represent the goal of a task as a reward function via Inverse Reinforcement Learning (IRL) methods. Different goal representations result in different ways of deriving a task plan towards the goal. The state of the art in these research directions is described below.

2.3.1 State Abstraction Learning

Without relying on pre-defined symbols such as object colors or positions, other researchers have explored learning symbols or concepts from demonstrations, and use learned concepts to express goals. Chao et al. [28] learn to ground concepts from demonstrations, and these concepts can be used to transfer knowledge for expressing future tasks goals. They start with an initial set of percepts, where each percept corresponds to a state abstraction function, along with an activation function defined in the abstracted state space. Given the start and end frames of a demonstration, they first identify the percepts that undergo state changes, then create new percepts by discovering the consistant pattern in those percepts state values at start and end frames. They are able to ground concepts by building up a library of percepts incrementally in a bottom up manner. The robot can learn the goal of a task with the expressivity of the grounded concepts represented by percepts. The design of the initial set of percepts is of critical importance in their approach. Their perceptual system is limited to mostly planar objects in 2D domain and simple scenes without any clutter. Similarly, Cubek et al. [35] propose to derive a symbolic description of task goals from perception. They first extract key frames where an object velocity falls toward zero, then restore the values of states at those key frames. They discover concepts by clustering in the state space using pre-defined similarity distances. The critical point is to search for clusters under various projections of the state space. However, there are many possible projections that can be applied on the state space. Thus, their method easily becomes intractable as the state dimension grows.

Mohan et al. [114] and Lindes et al. [99] propose to interactively learn grounded representation of words with situated interactive instruction. Robots can learn concepts efficiently because they can interactively request more instructions on unknown or unclear concepts.

2.3.2 Inverse Reinforcement Learning

Instead of representing the demonstrated goal as predicates, researchers have explored inverse reinforcement learning (IRL) [122] methods or similarly Inverse Optimal Control (IOC) [134] to represent the goal as a reward or cost function. As introduced in section 2.1.3 on RL problems, a policy is a function that maps from state to action space, and a reward function is a mapping from state to reward value. The goal of RL methods are to learn a policy that maximize the expected discounted rewards, such that a robot can perform a task following the policy. It can be seen that the reward function is important for learning a policy towards performing a task, yet it is challenging to manually designing a reward function that explicitly specifies the trade-offs between different factors in completing a task.

Given observations of demonstrations of a task, IRL methods learn the reward function that captures the goal, subgoals and constraints involved in the task. Learning the reward function of a task instead of directly learning the policy offers better generalization over different states of the world. Existing works [1, 149] learn reward functions for driving given expert demonstrations. When expert demonstrations are difficult to provide, advices can be incorporated [79] in addition to demonstrations while learning reward function. IRL methods require large amount of demonstrations to learn a reward function that can reproduce the expert policy.

CHAPTER 3

SRP: Semantic Robot Programming

We present the Semantic Robot Programming (*SRP*) [190] paradigm as a convergence of robot programming by demonstration and semantic mapping. Unlike works in procedural robot programming, we focus on understanding the goal rather than the motion trajectories or order of actions of a demonstrated task. In *SRP*, a user can directly program a robot manipulator by demonstrating a snapshot of their intended goal scene in workspace. The robot then parses this goal as a scene graph comprised of object poses and inter-object relations, assuming known object geometries. Task and motion planning is then used to generate goal-directed actions to realize the user's goal from an arbitrary initial scene configuration. Even when faced with different initial scene configurations, *SRP* enables the robot to seamlessly adapt to reach the user's demonstrated goal. For scene perception, we present the Discriminatively-Informed Generative Estimation of Scenes and Transforms (*DIGEST*) method to infer the initial and goal states of the world from RGBD images. The efficacy of *SRP* with *DIGEST* perception is demonstrated for the task of tray-setting with a Michigan Progress Fetch robot. Scene perception and task execution are evaluated with a public household occlusion dataset and our cluttered scene dataset.

3.1 Introduction

Many service robot scenarios, such as setting up a dinner table or organizing a shelf, require a computational representation of a user's desired world state. For example, how is the dinner table to be set, or how is the shelf to be organized. More specifically, what are the objects involved in the task, what are the desired poses of those objects, and what are the important spatial relationships between objects. Towards natural and intuitive modes of human-robot communication, we present the Semantic Robot Programming (*SRP*) paradigm for declarative robot programming over user demonstrated scenes. In *SRP*, we assume a robot is capable of goal-directed manipulation for realizing an arbitrary scene state in the world. A user can program such goal-directed robots by demonstrating their desired goal scene. *SRP* assumes such scenes can be perceived from partial RGBD observations, which has proven a challenging problem in itself.

Goal-directed manipulation requires a true closing of the loop between perception and action, beyond the existing intellectual silos. Advances in object detection [54, 137] from appearance has improved greatly in filtering of background noise and focused attention to objects of interest. However, the applicability of such vision-based methods robot perception remains unclear, especially for the purposes of goal-directed manipulation. This circumstance has given rise to new approaches to semantic mapping [87, 141, 61] to computationally model a robot's environment into perceivable objects with robot-actionable affordances.

We posit semantic mapping offers a springboard to new forms of robot programming, such as Semantic Robot Programming, where semantic maps provide a generic abstraction layer for robot programming. In our approach to this problem, we must bridge the gap of interoperation between semantic mapping and existing methods for goal-directed task planning [48, 92], grasp planning [165] and motion planning [156]. There have been methods for scene estimation [158] from robot RGBD sensing that used scene graphs expressed axiomatically as a semantic mapping abstraction. This abstraction allowed for ready use with modern task, grasp, and motion planning systems. The resulting of closing this loop with a semantic abstraction layer is envisioned to enable portable robot-executable expressions accessible across a variety of modalities, including: natural language, visual programming, and put-that-there gesturing [27, 72]. However, the computational cost of inference over scenes is asymptotically intractable as the number of objects grows. Later work by Narayanan et al. [119] has saved some computational cost by limiting the object pose search region with integrated object detector. However, their work does not abstract the scene

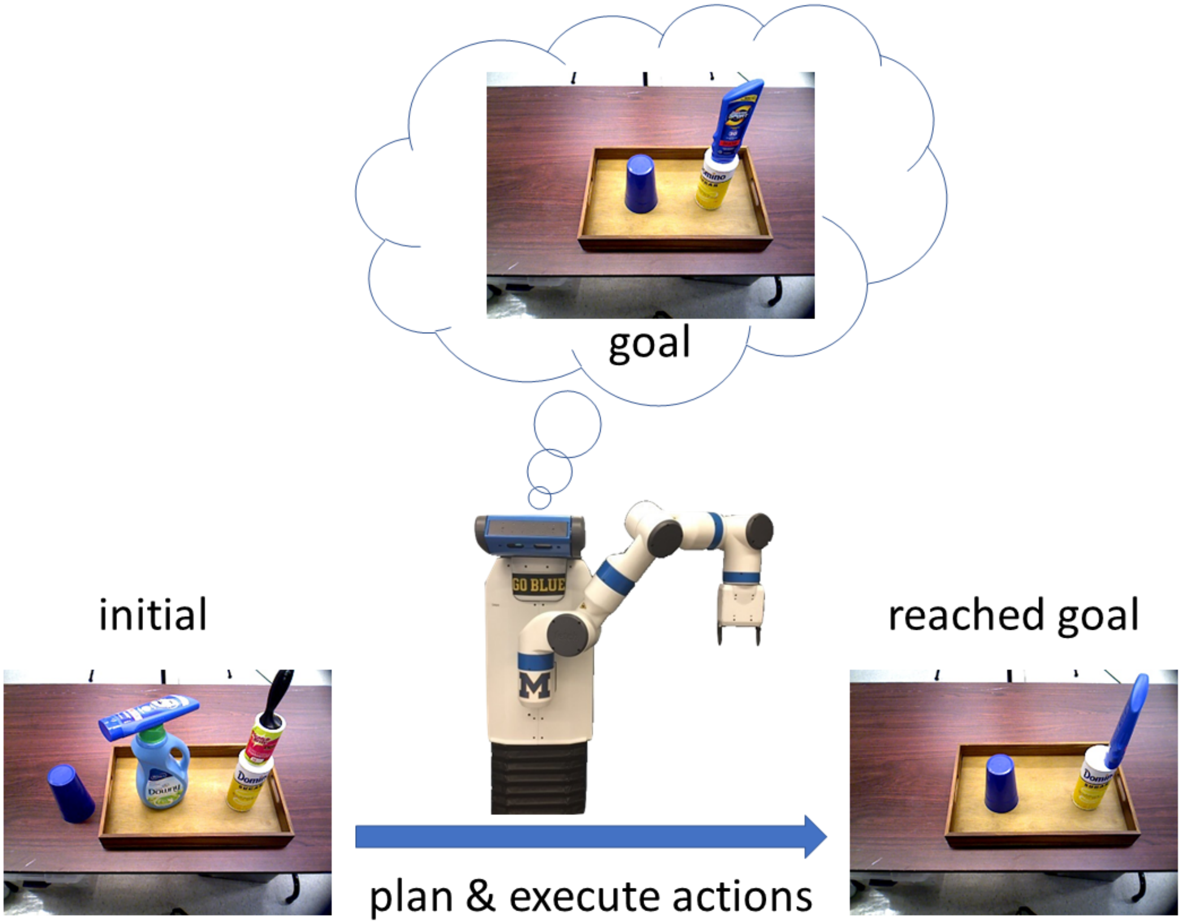


Figure 3.1: A robot preparing a tray through goal-directed manipulations. Given the observation of the user desired goal state and the initial state of the tabletop workspace, the robot first perceives the axiomatic scene graph of the goal and initial state, and then plan and execute goal-directed actions to prepare the tray the way the user desires.

into a axiomatic scene graph for robotic goal-directed manipulation. Building on object pose estimation work proposed by Sui et al. [158], we ground scene observations into axiomatic scene graphs, where objects and inter-object relations are grounded with symbols.

The paradigm of Semantic Robot Programming for robot manipulators with a complementary method for more tractable scene perception. *SRP* is a declarative approach to programming robots through demonstration, where users only need to demonstrate their desired state of the world. *SRP* is general across methods of perception, given the perceived scene is represented axiomatically. For scene perception, we present the Discriminatively-Informed Generative Estimation of Scenes and Transforms (*DIGEST*) method to infer the initial and goal scene states for *SRP* from RGBD images. *DIGEST* brings together discriminative object detection and generative pose estimation for inference of 6 DOF object poses in cluttered scenes, assuming the number of objects is known. Given perceived initial and goal scenes, the robot can plan and execute goal-directed manipulation to autonomously transit the world from the initial to the goal state.

We evaluate the *SRP* paradigm in tray-setting task scenario with the Michigan Progress Fetch robot (Figure 3.1). We benchmark the performance of *DIGEST* on a household occlusion dataset [6] and our cluttered scene dataset. We demonstrate that *SRP* is effective in understanding the goal of a task given a demonstrated snapshot of the goal scene. And, the robot is able to plan and execute goal-directed manipulation actions to reach the goal from various initial states of the world. We additionally found *DIGEST* performs favorably in comparison with state-of-the-art methods for scene perception, such as D2P [120], with fewer assumptions of prior knowledge.

3.2 Related Work

SRP builds on much existing work in robot Programming by Demonstration (PbD) and scene perception for manipulation. Similar to robot PbD, *SRP* aims to enable users to effectively communicate their objectives to robots for performing manipulation tasks. We posit advances in scene perception for manipulation offers new avenues for extending the ease and intuitiveness of robot PbD.

3.2.1 Programming by Demonstration

To improve communication of tasks from a user to a service robot, existing research has focused on learning low-level skills from users. Different approaches have been proposed in Programming by Demonstration (PbD) for low-level learning of skills, such as trajectories [118] [4] and control policy [29] [55] in robot *configuration space*. These methods are inherently limited to world states in *workspace* that are similar to the ones in the demonstrations. By representing the goal of a task in the *workspace* instead of in the *configuration space*, goal-directed manipulation can reason and plan its actions to reach the goal from arbitrary initial world states.

Other work has focused on the high-level aspects of a task. Veeraraghavan et al. [172] propose learning high level action plan for a repetitive ball collection task from demonstrations. Ekvall et al. [43] focus on learning task goals and use a task planner to reach the goal. Chao et al. [28] provide an interface for the user to teach task goals in a tabletop workspace. However, these methods wind up simplifying the scene perception problem by using planar objects, box-like objects or objects with distinguishing colors, that are far from real world scenarios. Recently, Yang et al. [184] have proposed learning action plans in real world scenario, similar to our robot programming paradigm that works with real world objects.

3.2.2 Scene Perception for Manipulation

Being able to perceive objects in real world scenarios and act on them remains a challenge. Some works are able to extract grasping point [31, 94, 164] in point cloud data, however, their methods do not provide a structural understanding of the scene, failing to support goal-directed manipulation on objects.

Although not directly targeted at scene perception for manipulation, work on object pose estimation are highly related to our work. Feature-based object pose estimation methods suh as spin images [69], FPFH [139], OUR-CVFH [7] and VFH [140], rely on feature matching between the object model and observation, however, the problem is that the performance of feature-based methods degrades as the environment becomes more cluttered and key features are occluded. To deal with occlusion, Zhang et al. [191] formulated a physics informed particle filter for grasp acquisition in planar scenes. Narayanan et al. proposed a generative approach named D2P [120], which outperforms feature-based method OUR-CVFH on the household occlusion dataset [6]. D2P renders multiple scene hypotheses, and use A* to search for the hypothesis that best explains the observation. Similarly, Sui et al. [159] proposed a generative approach for object pose estimation in clutter. Their method used discriminative method such as neural network based object detectors to narrow down the search space in the generative approach. We have adopted the similar strategy in our proposed scene estimation method *DIGEST*. In our experiments, we demonstrate that *DIGEST* outperforms D2P on the household occlusion dataset.

To plan goal-directed manipulations, knowing the object poses is not sufficient, however. The robot must have a structural understanding of the scene, that is, the inter-object spatial relations. Given observations of the scene, our work estimates a scene graph that represent the scene structure. Liu et al. [101] also estimate a scene graph given observations, however, their approach approximates objects as oriented bounding boxes. Sui et al. proposed a generative approach [158] for scene graph estimation and use Markov Chain Monte Carlo to search for the best scene graph hypothesis that explains the observations.

Both works by Narayanan et al. [120] and Sui et al. [158] assume that the robot knows what objects are present in the scene, and objects are standing in their upright poses, thus both methods can only estimate 3 DOF poses of objects (i.e., x, y, θ). However, these assumptions are too strong in real world scenarios. Instead, our scene estimation method *DIGEST* does not rely on any of these assumptions, and it can estimate 6 DOF poses of objects, as long as the number of objects in the scene is known.

3.3 Problem Statement

SRP with *DIGEST* assumes the number of objects N_c present in the scene, 3D mesh models $\mathrm{M} = \{m_1, \cdots, m_l\}$ for a set of objects. The robot is assumed capable of performing a set of manipulation actions $\mathbf{A} = \{a_1, \cdots, a_n\}$ with known pre-conditions and post-conditions on these objects. We assume as given RGB-D observation of the goal scene o_G specified by the user at time t, and the current scene o_I at a later time $t + T$. The objective of *SRP* is to plan a sequence of goal-directed manipulation actions $\{a_i, \cdots, a_j\}$ to rearrange objects in the world such that the inter-object relations in s_G are satisfied; where *DIGEST* infers the goal scene graph s_G and the initial scene graph s_I, respectively.

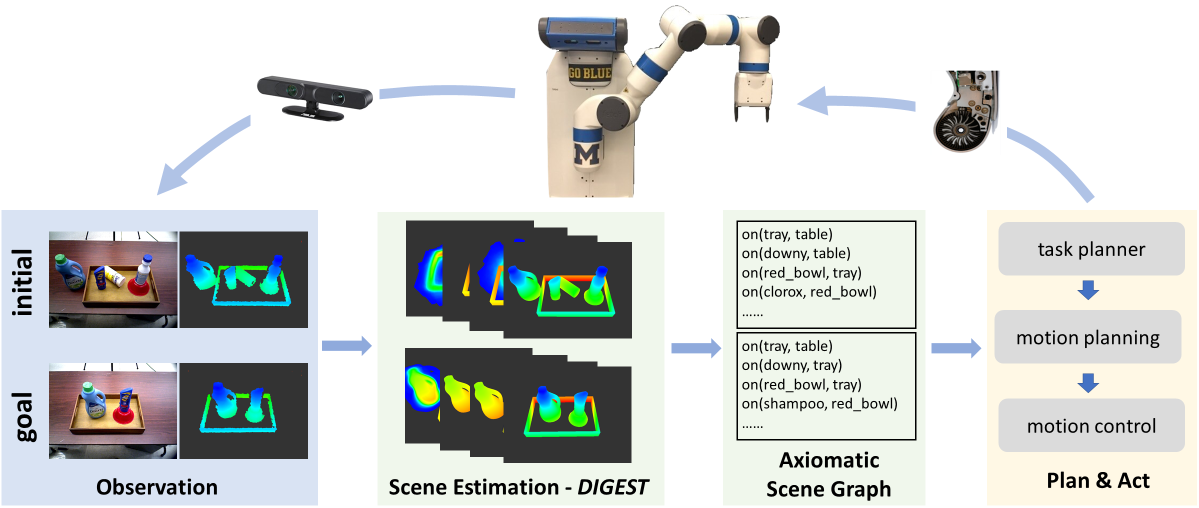


Figure 3.2: Our goal-directed robot programming has three stages: 1) Given the RGB-D observation of the goal and initial scene, we use the proposed scene estimation method *DIGEST* to detect object and estimate the 6 DOF pose of objects; 2) Axiomatic scene graphs can be derived from the estimated object poses, which express the inter-object spatial relations; 3) By describing the goal and initial scene graph by PDDL, the robot uses a task planner (e.g., STRIPS) to plan and execute a sequence of goal-directed actions to reorganize the objects in the scene, reaching the same inter-object relations in the goal scene graph.

We use a list of axiomatic assertions to describe a scene as a scene graph. The scene state at time t is expressed as a scene graph $s_t = \{v^i(x_t)\}_{i=0}^{K}$, where $v^i \in \{exist, clear, on, in\}$ is an axiomatic assertion parameterized by $x_t = \{q_t^j\}_{j=0}^{N_c}$, with q_t^j denoting the pose of jth object at time t, N_c being the number of objects, and K being the total number of axiomatic assertions. In our work, the assertions are limited to spatial relations that can be tested geometrically. The 6 DOF pose $q_t^j = [x_t^j, y_t^j, z_t^j, \phi_t^j, \psi_t^j, \theta_t^j]$ of each object is estimated, consisting 3D position (x_t^j, y_t^j, z_t^j) and orientation $(\phi_t^j, \psi_t^j, \theta_t^j)$. The scene graph can be inferred from the estimated object poses, as explained later in Section 3.4.2.

3.4 Methods

The *SRP* paradigm consists of the perception of goal and initial scene states, and the planning and execution stages, as shown in Figure 3.2. Given observations of a cluttered scene, the generative sampling inference process over object poses is informed by detections from a discriminative object detector. A scene graph encoding inter-object relations is geometrically inferred from an estimate of inferred object poses. The resulting scene graph is then expressed axiomatically for use in task planning and execution.

3.4.1 DIGEST Cluttered Scene Estimation

Given observed RGB-D image pair of a cluttered scene at time t, the objective is to estimate the object poses $q_t^j, j = 1, \cdots, N_c$. We utilize the discriminative power of a pre-trained object detector to first obtain a set of bounding boxes with object labels. These bounding boxes are used to guide the generative process of scene hypotheses sampling. An overview of the cluttered scene estimation is as illustrated in Figure 3.3.

3.4.1.1 Object Detection and Scene Hypotheses Generation

Given an RGB image, m bounding boxes are detected by the object detector. We use B_i ($0 \leq i \leq m$) to denote the bounding box. In the output of the object detector, each B_i is associated with a list of object detection confidence $v(L_j|B_i)$, where L_j is the object class. For each B_i, we generate an object candidate C_i,

$$C_i = \{\arg\max_{L_j} v(L_j|B_i),\ B_i\} \tag{3.1}$$

which is a set including the object label with the highest confidence measure and the associated bounding box. For m generated candidates, the number of scene hypotheses h equals to N_c chooses m, i.e.,

$$h = \begin{cases} \binom{m}{N_c} & \text{if } N_c \leq m \\ 1, & \text{otherwise} \end{cases} \tag{3.2}$$

Thus, if the number of candidates is greater or equal to the number of objects in the scene, each scene hypothesis H_i contains a combination of N_c candidates selected from m candidates. If the number of candidates is less than N_c, just one scene hypothesis with m candidates will be generated.

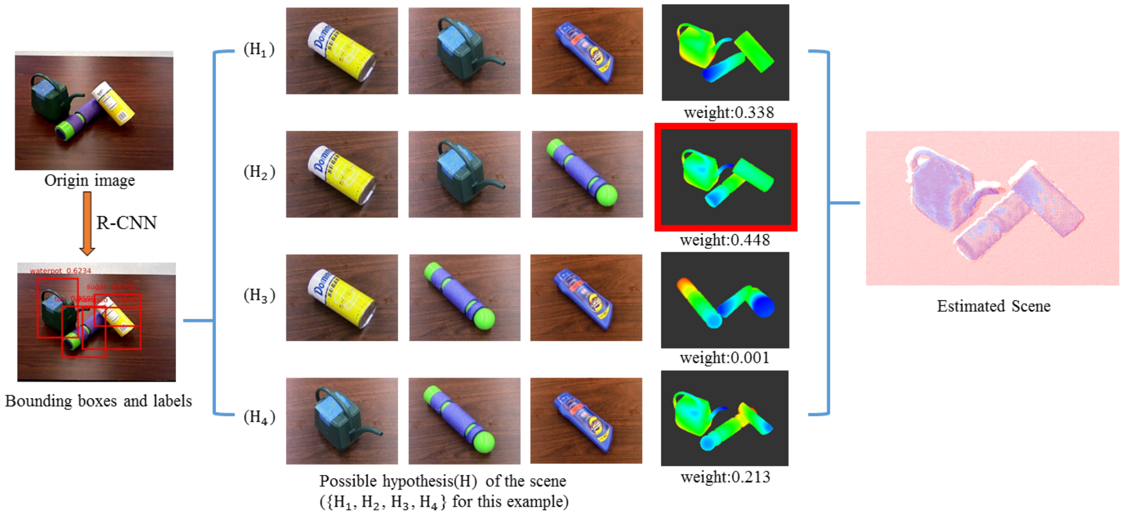


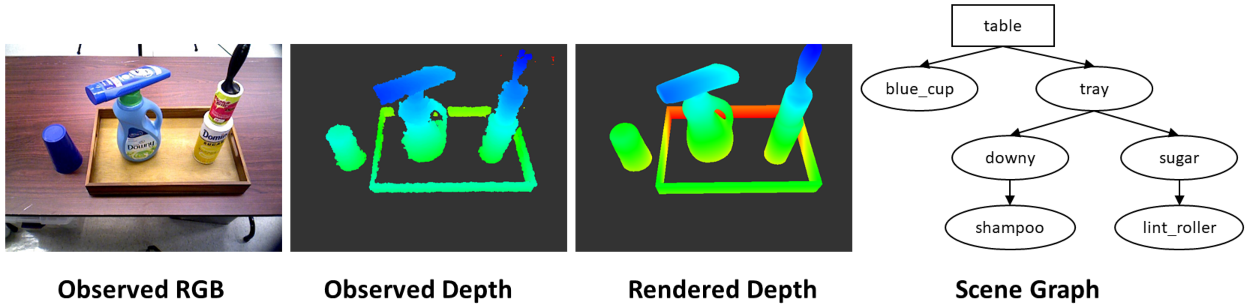
Figure 3.3: The proposed *DIGEST* method for cluttered scene estimation. First, the observed RGB image is passed through a R-CNN object detector trained on our grocery object dataset. The R-CNN object detector outputs a set of bounding boxes, with associated object label and detection confidence. Knowing the number of object present in the scene, possible scene hypotheses are enumerated, e.g., ${}^4C_3 = 4$ scene hypotheses are generated in this example. For each scene hypothesis, particle filtering is applied to estimate object poses that best explains the observed depth. After convergence, *DIGEST* outputs the estimated object poses for the most likely scene hypothesis.

3.4.1.2 Bootstrap Filtering for Pose Estimation

Each scene hypothesis H_i is modeled as a random state variable x_t, composed of a set of real-valued object poses. Object poses are assumed to be statistically independent. We model the inference of the state from robot observation as a Bayesian filter problem. Compared to traditional Bayesian filter problems, we have only one observation: a snapshot of the scene instead of a history of observations. Thus, we apply Iterated Likelihood Weighting [111] to bootstrap the scene estimation process, where $z_1 = z_2 = \cdots = z_t$ and the state transition in the action model is replaced by a zero-mean Gaussian noise. We approximate the belief distribution by a collection of N particles $\{x_t^{(j)}$ weighted by $w_t^{(j)}\}_{j=1}^N$,

$$p(x_t|z_{1:t}) \propto p(z_t|x_t) \sum_j w_{t-1}^{(j)} p(x_t|x_{t-1}^{(j)}, u_{t-1}) \tag{3.3}$$

$$x_t^{(j)} \sim \sum_j w_{t-1}^{(i)} p(x_t|x_{t-1}^{(i)}, u_{t-1}) \tag{3.4}$$



Observed RGB   Observed Depth   Rendered Depth   Scene Graph

Figure 3.4: An axiomatic scene graph example. In the scene graph derived from the estimated object poses, each node corresponds to an object, and each edge indicates the supporting relation between objects. *table* is by default the root node.

as described by [38]. To evaluate the weight $w_t^{(j)}$ for particle $x_t^{(j)}$, we render a depth image based on the object poses in $x_t^{(j)}$, and compare it against the observed depth image $\hat{z}_t^{(j)}$,

$$w_t^{(j)} = e^{-\lambda_r \cdot \mathrm{d}(z, \hat{r}_t^{(j)})} \tag{3.5}$$

where λ_r is a constant scaling factor. $\mathrm{d}(z, \hat{r}_t^{(j)})$ is the sum of the Euclidean distance between the 3D points projected back from depth images z, $\hat{r}_t^{(j)}$, using the intrinsic parameters of the camera. Pose estimation is performed over successive iterations that: 1) compute the weight of each particle, 2) normalize the weights to one, 3) draw N particles by importance sampling, and 4) diffuse each sampled particle by a zero-mean Gaussian noise. After maximum number of iterations, the most likely particle as the scene estimate for scene hypothesis H_i:

$$x_t = \arg\max_{x_t^{(j)}} p(x_t^{(j)} | z_{1:t}) \tag{3.6}$$

3.4.1.3 Final Scene Ranking

After particle filtering for all scene hypotheses, we have a scene estimate x_t for each scene hypothesis. We then rank them based on the likelihood of each x_t as computed earlier. The most likely x_t is taken as the scene estimate and is then used to derive the scene graph.

3.4.2 Scene Graph Structure

The objects pose estimation of a cluttered scene can be turned into an axiomatic scene graph. We use following axiomatic assertions: $exist(q^j)$ for the assertion that object j exists in the scene with pose q^j; $clear(q^i)$ for the assertion that the top of object i is clear and no other objects are stacked on it; $on(q^i, q^j)$ for the assertion that object i is stacked on object j; $in(q^i, q^j)$ for the assertion that object i is in object j. An example of a scene graph is given in Figure 3.4.

To assert the proximity relations between two objects i, j, we add a *virtual object* q^γ with geometry m^γ into the scene graph, with m^γ being a shape that can be arbitrarily defined based on the application, and q^γ being the identity pose in the frame of object i. Then, the proximity relation between objects i, j can be encoded by $\{has(q^i, q^\gamma),\ in(q^\gamma, q^j)\}$, where $has(q^i, q^\gamma)$ asserts that object i has a *virtual object* q^γ attached to its frame. When the parent object i is in a new location, the robot can adapt to the new scenario by placing the child object j within the region of m^γ attached to the frame of i.

To determine the stacking relations between the objects, we use simple heuristics. In the 3D mesh object models, the z-axis of each object is the gravitational axis when the object stands upright. The dimensions $\{h_x, h_y, h_z\}$ of the 3D box that encloses each object model are given as prior knowledge. In order to determine whether object i is being supported by another object, two heuristics are tested: (1) if one of the object axes (e.g., x-axis) is aligned with the gravitational axis, then the height h_i of the 3D volume occupied by the object equals to the corresponding dimension (e.g. h_x) of the provided 3D enclosing box. A simple rule $z^i - h^{table} > 0.5h^i$ is used to determine whether object i is being supported by another object; (2) if none of the object axes are aligned with the gravitational axis, then object i is being supported by another object.

The set of objects that is being supported by other objects is sorted with increasing z values of the object pose, and is denoted as O_s, the remaining objects are denoted as O_r. For each object $i \in O_\mathrm{s}$, a heuristic measure is used to determine which object $j \in O_\mathrm{r}$ is supporting i,

$$\arg\max_j f(r_b(q^i), r_t(q^j))$$

where $f(r_1, r_2)$ measures the overlapping area of two regions r_1, r_2, and $r_t(q^i), r_b(q^i)$ represent the projected region on the table of the top and bottom surface of object i, respectively. Once the supporting object for $i \in O_\mathrm{s}$ is identified, i is moved from set O_s to O_r. With the supporting relation between a pair of objects i, j identified, the corresponding axiomatic assertion is expressed as either $on(q^i, q^j)$ or $in(q^i, q^j)$, depending on the geometry type of the supporting object j being convex or concave.

We can extend the current set of axiomatic assertions to a more extensive set that describes the inter-object spatial relations in the scene graph in more detail, such as *in front of*, *to the left of*, *to the right of*, and *peg-in-hole* relations, and even *wrap around* with deformable object, etc. Depending on the domain of the task, the frame problem [107] can arise such that it can be tricky to find adequate collections of axioms for viable description of the robot task domain. In the scope of this dissertation, we focus on the robot task domain of rigid object organization tasks, i.e., putting rigid objects into satisfying spatial relations relative to other objects, and we limit the inter-object spatial relations of interest to *on*, *in* and *proximity*.

3.5 Implementation

3.5.1 RCNN object detector

We employ R-CNN [54] as our discriminative object detector as described in section 3.4.1.1. R-CNN first generates object bounding boxes given an image, then for each bounding box, it outputs the confidence measure through a deep convolutional neural network. For the sake of efficiency and performance, we replace the original selective search [170] with EdgeBox [193] for object proposal generation. We train an R-CNN object detector on our object dataset that includes 15 grocery objects. The dataset contains 8366 ground truth images (~557 average ground truth images for one object) and 60563 background images. We fine tuned our object detector on a pre-trained model on ImageNet [39].

3.5.2 Particle filtering and parallelization

During bootstrap filtering for pose estimation as described in section 3.4.1.2. Each object in each particle $x_t^{(j)}$ is initialized by candidate C_i in the scene hypothesis, the object label l_i determines which 3D mesh object model to use, and the initial pose is uniformly sampled inside the bounding box B_i. A parallel graphics engine rapidly renders depth images given all particles. CUDA is used to compute the weights of all particles in parallel. Through our experiment, we fix particle filter iteration to 400 and use 625 particles.

In the particle filtering process, the pose of each object is estimated sequentially. For example, if there are four hypothesized objects and 400 particle filter iterations, the pose of the object with the maximum detection confidence is estimated in the first 100 iterations. Then the pose of the object with the 2nd largest detection confidence is estimated in the next 100 iterations, with the first object fixed at the most likely pose. We carry on the estimation process iteratively for the remaining objects.

We ran our experiments on a computer with i7 2.60GHz CPU and Nvidia GeForce GTX 980M. It takes on average 30 seconds to finish 400 particle filter iterations for each scene hypothesis in our robot experiments, and the bottleneck of computation time is the likelihood calculation of each particle as discussed in Equation 3.5, and the rendering of depth images based on hypothesized object classes and poses represented in each particle. A detailed breakdown of the runtime of a typical particle filtering iteration is as shown in Figure 3.5.

The complexity of the scene estimation is $\mathcal{O}(NK \min{(m^n, m^{m-n})})$, where N is the number of particles, and K is the number of particle filtering iterations. $\mathcal{O}(\min{(m^n, m^{m-n})})$ is the complexity of $\binom{m}{n}$ that corresponds to the number of combinatorial scene hypotheses, where m is the number of detections in the observed scene, and n is known number of objects in the scene. As m gets much larger than to n, the number of scene hypotheses gets large, techniques such as Markov Chain Monte Carlo can be used to efficiently sample scene hypotheses rather than a brute force search over the complete set of scene hypotheses.

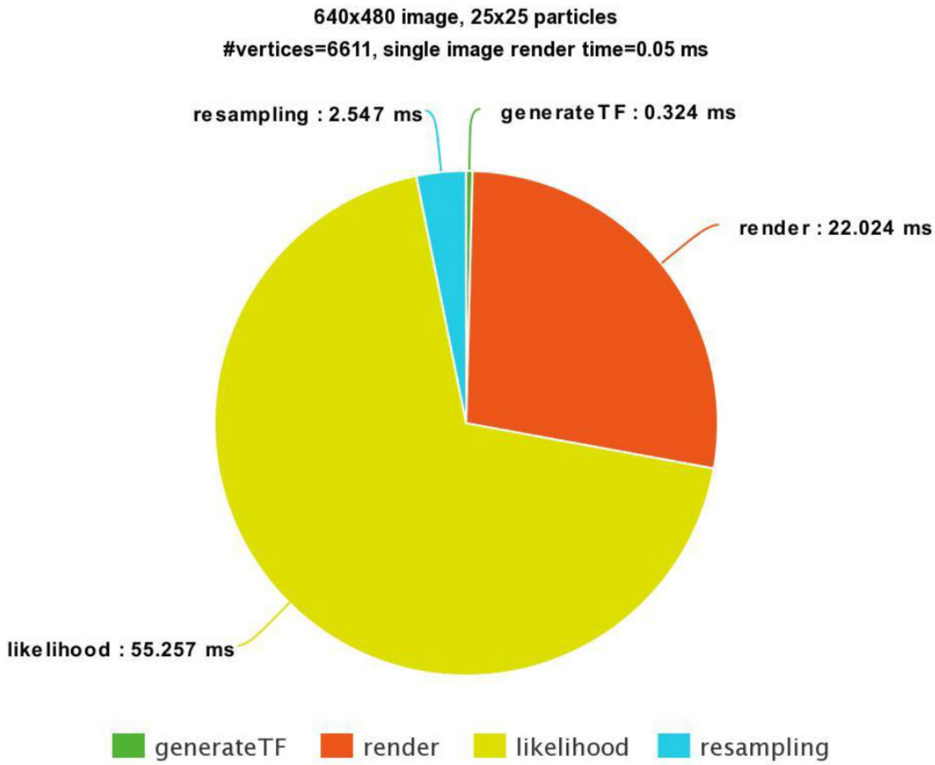


Figure 3.5: Runtime breakdown of a typical particle filtering iteration. *Generate TF*: time that takes to prepare the transformation (TF) of object meshmodels for rendering; *render*: time that takes to render the objects (there are in total 6611 vertices in objects mesh models in this particular example). The rendering time increases approximately linearly as the total number of vertices to be rendered increases; *likelihood*: time that takes to compute the likelihood; *resampling*: time that takes to resample particles.

3.5.3 Planning and Execution

Given the observation of the goal state of the world, the robot estimates the goal scene graph, and stores the desired inter-object relations by PDDL [110]. Similarly, the robot estimates and stores the initial inter-object relations by PDDL. With sets of PDDL that describe the initial and goal state, the robot uses a task planner to plan a series of goal-directed actions to rearrange objects in the initial scene, such that the same inter-object relations in the goal scene graph are satisfied. We use breadth first search STRIPS[47] as our task planner. Note that the robot does not need to rearrange the objects with the exact same poses as in the goal scene, as long as the same inter-object relations are achieved, similarly to how human would arrange a set of daily objects based on simple instructions.

The task planner gives a sequence of high-level pick-and-place actions. To pick an object, the robot is given a set of pre-computed grasp poses of the object using [164], and uses Moveit! [155] to check which grasp pose it can generate a collision-free trajectory for, and use that for grasping. To place an object, the robot sample place poses in the empty space that satisfies the desired inter-object relations, and again use the place pose it can generate a collision-free trajectory for.

3.6 Experiments

In our experiments, we first evaluate our scene estimation method on a public household occlusion dataset and our cluttered scene dataset, and then evaluate our overall semantic robot programming paradigm in tray setting tasks. *DIGEST* outperforms the state-of-the-art method D2P on the household occlusion dataset, and outperforms FPFH on our cluttered scene dataset. We demonstrate the effectiveness of our system for programming a robot to complete various tray-setting tasks through goal-directed manipulations. We run all experiments on a computer with an Titan X Graphics card and CUDA 7.5.

3.6.1 DIGEST: Cluttered Scene Estimation

To evaluate *DIGEST* on pose estimation, we benchmarked the performance of *DIGEST* on two different datasets: household occlusion dataset [6], and our cluttered scene dataset. The household occlusion dataset contains objects standing up right, thus it only affords benchmarking on 3 DOF object pose estimation. In our cluttered scene dataset, objects can be in arbitrary pose, and we use it for benchmarking on 6 DOF object pose estimation. Object pose estimation accuracy is calculated as the percentage of correctly localized objects over the total number of objects in the dataset. An object is correctly localized if the pose error falls within certain position error threshold Δt and rotation error threshold $\Delta\theta$. The position error is the Euclidean distance error in translation; the rotation error is the absolute angle error in orientation. For rotationally symmetric objects, the

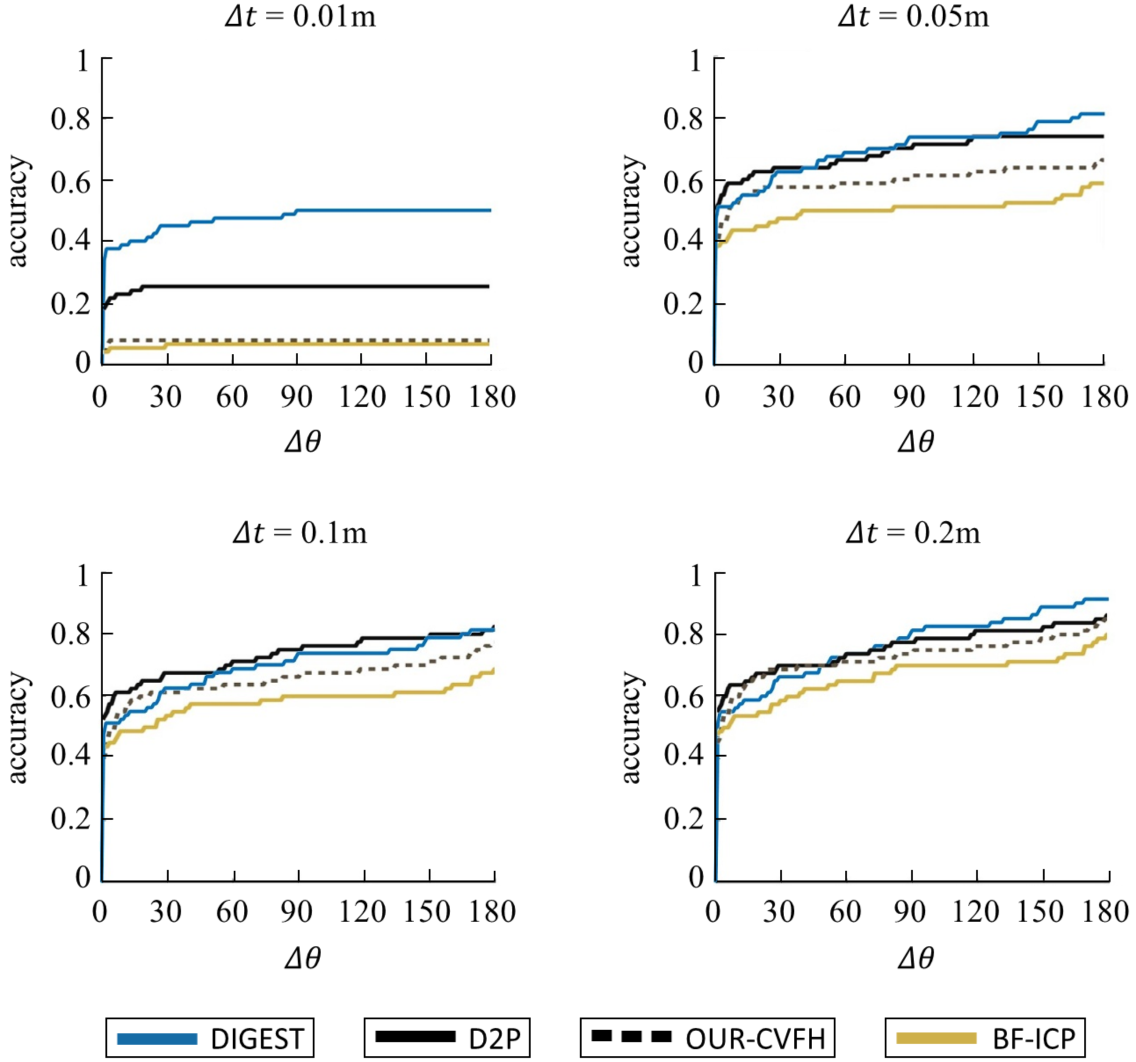


Figure 3.6: Object pose estimation benchmark of *DIGEST* on public household object dataset [6], compared with three baseline methods: D2P, OUR-CVFH and BF-ICP for different correctness criteria Δt, $\Delta\theta$. *DIGEST* outperforms D2P for strict correctness criteria, and performs on par with D2P for relaxed correctness criteria.

rotation error about the symmetric axis is ignored.

3.6.1.1 Household Occlusion Dataset – 3 DOF Object Poses

The household occlusion dataset contains 22 test scenes with 80 objects in total. The test scenes include objects such as milk bottles, laundry items, mugs and etc; We compare *DIGEST* against three baseline methods as described in [120], that is, D2P, OUR-CVFH [7], and Brute Force

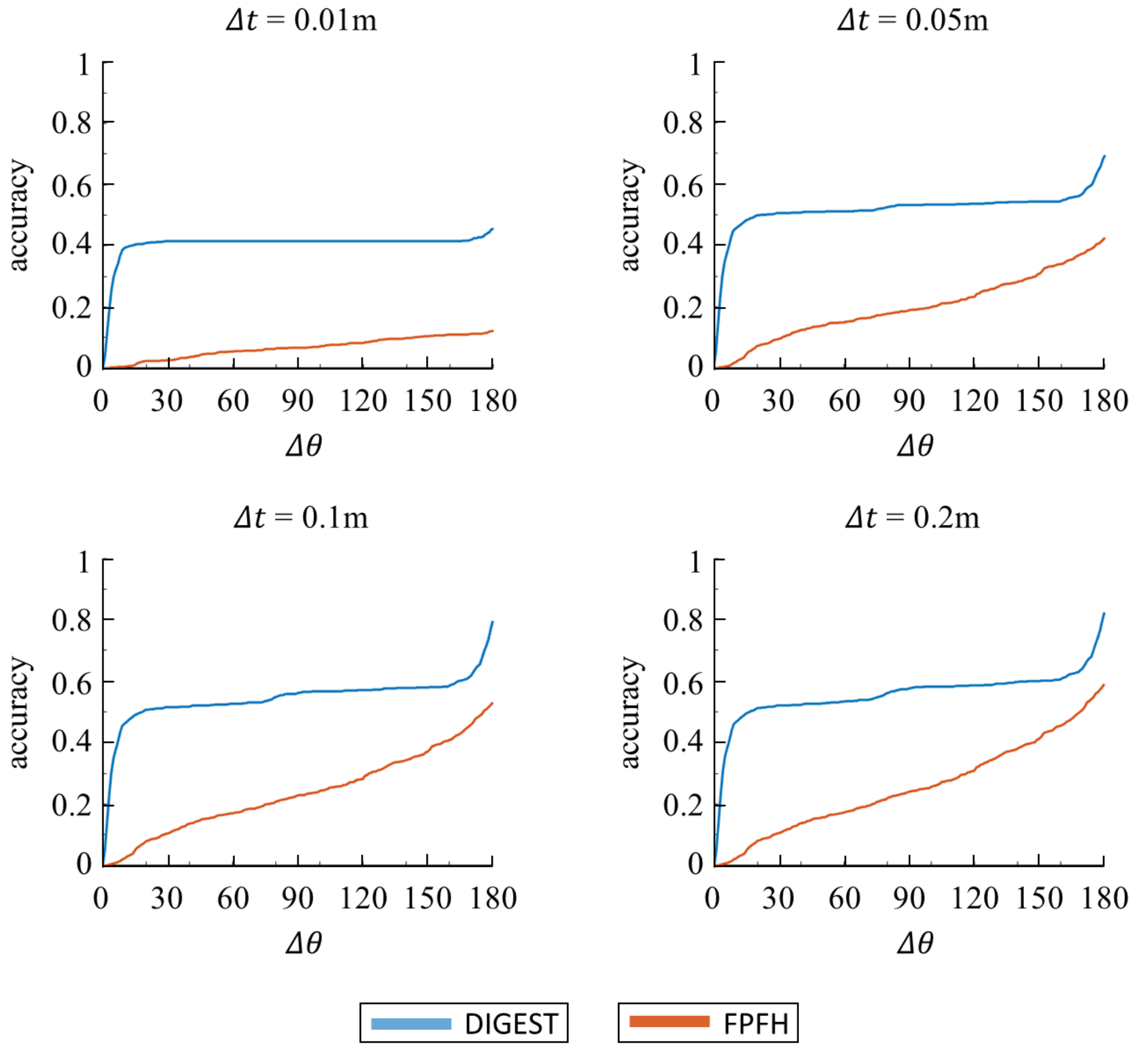


Figure 3.7: Object pose estimation benchmark of *DIGEST* on our cluttered scene dataset, compared with baseline method FPFH under different correctness criteria Δt, $\Delta\theta$. *DIGEST* outperforms FPFH with large margin.

ICP (BF-ICP). D2P also uses an R-CNN object detector as part of their pose estimation process. However, it is not clear what hyper parameters they choose during the training phase of the object detector. In order to avoid bias in the training of the object detector, we use their object detector on the household occlusion dataset.

When only little error is allowed for an estimated pose to be counted as correct, as shown in the left upper plot in Figure 3.6, the accuracy of *DIGEST* is nearly twice the accuracy of D2P. As we relax the tolerance on the pose estimation error, as shown in the other three plots in Figure 3.6,

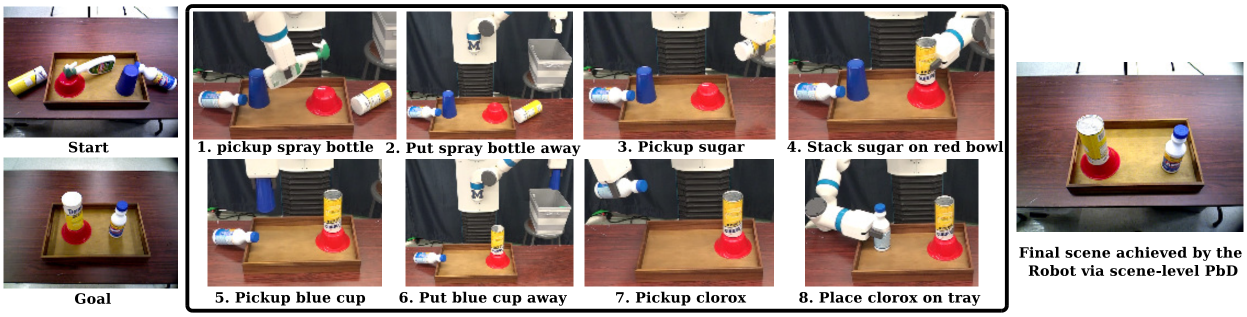


Figure 3.8: Our robot performing goal-directed manipulation (middle columns) to prepare a tray (right) satisfying the user-demonstrated goal (left bottom).

DIGEST performs on par with D2P. Overall, *DIGEST* outperforms D2P since (1) *DIGEST* explores the state space a lot more than D2P, as we do not discretize the state space, and (2) *DIGEST* does not use ICP for local search, which D2P employs for pose estimation. In terms of run time, *DIGEST* takes around 30 seconds (varying with the number of objects and the size of object mesh), which is faster than 139.74 seconds reported in D2P.

3.6.1.2 Cluttered Scene Dataset – 6 DOF Object Poses

We collect a cluttered scene dataset with 16 different sceness, and 72 objects in total. This dataset includes laundry, kitchen and toy items. The number of objects in each scene ranges from 3 to 7. This dataset is much more challenging than the household object dataset, as the objects can have random 6 DOF poses. We compare the performance of *DIGEST* with FPFH [139], as shown in Figure 3.7.

3.6.2 Semantic Robot Programming: Tray Setting

We designed our experiments around a service robot scenario, as illustrated in Figure 3.1. The robot needs to prepare a tray as specified by the user int the goal scene. We tested our system on scenes of 4 to 6 objects including the tray, with different inter-object relations, such as stacking and proximity relations. The robot is able to perceive the initial and goal state, then plan and execute goal-directed actions to satisfy the inter-object relations in the goal scene graph.

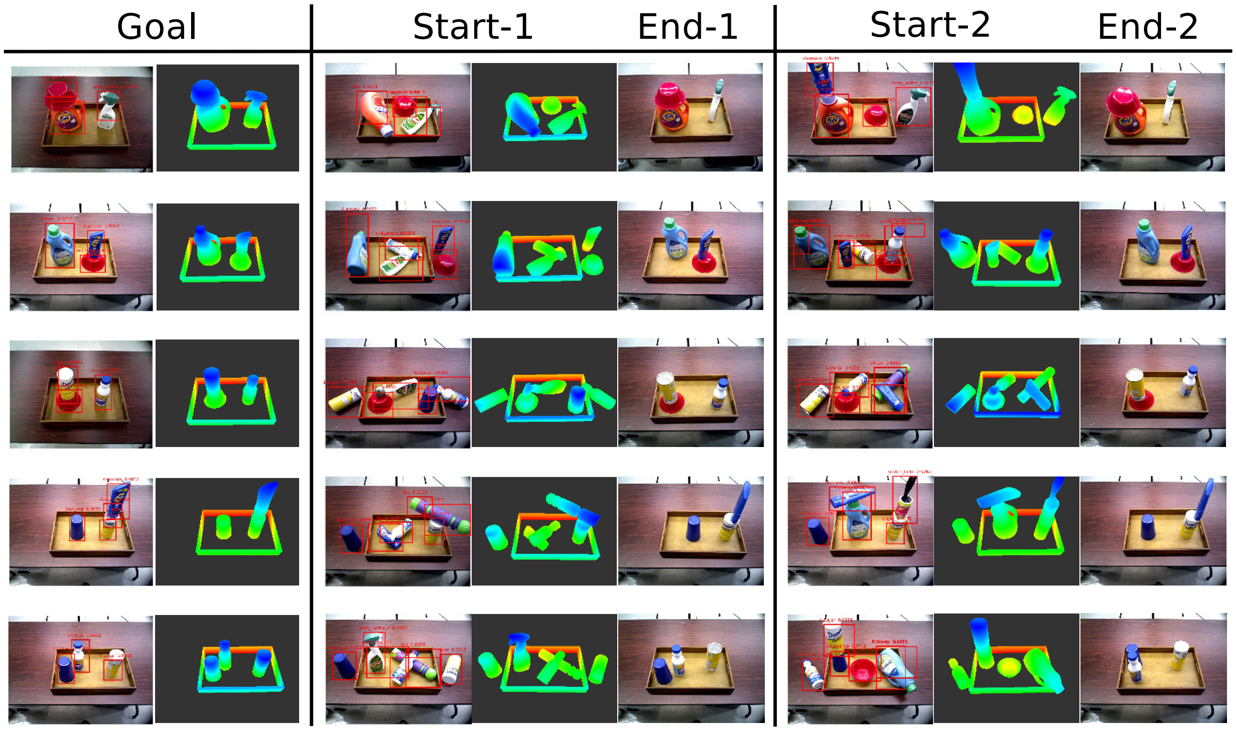


Figure 3.9: Our *SRP* system tested for 5 different tray preparation tasks. The left column shows the goal scene. For each goal, the robot starts from two different initial states and successfully performs goal-directed manipulations to prepare the tray. Depth images are rendered based on 6 DOF object poses output by *DIGEST*.

An example of *SRP* for goal-directed manipulation is shown in Figure 3.8. Based on the scene graph inferred from the object pose estimates, the robot generates a sequence of goal-directed actions to achieve the goal state. Our tray setting experiments are shown in Figure 3.9, and more detail in this video[1]. The goal and start scenes are well estimated as a collection of 6DOF poses of objects. The robot successfully sets up a tray as the user desired in 10 out of 10 different tray setting experiments.

3.7 Conclusion

We have presented Semantic Robot Programming as a paradigm for users to easily program robots in a declarative goal-directed manner. We demonstrate the effectiveness of *SRP* using the pro-

[1] https://youtu.be/ZJLD_6v88KA

posed *DIGEST* scene perception method on two datasets of objects in occlusion and clutter: both house occlusion dataset and our cluttered scene dataset. Through our approach to generative-discriminative perception, *SRP* with *DIGEST* is able to perceive, reason, and act to realize an arbitrary user-demonstrated goal in cluttered scenes.

SRP provides many interesting directions to pursue, such as motion planning over sequences of general manipulation actions. Currently, grasp point localization [164] is used to select good grasp poses for object picking. However, such selected grasp poses are not necessarily appropriate for a later placement actions. Visual inspection on selected grasps is done before robot execution. Ideally, appropriate grasp poses would be provided by a manipulation affordance mechanism, such as Affordance Templates [58] associating robot action with an object. Such affordance mechanisms would allow for investigation of more flexible task and motion planning over sequences of actions. We further posit scene perception can be made to run in interactive-time through a thoughtful parallelized implementation, enabling potentially interactive planning and manipulation execution.

CHAPTER 4

CT-Map: Contextual Temporal Semantic Mapping

In the previous chapter, an *SRP* framework was described for tabletop tasks on a manipulator robot. In many tasks that a user would desire, a larger workspace beyond the tabletop can be involved. For example, a user would like to program the robot to organize the living room rather than a single tabletop. Due to the limited field of view of robot sensors, neither the demonstrated goal scene for organizing the living room nor the current world state can be captured with a single observation. Instead, the robot should infer either the goal or the current world state from multiple observations across the scene. Thus, to scale *SRP* from tabletop scale to larger scale (e.g., room level), we would need to scale up the scene perception system, and tabletop manipulation to mobile manipulation actions. As a step towards *SRP* at large scale, this chapter discusses a semantic mapping approach for scaling up the scene perception system, where objects are simultaneously detected and localized given streaming observations across a scene. The issues for scaling up robot actions towards mobile manipulation actions are later described in Chapter 5.

Given streaming observations from the robot perception sensor (e.g. RGB-D camera), we propose to semantically map an observed scene with simultaneously detected and localized objects in an on-line fashion. The semantic map consists of a list of objects with their class labels and 6 degree-of-freedom poses. As new observation becomes available, our approach incrementally updates the semantic map, and remains computationally tractable as the number of objects increases. Unlike *DIGEST* (discussed in the previous chapter), where objects are treated independently during the scene estimation process, we now explicitly model the dependencies between objects. Further, we will show how modeling the dependencies between objects can benefit the scene estimation process.

We present a filtering-based method for semantic mapping to simultaneously detect objects and localize their 6 degree-of-freedom pose, called Contextual Temporal Mapping (or *CT-Map*) [189]. We represent the semantic map in *CT-Map* as a belief over object classes and poses across an observed scene. Inference for the semantic mapping problem is then modeled in the form of a Conditional Random Field (CRF). *CT-Map* is a CRF that considers two forms of relationship potentials to account for contextual relations between objects and temporal persistence of object poses, as well as a measurement potential on observations. A particle filtering algorithm is then proposed to perform inference in the *CT-Map* model. We demonstrate the efficacy of the *CT-Map* method with a Michigan Progress Fetch robot equipped with a RGB-D sensor. Our results demonstrate that the particle filtering based inference of *CT-Map* provides improved object detection and pose estimation with respect to baseline methods that treat observations as independent samples of a scene.

4.1 Introduction

For robots to effectively operate and interact with objects, they need to understand not only the metric geometry of their surroundings but also its semantic aspects. When requested to organize a room or search for an object, robots must be able to reason about object locations and plan goal-directed mobile manipulation accordingly. We aim to enable robots to semantically map the world at the object level, where the representation of the world is a belief over object classes and poses. With the recent advances in object detection via neural networks, we have stronger building blocks for semantic mapping. Yet, such object detections are often times noisy in the wild, due to biases and insufficient diversity in training dataset. In our work, we aim to be robust to false detections from such networks. We model the object class as part of our hidden state for generative inference, rather than making hard decisions on class labels as given by the detector.

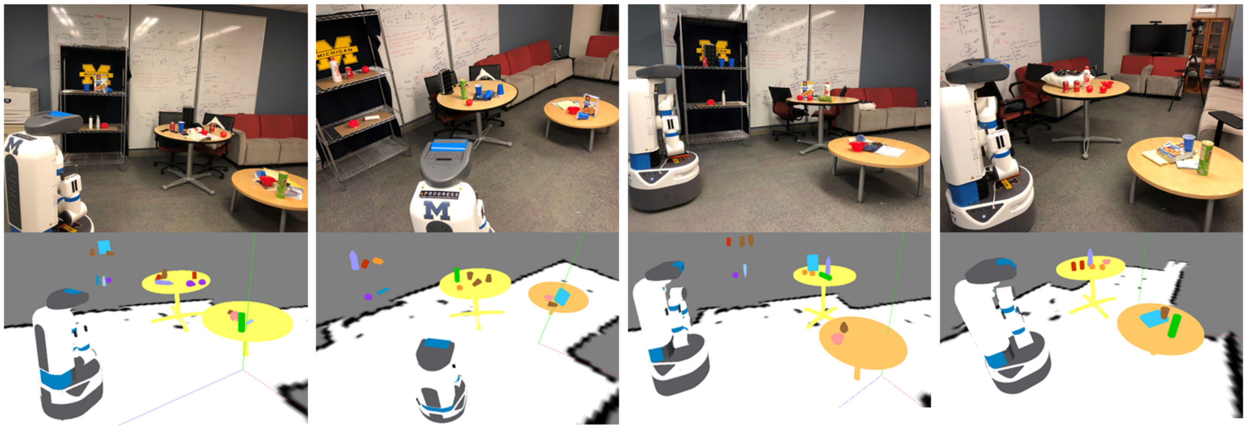


Figure 4.1: Robot semantically maps a student lounge in four different visits. Each column shows an RGB snapshot of the environment, together with the corresponding semantic map composed by the detected and localized objects. We propose Contextual Temporal Mapping (*CT-Map*) method to simultaneously detect objects and localize their 6 DOF pose given streaming RGB-D observations. To achieve this, we probabilistically formulate semantic mapping problem as a problem of belief estimation over object classes and poses. We use Conditional Random Field (CRF) to model contextual relations between objects and temporal consistency of object poses. (Best viewed in color)

Given streaming RGB-D observations, our goal is to infer object classes and poses that explain observations, while accounting for **contextual** relations between objects and **temporal** persistence of object poses. Instead of assuming that every object is independent in the environment, we aim to explicitly model the *object-object* contextual relations during semantic mapping. More specifically, objects from the same category (e.g., food category) are expected to co-occur more often than objects that belong to different categories. Additionally, physical plausibility should be enforced to prevent objects from intersecting with each other, as well as floating in the air.

Temporal persistence of object poses also plays an important role in semantic mapping. We assume smoothness between objects poses across consecutive frames. When objects are not being directly observed, they could stay where they were observed in the past, or gradually change their semantic locations over time. For example, a cereal box that was observed on a table can be moved to a cupboard at a later time. Under cases of occlusion, modeling temporal persistence can potentially help the localization of partially observed objects. Through temporal persistence modeling, the robot could gain a notion of object permanence, i.e., believing that objects continue to exist even when they are not being directly observed.

Considering both contextual and temporal factors in semantic mapping, we propose the **Contextual Temporal Mapping** (***CT-MAP***) method to simultaneously infer object classes and poses. Examples of semantic maps generated by *CT-Map* are shown in Figure 4.1. To avoid deterministically representing the world as a collection of recognized objects with poses, we maintain a belief over the object classes and poses across observations.

For generative inference, *CT-MAP* probabilistically formalizes the semantic mapping problem in the form of a Conditional Random Field (CRF). Dependencies in the CRF model capture the following aspects: 1) compatibility between the latent semantic mapping variables and observations, 2) contextual relations between objects, and 3) temporal persistence of object poses. We propose a particle filtering based algorithm to perform generative inference in *CT-MAP*, inspired by Limketkai et al [97]. Note that the proposed inference algorithm is an instance of approximate nonparametric belief propagation [157].

We evaluate the proposed semantic mapping method *CT-MAP* with the Michigan Progress Fetch robot. The performance of *CT-MAP* is quantitatively evaluated in terms of object detection and pose estimation accuracy. We show that *CT-MAP* is effective in simultaneously detecting and localizing objects in cluttered scenes. We demonstrate object detection performance superior to Faster R-CNN [138], and accurate 6 DOF object pose estimation compared to 3D registration methods such as ICP, and FPFH [139]. We also highlight examples in which our method benefits from modeling temporal persistence of object poses and object contextual relations.

4.2 Related Work

Our work semantically maps the world through simultaneous object detection and 6 DOF object pose estimation. Contextual relations between objects and temporal persistence of object poses are being modeled for better scene understanding. Here we discuss the related works in a) semantic mapping, b) object detection and pose estimation, c) object contextual relations, and d) object temporal dynamics modeling.

Semantic Mapping Considering the plethora of work [84] in the field of semantic mapping which vary in semantic representations, we limit our focus to the works that provide object-level semantics. Works in semantic SLAM [13, 142, 21] demonstrated SLAM at the object level. Similarly, we aim at providing a semantic map of the world at the object level, and we focus on mapping while making use of existing metric slam method (e.g., ORB-SLAM [117]) to stay localized.

A widely used approach for semantic mapping is to augment 3D reconstructed map with objects. Civera et al. [32] ran an object detection thread parallelly with a monocular SLAM thread. They registered objects to the map by aligning the object faces relying on the SURF features. Ekvall et al. [42] actively recognized objects based on SIFT features, and integrated object recognition with SLAM for triangulation of object locations. However, the methods of Civera et al. and Ekvall et al. do not address with false detections, and their experiments were carried out in environments with no clutter.

To be robust to false detections, Pillai et al. [133] proposed aggregating object evidence over multiple frames to get better detection, compared to single frame object detection. However, their method relied on 3D geometric segmentation that singulates objects from the background, which is vulnerable when dealing with clutter. Sünderhauf et al. [160] combined object detection over multiple frames and 3D geometric segmentation to get reasonable object boundaries. They produced 3D reconstructed map with object instance segments as central semantic entities. However, their method did not provide object pose information, which is critical for robotic manipulation tasks.

Other works have focused on scene labeling of 3D map as a parallel SLAM thread is running in the background. Similar research has proposed different methods for single frame scene labeling [192, 154, 108, 180], and fused labels across multiple frames to generate a dense 3D semantic map. Our work focus on detecting and localizing object entities in the environment, instead of dense labeling of every surfel or voxel in the reconstructed 3D map.

Object Detection and Pose Estimation Deep neural network based object detectors [135, 100, 138] are nowadays widely adopted for focusing attention in region of interest given an image. Works in object pose estimation adopt these object detectors to get prior on object locations. Zeng et al. [190] generated scene hypotheses based on object detections returned by R-CNN [54], and they used Bayesian based bootstrap filter to estimate object poses. Similarly, Sui et al. [158] and Narayanan et al. [121] proposed generative approach for object pose estimation given RGB-D observation. Discriminative object pose estimation methods use local [69, 139] or global [140, 7] descriptors to estimate object poses via feature matching. However, feature-based methods are sensitive to the clutterness in the environment. Our work takes the generative approach and builds on Zeng et al. [190] for object pose estimation through Bayesian filtering, while [190] modeled objects independently and took single image at input, we model the contextual dependencies between objects and temporal persistence of each object instance given streaming data.

Works that simultaneously detect and localize objects are highly related to our work. Xiang et al. [181] proposed PoseCNN as a novel network for object detection and 6 DOF object pose estimation given a RGB image. Tremblay et al. [169] and Tekin et al. [163] converted the problem of simultaneous object detection and pose estimation into a problem of detecting the vertices of object bounding cuboid. Unlike these works that take single image as input and outputs deterministic estimate of object poses, our work maintains a belief over object classes and poses across observations.

Given streaming data, Salas-Moreno et al. [142] assumed repeated object instances in the environment to effectively recognize and localize objects. However, their model lacks inter-object dependences. Tateno et al. [162] incrementally segmented 3D surface reconstructed by an underlying SLAM thread, then 3D segments were recognized as objects and object poses were estimated via 3D descriptor matching. Their work is similar to *CT-Map* in terms of the output. However, they depend on 3D geometric segmentation which is not guaranteed to segment objects out in clutter.

In addition, they require dense SLAM with small voxel size which is hard to scale.

Object Contextual Relations Contextual relations play a key role in modeling spatial relations between objects for scene understanding. Koppula et al. [81] showed semantic labeling on point clouds using co-occurrence and geometric relations between objects. Jiang et al. [68] explored indirectly modeling object contextual relations by hallucinating human interactions with the environment. Similarly, [52, 60, 78, 45, 10] have proven modeling *object-object* and *object-place* contextual relations to be useful in place recognition, object detection and object search tasks. In our work, we mainly utilize *object-object* contextual relations in terms of co-occurrence and geometric relations.

Object Temporal Dynamics Modeling We need to maintain the belief over object poses even when objects are not being observed. Different types of the objects share different characteristics of dynamics. For example, structural objects such as furnitures tend to stay approximately at the same location, while small objects such as food items can often be moved from one place to another. Bore et al. [20] proposed to learn long-term object dynamics over multiple visits of the same environment. Toris et. al. [167] proposed a temporal persistence model to predict the probability of an object staying at the location where it is last observed after certain time period. We are inspired by the temporal persistence model proposed in [167], and we reason about the possible locations of an object observed in the past based on the contextual relations between objects.

4.3 Problem Formulation

We focus on semantic mapping at the object level. Our proposed *CT-Map* method maintains a belief over object classes and poses across an observed scene. We assume that the robot stays localized in the environment through an external localization routine (e.g., Beeson et al. [14] and ORB-SLAM [117]). The semantic map is composed by a set of N objects $O = \{o^1, o^2, \cdots, o^N\}$.

Each object $o^i = \{o^c, o^g, o^\psi\}$ contains the object class $o^c \in \mathcal{C}$, object geometry o^g, and object pose o^ψ, where $\mathcal{C}$ is the set of object classes $\mathcal{C} = \{c_1, c_2, \cdots, c_n\}$.

At time t, the robot is localized at x_t. The robot observes $z_t = \{I_t, S_t\}$, where I_t is the observed RGB-D image, and S_t are semantic measurements. The semantic measurements $s_k = \{s_k^s, s_k^b\} \in S_t$ are returned by an object detector (as explained in section 4.5.1), which contains: 1) a object detection score vector s_k^s, with each element in s_k^s denoting the detection confidence of each object class, and 2) a 2D bounding box s_k^b.

We probabilistically formalize the semantic mapping problem in the form of a CRF, as shown in Figure 4.2. Robot pose x_t and observation z_t are known. The set of objects O are unknown variables. We model the contextual dependencies between objects and the temporal persistence of each individual object over time. The posterior probability of the semantic map is expressed as:

$$
\begin{aligned}
&p(O_{0:T}|x_{0:T}, z_{0:T}) = \\
&\frac{1}{Z} \prod_{t=0}^{T} \prod_{i=1}^{N} \phi_p(o_t^i, o_{t-1}^i, u_{t-1}^i) \phi_m(o_t^i, x_t, z_t) \prod_{i,j} \phi_c(o_t^i, o_t^j)
\end{aligned}
\tag{4.1}
$$

where Z is a normalization constant, and action applied to object o^i at time t is denoted by u_t^i. ϕ_p is the *prediction potential* that models the temporal persistence of the object poses. ϕ_m is the *measurement potential* that accounts for the observation model given 3D mesh of objects. ϕ_c is the *context potential* that captures the contextual relations between objects.

4.3.1 Prediction Potential

We use two different prediction models for predicting object pose, depending on whether the object is in the field of view or not. If the object is being observed, we model the action u as a continuous random variable that follows a Gaussian distribution with zero mean and small variance Σ. This assumption leads to prediction of small object movements in 3D to be modeled as:

$$o_t^\psi \sim \mathcal{N}(o_{t-1}^\psi, \Sigma)$$

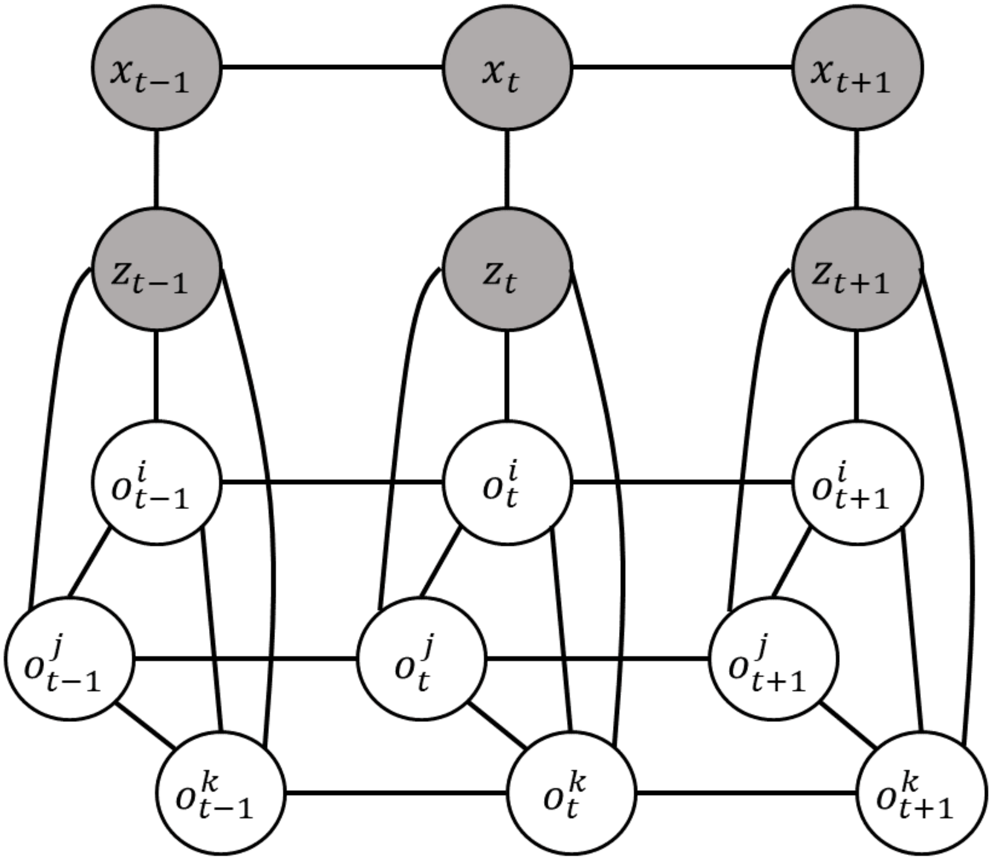


Figure 4.2: Graphical model of the semantic mapping problem. Observed variables are robot poses x_t and observations z_t. Unknown variables are objects $\{o^1, o^2, \cdots, o^N\}$. We compute the posterior over objects while modeling contexual relations between all pairs of objects at each time point, and temporal persistence of each object across consecutive time points.

which allows us to express the prediction potential as:

$$\phi_p(o^i_t, o^i_{t-1}, u^i_{t-1}) = \exp(-(o^\psi_t - o^\psi_{t-1})^T \Sigma^{-1} (o^\psi_t - o^\psi_{t-1})) \tag{4.2}$$

When object o^i is not in the field of view for a significant period of time, it can be either located at the same location or moved to a different location due to the actions applied by other agents. As stated by Toris et al. [167], the probability of the object o^i still being at the same location where it was last seen is a function of time. To take into account the fact that object o^i can be moved to other locations, we model the temporal action u^i with a discrete random variable $\{u_{stay},\ u_{move}\}$. Specifically, u_{stay} denotes no action and the object stays at the same location, and u_{move} denotes a move action is applied and the object is moved to other locations. And these high-level actions follow certain distribution $p(u^i, \Delta t)$,

$$p(u^i = u_{stay}, \Delta t) = r_1 + r_2 \exp(-\frac{\Delta t}{\mu^i}) \tag{4.3}$$

$$p(u_{stay}, \Delta t) + p(u_{move}, \Delta t) = 1 \tag{4.4}$$

where $r_1,\ r_2$ are constants, and Δt is the time duration that object o^i is not being observed. As Δt increases, the probability of u_{stay} decays, and eventually $p(u_{stay}, \Delta t) = r_1$ as $\Delta t \to \infty$. For different objects o^i, the coefficients μ^i that control the speed of the decay are different. We provide heuristic μ^i for different objects in our experiments, while these coefficients can also be learned as introduced by Toris et al. [167].

4.3.2 Measurement Potential

The measurement potential of object o^i_t is expressed as:

$$\phi_m(o^i_t, x_t, z_t) = \begin{cases} \delta, & \text{if } o^i_t \text{ is out of view} \\ g(o^i_t, x_t, z_t), & \text{otherwise} \end{cases}$$

We use non-zero constant δ to account for cases where objects are not in the field of view. $g(o^i_t, x_t, z_t)$ measures the compatibility between the observation z_t and $o^i_t,\ x_t$,

$$g(o^i_t, x_t, z_t) = \sum_{s_k \in S_t} h(o^i_t, s^s_k) l(s^b_k, b(o^i_t, x_t)) f(o^i_t, x_t, I_t)$$

where $h(o^i_t, s^s_k)$ is the confidence score of class o^c_t from the detection confidence vector s^s_k. Function l evaluates the intersection over minimum area of two bounding boxes. $b(o^i_t, x_t)$ is the minimum enclosing bounding box of projected o^i_t in image space based on x_t.

We assume known 3D mesh models of objects. Function $f(o^i_t, x_t, I_t)$ computes the similarity between the projected o_t^i and I_t inside bounding box $b(o_t^i, x_t)$, as explained in detail in section 4.5.2. In the case that robot has observed object o^i in the past, and the belief over o^i indicates that it is in the field of current view of the robot. If the robot cannot detect object o^i, then the object could be occluded, in which case we use $g(o_t^i, x_t, z_t) = f(o_t^i, x_t, I_t)$ for the object to be potentially localized.

4.3.3 Context Potential

There exist common contextual relations between object categories across all environments. For example, a cup would appear on a table much more often than on the floor, and a mouse would appear besides a keyboard much more often than besides a coffee machine. We refer to these common contextual relations as *category-level* contextual relations. In a specific environment, there exist contextual relations between certain object instances. For example, a TV always stays on a certain table, and a cereal box is usually stored in a particular cabinet. We refer to these contextual relations in a specific environment as *instance-level* contextual relations.

We manually encode *category-level* contextual relations as prior knowledge to our model, which also can be learned from public scene dataset (e.g., McCormac et al. [109]). Because *instance-level* contextual relations vary across different environments, these relations of a specific environment must be learned over time. The *context potential* is composed by *category-level* potential ϕ_{cat} and *instance-level* potential ϕ_{ins},

$$\phi_c(o_t^i, o_t^j) = w_1\phi_{cat}(o_t^i, o_t^j) + w_2\phi_{ins}(o_t^i, o_t^j) \tag{4.5}$$

We model $\phi_c(o_t^i, o_t^j)$ as mixture of Gaussians, with $\phi_{cat}(o_t^i, o_t^j)$ and $\phi_{ins}(o_t^i, o_t^j)$ each being a Gaussian component.

In our experiments, we manually designed ϕ_{cat} as prior knowledge, and ϕ_{ins} is updated via Bayesian updates. The principle while designing ϕ_{cat} follows two constraints: 1) simple physical constraints such as no object intersection is allowed, and objects should not be floating in the air,

```
Algorithm 1: Particle filtering in CT-Map
Input: Observation $z_t$, robot pose $x_t$, particle set for each object
       $Q^i_{t-1} = \{\langle o^{i(k)}_{t-1}, \alpha^{i(k)}_{t-1} \rangle | k = 1, \cdots, M\}$
Resample $M$ particles $o^{i(k)}_{t-1}$ from $Q^i_{t-1}$ with probability proportional to importance weights
$\alpha^{i(k)}_{t-1}$ ;
for $i = 1, \cdots, N$ do
    for $k = 1, \cdots, M$ do
        Sample $o^{i(k)}_t \sim \phi_p(o^i_t, o^{i(k)}_{t-1}, u_{t-1})$ ;
        Assign weight $\alpha^{i(k)}_t \propto \phi_m(o^{i(k)}_t, x_t, z_t) \prod_{j \in \Gamma(i)} \phi_c(o^{i(k)}_t, o^j_{t-1})$ ;
    end
end
```

and 2) object pairs that belong to the same category co-occur more often than objects from different categories.

4.4 Inference

We propose a particle filtering based algorithm to perform inference in *CT-MAP*, as given in Algorithm 1. Nonparametric Belief Propagation [157] [64] is not directly applicable to our problem because we are dealing with high-dimensional data. Sener et al. proposed recursive CRF [147] that deals with discrete hidden state with forward-backward algorithm, while our hidden state is mixed, i.e., object class label in discrete space and object pose in continuous space.

Instead of estimating the posterior of the complete history of objects $O_{1:T}$ as expressed in Equation 4.1, *CT-Map* can recursively estimate the posterior of each object $o^i_t \in O_t$. This approach to inference is similar to the CRF-filter proposed by Limketkai et al. [97]. We represent the posterior of object o^i_t with a set of M weighted particles, i.e., $Q^i_t = \{\langle o^{i(k)}_t, \alpha^{i(k)}_t \rangle | k = 1, \cdots, M\}$, where $o^{i(k)}_t$ contains object class and pose information as introduced in 4.3.1, and $\alpha^{i(k)}_t$ is the associated weight for the k^{th} particle. In each particle filtering iteration, particles are first resampled based on their associated weights, then propagated forward in time through object temporal persistence, and re-weighted according to the measurement and context potentials.

We associate bounding boxes across consecutive frames based on their overlap. Only if a bounding box has been consistently associated for certain number of frames will we start initiating object class and pose estimation for that bounding box. The initial set of particles given a detected bounding box s^b_k are drawn as following: 1) first we sample the object class o^c based on the corresponding detection confidence score vector s^s_k; 2) then we sample the 6 DOF object pose o^ψ inside s^b_k, by putting the object center around the 3D points at the center region of s^b_k, with orientation uniformly sampled.

To sample the pose of $o^{i(k)}_t$ from $\phi_p(o^i_t, o^{i(k)}_{t-1}, u_{t-1})$ (Step 4 in Algorithm 1), there are two cases as following:

- If $o^{i(k)}_{t-1}$ is within the field of view of the robot, we sample $o^{i(k)}_t$ according to Equation 4.2.
- If $o^{i(k)}_{t-1}$ is not within the field of view of the robot, we first sample the high-level action $\{u_{stay}, u_{move}\}$ according to Equation 4.3.
    - If u_{stay} is sampled, then $o^{i(k)}_t$ is sampled based on Equation 4.2.
    - If u_{move} is sampled, then another object o^j is uniformly sampled from $O \setminus o_i$, which indicates the place that o^i has been moved to. $o^{i(k)}_t$ is then sampled from the region that o^j can physically support.

In step 5 of Algorithm 1, we use $\Gamma(i)$ to denote the indices of objects that are in the neighborhood of object $o^{i(k)}_t$. Because each neighbor object o^j_{t-1} is represented by M particles, it is computationally expensive to evaluate the context potential $\phi_c(o^{i(k)}_t, o^j_{t-1})$ against each particle of o^j_{t-1}. Thus, we only evaluate the context potential against the most likely particle of o^j_{t-1}. The resulting complexity of the inference algorithm is $\mathcal{O}(NM)$, where N is the number of objects, and M is the number of particles used to represent the belief for each object.

The proposed particle filtering based inference algorithm is an instance of nonparametric belief propagation [157]. In contrast to the push message passing based belief propagation [157], our sampling process in the space of a node is mainly driven by the marginal belief of that node, instead of driven by the pairwise potential. This sampling process has proven to be more effective [40] when the pairwise potential is not peaky.

4.5 Implementation

4.5.1 Faster R-CNN object detector

We deploy Faster R-CNN [138] as our object detector. Given the RGB channel of our RGB-D observation, we apply the object detector and get the bounding boxes from the region proposal network, along with the corresponding class score vector. Then we apply non-maximum suppression to these boxes and merge boxes that have Intersection Over Union (IoU) larger than 0.5. For training, our dataset has 970 groundtruth images for 13 object classes. Each image has around 10 labeled objects. We fine-tuned the object detector based on VGG16 [150] pretrained on COCO [98]. In case of overfitting, we fine-tuned the network for 3000 interations with 0.001 learning rate.

4.5.2 Similarity function

We assume as given the 3D mesh model of objects. Thus, we can render the depth image of o_t^i based on its object class o^c and 6 DOF pose o^ψ in the frame of x_t. With rendered depth image $I(o_t^i, x_t)$, we define the similarity function $f(o_t^i, x_t, I_t)$ as

$$f(o_t^i, x_t, I_t) = e^{-\lambda d(I(o_t^i, x_t), I_t)} \tag{4.6}$$

where λ is a constant scaling factor. $d(I(o_t^i, x_t), I_t)$ is the sum of squared differences between the depth values in observed and rendered depth images.

4.6 Experiments

We collected our indoor scene dataset with a Michigan Progress Fetch robot for evaluation on our proposed *CT-Map* method. Our indoor scene dataset contains 20 RGB-D sequences of various indoor scenes. We measure the quality of inference for various scenes in terms of 1) object detection and 2) pose estimation. Thus, we follow the mean average precision (mAP) metric and 6

	Faster R-CNN	*T-Map*	*CT-Map*
mAP	0.607	0.715	0.871

Table 4.1: mAP on our scene dataset.

DOF pose estimation accuracy for benchmarking our method. We also show qualitative examples of our semantic maps in Figure 4.1. More qualitative examples are provided in this video[1]. On a computer with i7 2.60GHz CPU and Nvidia GeForce GTX 980M, our implementation of the semantic mapping algorithm runs at 1 FPS (2 particle filtering iterations per frame) on average. The complexity of the inference is as discussed previously.

Across all experiments, we use $w_1 = w_2 = 0.5$ in Equation 4.5 to treat *category-level* and *instance-level* potentials equally. If an object has not been observed for infinite long period of time, we assume that object has equal probabilities of either staying at the same location or not. Thus, we use $r_1 = r_2 = 0.5$ in Equation 4.3.

4.6.1 Object Detection

We have noisy object detections coming from baseline Faster R-CNN object detector, while *CT-Map* can correct some false detections by modeling the object class as part of our hidden state. To evaluate the object detection performance of *CT-Map*, we take the estimated 6 DOF pose of all objects in the scene at the end of each RGB-D sequence in our dataset, and project them back onto each camera frame in that sequence to generate bounding boxes with class labels. We run two semantic mapping processes by considering different sets of potentials: 1) Temporal Mapping (*T-Map*): we consider prediction potential in the CRF model; 2) Contextual Temporal Mapping (*CT-Map*): we consider both prediction and context potential in the CRF model, which is the proposed method. For both *T-Map* and *CT-Map*, we include the measurement potential on observation.

We use mAP as our object detection metric. As shown in Table 4.1, *T-Map* improves upon the baseline method Faster R-CNN by incorporating prediction and observation potentials, and *CT-Map* improves the performance further by additionally incorporating context potential. Faster

[1] https://youtu.be/W-6ViSlrrZg

R-CNN did not perform quite well on the test scenarios because the training data do not necessarily cover the variances encountered at test time. Though the performance of Faster-RCNN can be further improved by providing more training data, *CT-Map* provides more robust object detection when training remains limited.

In some cases, objects are not being reliably detected by Faster R-CNN due to occlusion. If an object has been observed in the environment in the past, our method makes predictions on locations that objects can go by modeling the temporal persistence of objects. Thus, even if a detection is not fired on the object due to occlusion, our method can still localize the object and claim a detection. However, in cases where an object is severely occluded and the depth observation lacks enough geometric information from the object, our method will not be able to localize the object. Example detection results highlighting the benefits of the proposed method compared to baseline Faster R-CNN are shown in Figure 4.4.

4.6.2 Pose Estimation

For each RGB-D sequence in our dataset, we locate the frames that each object is last seen, and project the depth frame back into 3D point clouds using known camera matrix. We then manually label the ground truth 6 DOF pose of objects. We compare the estimated object poses at the end of each RGB-D sequence against the ground truth.

Pose estimation accuracy is measured as $accuracy = \frac{N_{correct}}{N_{total}}$, where $N_{correct}$ is the number of objects that are considered correctly localized, and N_{total} is the total number of objects that are present in the dataset. If the object pose estimation error falls under certain position error threshold Δt and rotation error threshold $\Delta\theta$, we claim that the object is correctly localized. Δt is the translation error in Euclidean distance, and $\Delta\theta$ is the absolute angle difference in orientation. For symmetrical objects, the rotation error with respect to the symmetric axis is ignored.

We apply the Iterative Closest Point (ICP) and Fast Point Feature Histogram (FPFH) [139] algorithms as our baselines for 6 DOF object pose estimation. For each RGB-D sequence in our dataset, we take the 3D point clouds of the labeled frame, and crop them based on ground truth bounding boxes. These cropped point clouds are given to the baselines as observations, along with object 3D mesh models. ICP and FPFH are applied to register the object model to the cropped observed point cloud. We allow maximum iterations of 50000.

Our proposed method *CT-Map* significantly outperforms ICP and FPFH by a large margin. As our generative inference iteratively samples object pose hypotheses and evaluates them against the observations, *CT-Map* does not suffer from local minima as much as discriminative methods such as ICP and FPFH.

4.7 Conclusion

We propose a semantic mapping method *CT-Map* that simultaneously detects objects and localizes their 6 DOF pose given streaming RGB-D observations. *CT-Map* represents the semantic map with a belief over object classes and poses. We probabilistically formalize the semantic mapping problem in the form of a CRF, which accounts for contextual relations between objects and temporal persistence of object poses, as well as measurement potential on observation. We demonstrate that *CT-Map* outperforms Faster R-CNN in object detection and FPFH, ICP in object pose estimation. As a step forward discussed in the next chapter, we would like to investigate the inference problem of object semantic locations given partial observations of an environment, e.g., inferring a query object to be on a dining table, or in a kitchen cabinet. Ideally, maintaining a belief over object semantic locations can serve as a notion of generalized object permanence, and facilitate object search tasks.

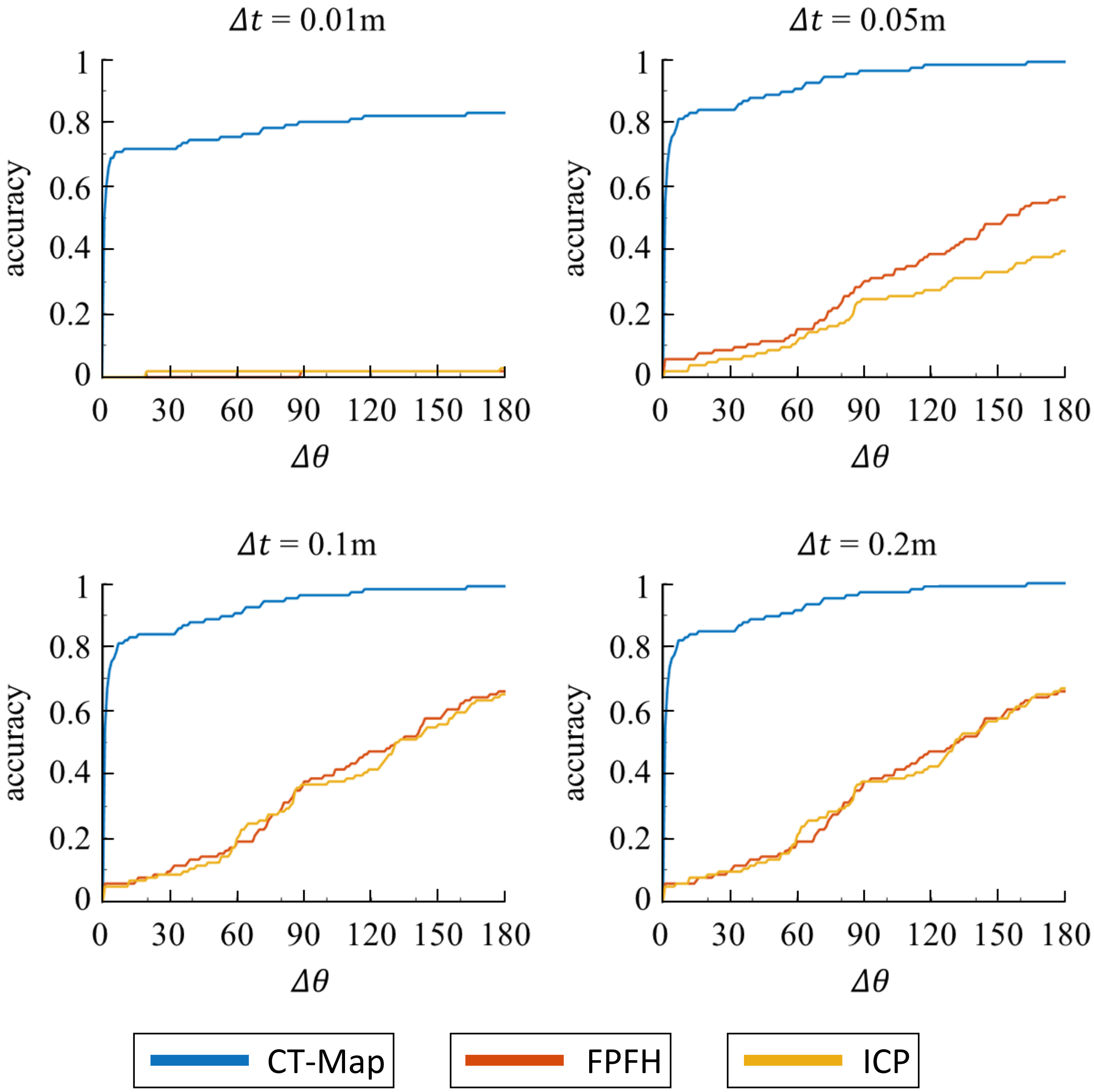


Figure 4.3: Object pose estimation of *CT-Map*, compared with FPFH and ICP based baselines. Different plots correspond to different pose estimation correctness criteria defined by position error threshold Δt and rotation error threshold $\Delta\theta$. Our method outperforms FPFH and ICP with a large margin.

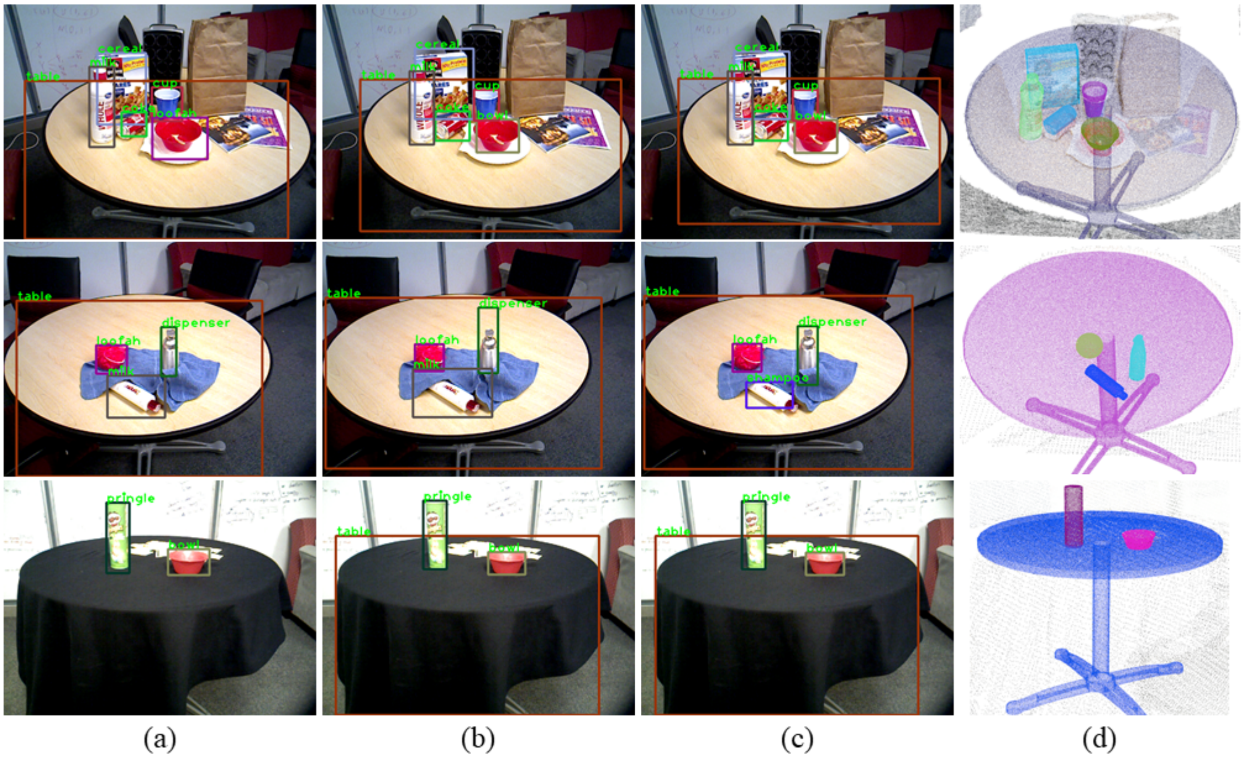


Figure 4.4: Mapping examples highlighting detection improvements: (a) raw detection results from baseline Faster R-CNN; (b) detection results from *T-Map* when only considering measurement and prediction potential; (c) detection results *CT-Map* when considering measurement, prediction and context potential; (d) 6 DOF object pose estimates from *CT-Map*. We generate bounding boxes in column (b) and (c) by projecting the localized 3D objects into 2D image space, and finding the minimum enclosing boxes of the projections. The first row shows Faster R-CNN gives false detection on the red bowl as ”loofah”, while both *T-Map* and *CT-Map* correct the wrong label ”loofah” into ”bowl”. The second row shows Faster R-CNN gives false detection on the shampoo bottle as ”milk”, and *T-Map* fails to correct the wrong label because the geometry of milk and shampoo is similar, while *CT-Map* successfully corrects the wrong label into ”shampoo” based on the context. The third row shows Faster R-CNN does not detect the table due to the appearance change induced by the table cloth, while both *T-Map* and *CT-Map* successfully detect and localize the table. Because the table used to be observed around that location in the past, and our methods benefit from modeling the temporal persistence of object poses. (Best viewed in color)

CHAPTER 5

GOP/SLiM: Generalized Object Permanence with Semantic Linking Maps for Active Visual Object Search

In this chapter, we focus on bringing the overall *SRP* framework to a large scale (e.g. floor level). Building on the *SRP* framework as discussed in chapter 3 and on-line semantic mapping approach presented in chapter 4, we aim to enable a mobile manipulator robot to perform user desired tasks at a large scale, given a user demonstrated goal scene. The robot sensor has a limited field of view, leading to the cases where objects required for the user desired task are not within the field of view at task execution time. Thus, the challenge is to effectively search and retrieve objects required for the task from the environment.

Psychology research [57] [116] reveals that object permanence plays an important role in reasoning of object locations in cognitive development. Specifically, object permanence refers to the understanding that an object that was observed continues to exist even it cannot be perceived. We aim to model a generalized version of object permanence to reason about possible locations of an object that may not have been observed in the environment before, and is currently not being directly observed. Landmark objects can help this reasoning by narrowing down the search space significantly. More specifically, we can exploit long-term occurrence history, short-term recent observations, and common sense knowledge about common spatial relations between landmark and target objects. For example, seeing a table and knowing that cups can often be found on tables aids the discovery of a cup. Such correlations can be expressed as distributions over possible pairing relationships of objects. We introduce Generalized Object Permanence (GOP) as the problem of modeling such correlations. We propose to formally model generalized object permanence through a factor graph. Each node in the factor graph corresponds to the spatial relation between a pair of objects. Each node is associated with multiple factors with each representing a source of inter-object relation information. Inference on the factor graph leads to marginal beliefs on inter-object spatial relations between each pair of objects across the environment.

With modeled GOP, we propose an active visual object search method through our introduction of the Semantic Linking Maps (*SLiM*) model [188]. *SLiM* simultaneously maintains the belief over a target object's location as well as landmark objects' locations, while accounting for probabilistic inter-object spatial relations. We build *SLiM* on *CT-Map* (as discussed in the previous chapter), by extending *CT-Map* to consider probabilistic inter-object spatial relations. Based on *SLiM*, we describe a hybrid search strategy that selects the next best view pose for searching for the target object based on the maintained belief. We demonstrate the effectiveness of our *SLiM*-based search strategy through comparative experiments in simulated environments. We further demonstrate the real-world applicability of *SLiM*-based search in scenarios with a Fetch mobile manipulation robot.

5.1 Introduction

Being able to effectively search for objects in an environment is crucial for service robots to autonomously perform tasks [73, 173, 59]. When asked where a target object can be found, humans are able to give hypothetical locations expressed by spatial relations with respect to other objects. For example, a *cup* can be found "on a table" or "near a sink". *Table* and *sink* are considered landmark objects that are informative for searching for the target object *cup*. Robots should be able to reason similarly about objects locations, as shown in Figure 5.1.

Previous works [78, 90, 167] assume landmark objects are static, in that they mostly remain where they were last observed. This assumption can be invalid for dynamic landmark objects that change their location over time, such as chairs, food carts and toolboxes. Temporal assumptions can mislead the search process if the prior on the landmarks' locations is too strong. Further, there also exists uncertainty in the spatial relations between landmark objects and the target object, and between landmark objects themselves. For example, a *cup* can be "in" or "next to" a *sink*.

Considering the problem of dynamic landmarks, we propose the Semantic Linking Maps (*SLiM*) model to account for uncertainty in the locations of landmark objects during object search. Building on Lorbach et al. [104], we model inter-object spatial relations probabilistically via a factor graph. The marginal belief on inter-object spatial relations inferred from the factor graph is used in *SLiM* to account for probabilistic spatial relations between objects.

Using the maintained belief over target and landmark objects' locations from *SLiM*, we propose a hybrid strategy for active object search. We select the next best view pose, which guides the robot to explore promising regions that may contain the target and/or landmark objects. Previous works [178, 53, 152, 11] have shown the benefit of purposefully looking for landmark objects (*Indirect Search*) before directly looking for the target object (*Direct Search*). The proposed hybrid search strategy draws insights from both indirect and direct search. We demonstrate the effectiveness of the proposed hybrid search strategy in our experiments.

We describe the Semantic Linking Maps model as a Conditional Random Field (CRF). Our description of *SLiM* as a CRF allows us to simultaneously maintain the belief over target and landmark object locations with probabilistic modeling over inter-object spatial relations. We also describe a hybrid search strategy based on *SLiM* that draws upon ideas from both indirect and direct search representations. This *SLiM*-based search makes use of the maintained belief over objects' locations by selecting the next best view pose based on the current belief. In our experiments, we show that the proposed object search approach is more robust to noisy priors on landmark locations by simultaneously maintaining belief over the locations of target and landmark objects.

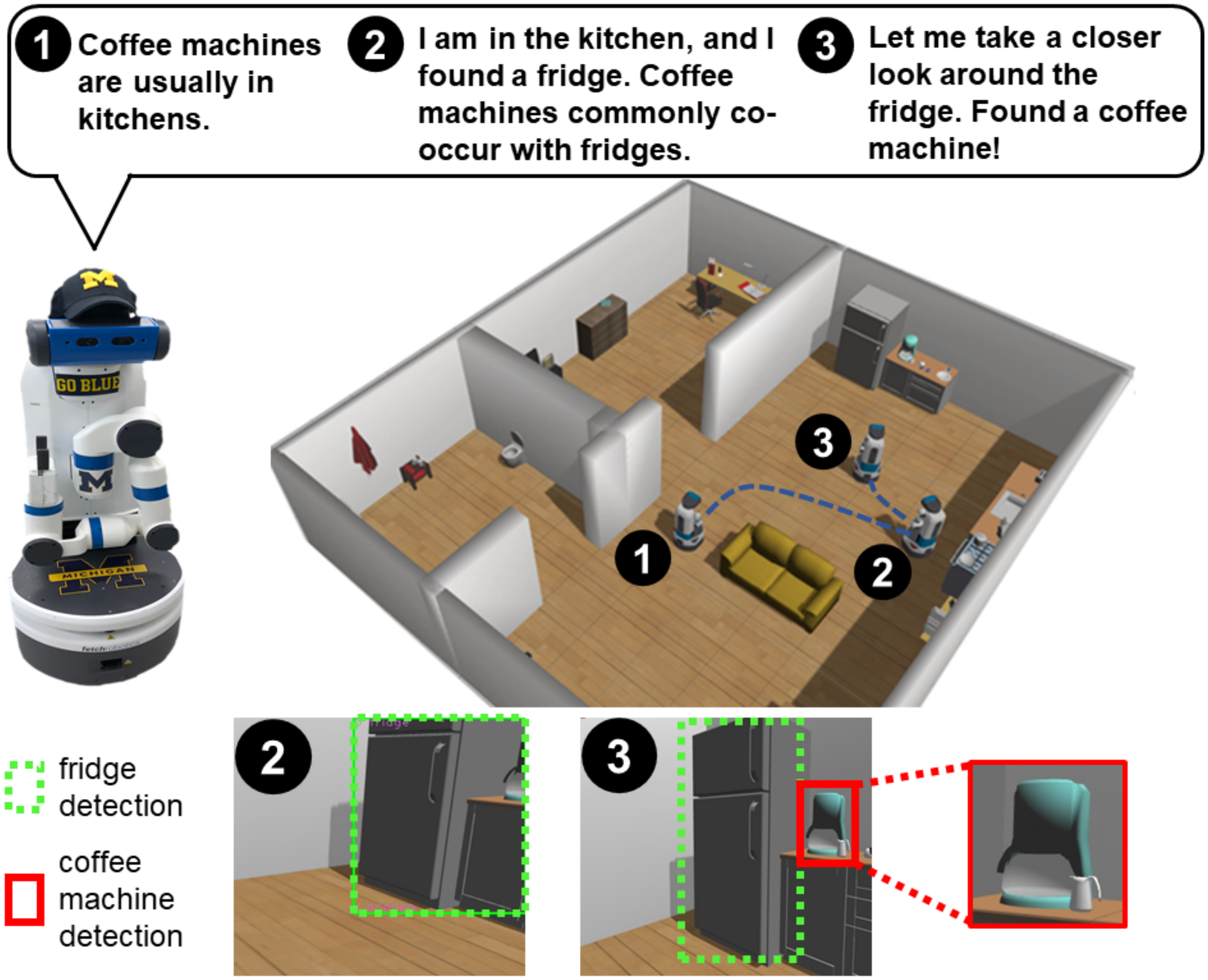

1 Coffee machines are usually in kitchens.
2 I am in the kitchen, and I found a fridge. Coffee machines commonly co-occur with fridges.
3 Let me take a closer look around the fridge. Found a coffee machine!
GO BLUE
MICHIGAN
1
2
3
fridge detection
coffee machine detection
2
3

Figure 5.1: Robot tasked to find a coffee machine.

5.2 Related Work

Existing works have studied object search with different assumptions on prior knowledge of the environment. Some assume priors on landmark objects' locations in the environment, and utilize the spatial relations between the target object and landmark objects to prioritize regions to search. Kollar et al. [78] utilize object-object co-occurrences extracted from image tags on Flickr.com to infer target object locations. Kunze et al. [90] expanded the generic notion of co-occurrences to more restrictive spatial relations (e.g. "in front of", "left of"), which provide more confined regions to search, thus improving the search efficiency. Toris et al. [167] proposed to learn a temporal model on inter-object spatial relations to facilitate search. These methods assume the landmark objects to be static, however, we believe accounting for the uncertainty in landmark objects' locations is important for object search.

Existing works have also explored known priors on spatial relations between landmark and target objects. Given exact spatial relations between landmark and target objects, Sjöö et al. [152] used an *indirect object search* strategy [178, 53], where the robot first searches for landmark objects, and then searches for a target object in regions satisfying given spatial relations. On the other hand, given a probabilistic distribution over the spatial relations between objects, Aydemir et al. [11] formulate the object search problem as a Markov Decision Process. In our work, we learn the probabilistic inter-object spatial relations by building on ideas of Lorbach et al. [104], where inter-object relations are being probabilistically modeled via a factor graph.

There are also works that do not assume prior knowledge of the environment. Researchers have explored object search with visual attention mechanisms [148, 151, 112], such as saliency detection. Similar to [78, 90], other research [103, 44, 70] utilizes object-object co-occurrences to guide the search for a target object. Positive and negative detections of landmark objects will result in an updated belief over the target object. We expand object-object co-occurrences to finer-grained spatial relations between objects, i.e., “in”, “on”, “proximity”, “disjoint”, which specify more confined regions for object search.

Other literature [177, 89, 174] has also explored object-place relations to facilitate object search. Wang et al. [177] build a belief road map based on object-place co-occurrences for efficient path planning during object search. Kunze et al. [89] bootstraps commonsense knowledge on object-place co-occurrences from the Open Mind Indoor Common Sense (OMICS) dataset. Samadi learned similar knowledge by actively querying the World Wide Web (WWW). Our work also takes object-place co-occurrences into account. Aydemir et al. [10] made use of place-place co-occurrences to infer the type of the room next door, as the robot explores an environment during search. Other research [182, 179, 96] enable robots to search for objects through manipulations of objects to reveal objects in clutter.

From the perspective of planning for object search, we use a greedy strategy with a horizon of one step, similar to [90, 167, 12, 152, 103, 44]. Thus, the robot always selects the next best view pose that maximize on our proposed utility function without accounting for expected accumulated utility in the future. Other research [78, 177, 182, 179, 96, 11] account for expected accumulated utility or cost in the future by path planning, or formulating a MDP or POMDP problem around the object search task. Furthermore, object search tasks as well as mobile manipulation tasks in general can benefit from task and motion planning [71]. Lo et al. [102] proposed task and motion planning for task-oriented navigation tasks, which can also be extended for object search tasks to minimize the travelled path during the search. Instead of designing integrated task and motion planning systems for particular tasks, Srivastava et al. [153] provides a generic interface layer for any off-the-shelf task planners and motion planners for task and motion planning.

5.3 Problem Statement

Let $O = \{o^1, o^2, \cdots, o^N\}$ be the set of objects of interest, including landmark objects and the target object for search. Given observations $z_{0:T}$ and robot poses $x_{0:T}$, we aim to maintain the belief over object locations $P(O_T|x_{0:T}, z_{0:T})$, while accounting for the probabilistic spatial relations R_{ij} between objects $o^i, o^j \in O$. For this work, we consider the set of spatial relations to be $R_{ij} \in$ {*In, On, Contain, Support, Proximity, Disjoint*}. For example, the relation $R_{ij} = In$ indicates that object o_i is inside object o_j. The probabilistic spatial relations between object o^i, o^j is represented by the belief over R_{ij}, denoted as $\mathcal{B}(R_{ij})$.

Based on the maintained belief $P(O_T|x_{0:T}, z_{0:T})$, the robot searches for the target object by selecting the next best view pose ranked by an utility function $U : \vec{\tau} \mapsto \mathbb{R}$. $\vec{\tau}$ specifies the 6 DOF of camera view pose. The utility function U trades off between navigation cost and the probability of search success. Upon a user request to find a target object, the robot iterates between the belief update of objects' locations and view pose selection, until the target object is found or the maximum search time is reached.

5.4 Semantic Linking Maps

For Semantic Linking Maps (*SLiM*), we consider inter-object spatial relations, while maintaining the belief over target and landmark objects' locations. Building on *CT-Map* [189] as discussed in the previous chapter, we probabilistically formalize the object location estimation problem via a Conditional Random Field (CRF). The model is now extended to account for probabilistic inter-object spatial relations, as shown in Figure 5.2.

The posterior probability of the object locations $O = \{o^1, o^2, \cdots, o^N\}$ is expressed as:

$$\begin{aligned} & p(O_{0:T}|x_{0:T}, z_{0:T}) = \\ & \frac{1}{Z} \prod_{t=0}^{T} \prod_{i=1}^{N} \phi_p(o_t^i, o_{t-1}^i) \phi_m(o_t^i, x_t, z_t) \prod_{i,j} \phi_{c,\mathcal{B}(R_{ij})}(o_t^i, o_t^j) \end{aligned} \tag{5.1}$$

where Z is a normalization constant. Robot pose x_t and observation z_t are known. We assume that the robot stays localized given a metric map of the environment.

$\phi_p(o_t^i, o_{t-1}^i)$ is the *prediction potential* that models the movement of an object over time. We assume objects to remain static or move with temporal coherence during the search, i.e.

$$\phi_p(o_t^i, o_{t-1}^i) = e^{-(o_t^i - o_{t-1}^i)^T \Sigma^{-1} (o_t^i - o_{t-1}^i)}$$

$\phi_m(o_t^i, x_t, z_t)$ is the *measurement potential* that accounts for the observation model, and $z_t = \{z_t^1, z_t^2, \cdots, z_t^N\}$ is object detection at time t. Each z_t^i represents (potentially noisy) detections fired for object o^i. At time t, because z_t^i, o^j for $j \neq i$ are independent, we simplify $\phi_m(o_t^i, x_t, z_t)$ to $\phi_m(o_t^i, x_t, z_t^i)$ s.t.,

$$\phi_m(o_t^i, x_t, z_t^i) = \begin{cases} P_{FN}, & \text{if } o_t^i \in E_t^i \text{ and } z_t^i = \emptyset \\ P_{TN}, & \text{if } o_t^i \notin E_t^i \text{ and } z_t^i = \emptyset \\ P_{TP}, & \text{if } \pi(o_t^i) \in z_t^i \\ P_{FP}, & \text{otherwise} \end{cases} \tag{5.2}$$

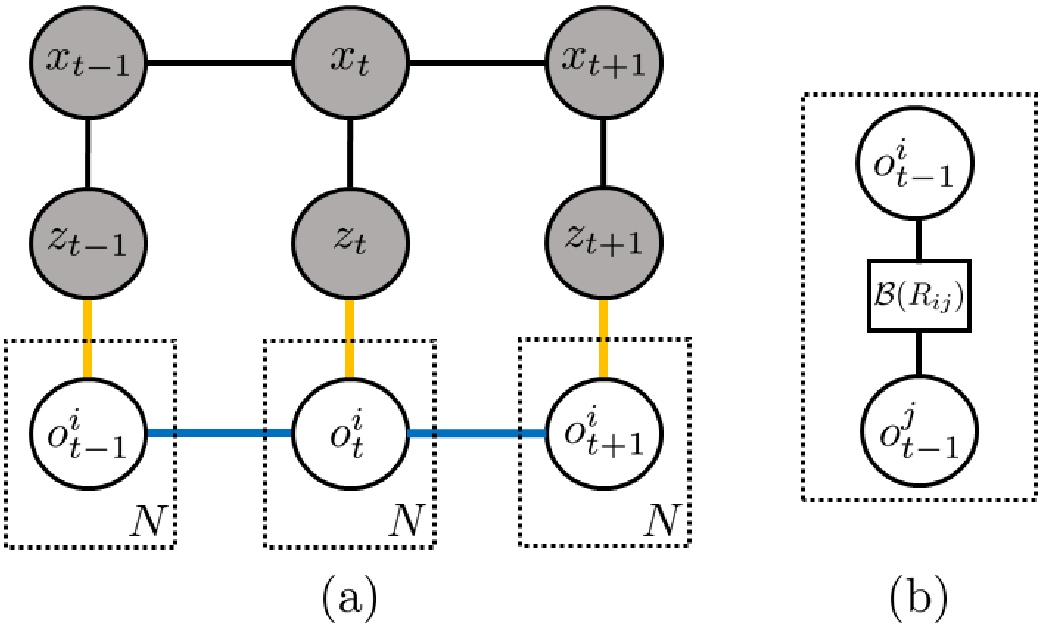


Figure 5.2: CRF-based *SLiM* model: (a) Known: $\{x^t\}$ robot poses, $\{z^t\}$ sensor observations; Unknown: $O_t = \{o^1_t, o^2_t, \cdots, o^N_t\}$. (b) Plate notation: at time t, the spatial relations between each object pair o^i, o^j is parameterized by the belief over their spatial relations $\mathcal{B}(R_{ij})$.

where $P_{FN}, P_{TN}, P_{TP}, P_{FP}$ stands for the probability of false negative, true negative, true positive, and false positive. E^i_t is the effective observation region for object o^i given the robot's camera pose at time t. Note, the robot has a larger effective observation region for larger objects, because they can be reliably detected from further away compared to small objects. π is the camera projection matrix, and $\pi(o^i_t) \in z^i_t$ denotes that the projected object lies in the detected bounding box in z^i_t.

We model the spatial relations between objects with *context potential* $\phi_{c,\mathcal{B}(R_{ij})}$. Here, we extend ϕ_c from our previous work by paramtererizing it with the belief $\mathcal{B}(R_{ij})$ over the inter-object spatial relation between o^i, o^j,

$$\phi_{c,\mathcal{B}(R_{ij})} = \sum_r \mathcal{B}(R_{ij} = r)\phi_{c,r}(o^i_t, o^j_t, R_{ij} = r) \tag{5.3}$$

where r can take any value in the set of possible relations $\{$*In, On, Contain, Support*, *Proximity, Disjoint*$\}$.

For $r \in \{$*In, On, Contain, Support*$\}$, $\phi_{c,r}(o^i_t, o^j_t, R_{ij} = r)$ is equal to 1 if objects o^i_t, o^j_t satisfy the spatial relation given the width, length and height of the object, otherwise 0. For $r =$ *Proximity*,

$\phi_{c,r}(o_t^i, o_t^j, R_{ij} = Proximity)$ corresponds to a Gaussian distribution that models $o_t^j \sim \mathcal{N}(o_t^i, \Sigma^{ij})$ and Σ^{ij} is determined by the size of objects o^i, o^j. The larger the size of o^i, o^j, the larger the variance in Σ^{ij}. For $r =$ ***Disjoint***, $\phi_{c,r}(o_t^i, o_t^j, R_{ij} = Disjoint) = 1 - \sum_{r \neq Disjoint} \phi_{c,r}(o_t^i, o_t^j, R_{ij} = r)$.

5.4.1 Inference

We propose a collaborative particle filtering based inference method for maintaining the belief over object locations, as shown in Algorithm 2. Instead of estimating the posterior of the complete history of object locations $p(O_{0:T}|x_{0:T}, z_{0:T})$, we recursively estimate the posterior probability of each object $o_t^i \in O_t$, similarly to [189, 97]. The complexity of the inference algorithm is $\mathcal{O}(NKM^2)$, where N is the number of objects, K is the number of neighbor objects, and M is the number of particles used to represent the belief of each object. In step 6 as described in Algorithm 2), for each neighbor object, the context potential is computed between each of the M particle $o^i(k)$ and object o^j which is represented by M weighted particles $o^j(.)$, thus the quadratic term M^2 in the complexity. Further works can be done to decrease the complexity down to $\mathcal{O}(NKMC)$ by sampling C representative and divergent particles among particles $o^j(.)$, where C is much less than M.

We represent each object with M weighted particles. To deal with particle decay, for each object o^i, we reinvigorate the particles by sampling from known room areas, as well as sampling around other objects o^j based on $\mathcal{B}(R_{ij})$. Across our experiments, we use 100 particles for each object, and we only establish the edge of *context potential* between objects o^i, o^j if $1 - \mathcal{B}(R_{ij} =$ *Disjoint*$) > 0.2$. Examples of the belief update over time are available in Figure 5.3.

The proposed collaborative particle filtering based inference algorithm is an instance of approximate nonparametric belief propagation [157]. Similarly to CT-Map in previous chapter, our sampling process in the space of a node is mainly driven by the marginal belief of that node, instead of driven by the pairwise potential as in [157]. And this sampling process has proven to be more effective [40] when the pairwise potential is not peaky.

Algorithm 2: Inference of objects locations in *SLiM*.

```
Input: Observation $z_t$, Robot pose $x_t$,
       Particle set for each object:
           $o^i_{t-1} = \{\langle o^{i(k)}_{t-1}, \alpha^{i(k)}_{t-1}\rangle | k = 1, \cdots, M\}, i \in 1 : N$
Resample $M$ particles $o^{i(k)}_{t-1}$ from $o^i_{t-1}$ with probability proportional to importance weights
  $\alpha^{i(k)}_{t-1}$ ;
for $i = 1, \cdots, n$ do
    for $k = 1, \cdots, M$ do
        Sample $o^{i(k)}_t \sim \phi_p(o^i_t, o^{i(k)}_{t-1})$ ;
        Assign weight $\alpha^{i(k)}_t \propto \phi_m(o^{i(k)}_t, x_t, z_t) \prod_{j \in \Gamma(i)} \phi_{c,\mathcal{B}(R_{ij})}(o^{i(k)}_t, o^j_{t-1})$ ;
          where $\phi_{c,\mathcal{B}(R_{ij})}(o^{i(k)}_t, o^j_{t-1}) = \sum_r \sum_{l=1}^{M} \mathcal{B}(R_{ij} = r)\alpha^{j(l)}_{t-1}\phi_{c,r}(o^i_t, o^j_t, R_{ij} = r)$
    end
end
```

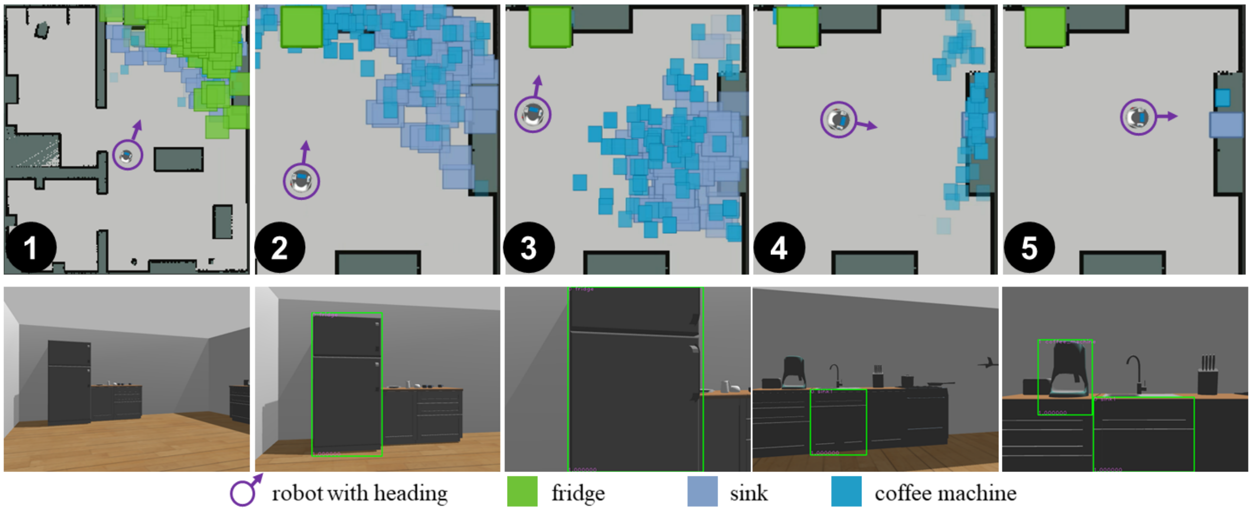


Figure 5.3: Examples of belief updates in *SLiM*. given observations. *Upper*: Evolution of particles of *fridge, sink, coffee machine* over time. *Lower*: RGB observation (with object detection) over time. (Best viewed in color).

5.4.2 Probabilistic Inter-Object Spatial Relations

To get the belief over inter-object spatial relations $\mathcal{B}(R_{ij})$ for each object pair $o^i, o^j \in O$, we model the generalized object permanence based on past observations of the environment and common sense knowledge on inter-object spatial relations. Building on preceding work by Lorbach et al. [104], we model GOP through a factor graph as shown in Figure 5.4. We generalize [104] by

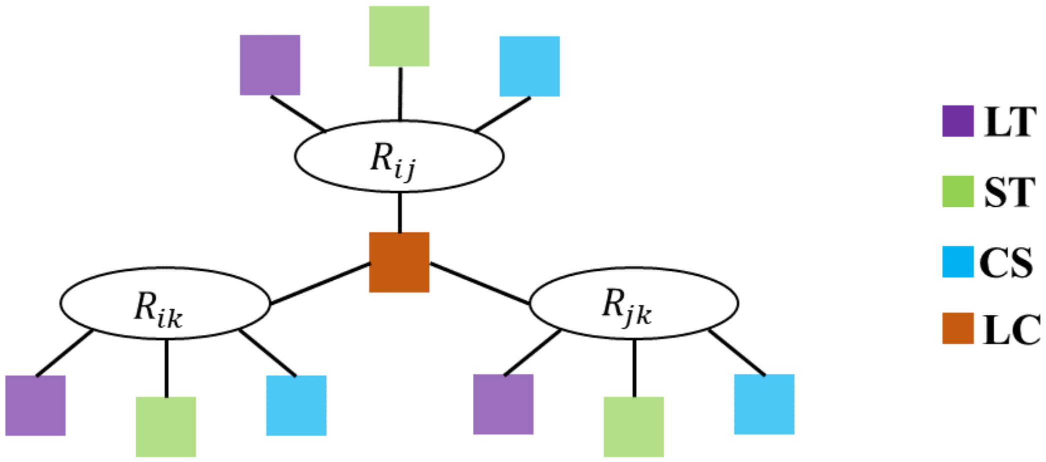


Figure 5.4: The SLiM factor graph to model for Generalized Object Permanence. SLiM accounts for various factors in the semantic relation R_{ij} between any two objects, i and j: **LT**: long term memory, **ST**: short-term memory, **CS**: common sense knowledge, **LC**: scene consistency.

relaxing the assumption on known spatial relations between landmark objects.

The factor graph $G : \{\mathbb{V}, \mathbb{F}, \mathbb{E}\}$ consists of variable vertices $\mathbb{V} = \{R_{ij} | \forall_{i \neq j}\ o^i, o^j \in O\}$, factor vertices $\mathbb{F} = \{F_{CS}, F_{LT}, F_{ST}, F_{LC}\}$ and edges $\mathbb{E}$ which connect factor vertices with variable vertices. Specifically, $F_{CS} : R_{ij} \mapsto \mathbb{R}$ is a unary factor that considers *commonsense knowledge* on spatial relation between objects,

$$F_{CS}(R_{ij}) = \text{Frequency}(R_{ij})$$

Similar to [104], we extract commonsense knowledge on R_{ij} from the Google Image search engine by counting the frequency of certain spatial relation between objects o^i, o^j. For example, the frequency of $R_{cup,table} = On$ is computed as the number of search results of a query "cup on the table" divided by the number of search results of a query "on the table". These extracted frequencies can be noisy. For example, the frequency of "laptop on kitchen" is larger than 0. However, it is not a valid expression because it refers to a laptop being on top of the room geometry of a kitchen. We manually encode the $F_{CS}(R_{ij})$ for invalid expressions to 0.

$F_{LT} : R_{ij} \mapsto [0, 1]$ is a function that bookkeeps the frequency of occurrences of R_{ij} in past observation of the environment. This factor accounts for long term knowledge of inter-object relations based on past observations in the long run. For example, *cereal* is usually stored on a particular *cupboard*, and *silverware* is usually stored in a particular *drawer*.

The short-term memory factor $F_{ST} : R_{ij} \mapsto [0, 1]$ informs the reasoning about a particular inter-object relation observed in the near past. Specifically, F_{ST} is modeled as

$$F_{ST}(R_{ij}) = e^{-\frac{\Delta t}{\mu}} \tag{5.4}$$

where Δt is the time that has past since last observing R_{ij}. μ controls how fast the probability of R_{ij} decays from the short-term perspective. We provide heuristic μ in our experiments, while these parameters can be learned as discussed by Toris. et al. [167].

$F_{LC} : (R_{ij}, R_{ik}, R_{jk}) \mapsto \{0, 1\}$ is a triplet factor that considers *logical consistency* between a triplet of objects o^i, o^j, o^k,

$$F_{LC}(R_{ij}, R_{ik}, R_{jk}) = \begin{cases} 1, & \text{if consistent.} \\ 0, & \text{otherwise.} \end{cases}$$

For example, if o^i is in o^j, and o^j is in o^k, then o^i should be in o^k to satisfy logical consistency, i.e., $F_{LC}(R_{ij} = \textit{In}, R_{ik} = \textit{In}, R_{jk} = \textit{In}) = 1$. Previous work [104] assumes the spatial relations between landmark objects to be known, and only relations $R_{target,j}$ connecting target object o_{target} and landmark object o_j to be unknown. Their pairwise factor enforcing *logical consistency* is a binary function $F_{LC} : (R_{target,j}, R_{target,k}) \mapsto \{0, 1\}$. In contrast, our formulation employs a trinary factor F_{LC} considering all possible combinations of (R_{ij}, R_{ik}, R_{jk}) and evaluating their logical consistency.

By applying Belief Propagation [86] on the factor graph formulated as above, we can get the marginal belief over inter-object relations $\mathcal{B}(R_{ij})$ between all object pairs. We use the libDAI [115] library for inference. An example of the probabilistic inter-object spatial relations inferred from the factor graph is as shown in Figure 5.6, and it is used in our experiments.

5.5 Search Strategy

Based on the belief over the object locations, we actively search for the target object, by generating promising view poses and select the best one ranked by a utility function. Given the particle set $\langle o_t^{(k)}, \alpha_t^{(k)} \rangle$ of the target object o as being maintained in 5.4, we fit Gaussian Mixture Models (GMMs) through Expectation Maximization to the particles by auto selecting the number of clusters [46],

$$\langle o_t^{(k)}, \alpha_t^{(k)} \rangle \sim \langle \mathcal{N}(x_n, \Sigma_n), \omega_n \rangle \tag{5.5}$$

5.5.1 View Pose Generation

For each Gaussian component $\mathcal{N}(\vec{x}_n, \Sigma_n)$, we generate a set of camera view pose candidates $\{\vec{\tau}_n^i = (\vec{c}_n^i, \vec{\psi}_n^i)\}$, where $\vec{c}_n$ and $\vec{\psi}_n$ denote the translation and the rotation of the camera respectively.

Initially, we sample the location of the camera $\vec{c}_n$ evenly from a circle with a fixed radius around the center $\vec{x}_n$ of the Gaussian component, and assign a default value to rotation $\vec{\psi}_n$. Note, that these initially sampled view poses can put the robot in collision with the environment, and the camera is not necessarily looking at $\vec{x}_n$. Thus, we formulate a view pose optimization problem under constraints as below,

$$\underset{\vec{\tau}_n}{\text{argmin}}\ 1 - \vec{v}_n \cdot \frac{\vec{x}_n - \vec{c}_n}{\|\vec{x}_n - \vec{c}_n\|} \ \text{ s.t } \vec{x}_n \in E_{\vec{\tau}_n}, \quad c(\vec{\tau}_n) > 0$$

where $\vec{v}_n$ is the view direction given $\vec{\tau}_n$, $E_{\vec{\tau}_n}$ denotes the effective observation region of the target object at camera pose $\vec{\tau}_n$, and $c : \vec{\tau} \mapsto \mathbb{R}$ is a function that computes a signed distance of a configuration $\vec{\tau}$ to the collision geometry of the environment.

5.5.2 View Pose Selection

We propose two different utility functions to rank the view pose candidates:

5.5.2.1 Direct Search utility

$\mathbf{U}_{\text{DS}}$ encourages the robot to explore promising areas that could contain the target object while accounting for navigation cost,

$$\mathbf{U}_{\text{DS}}(\vec{\tau}_k) = \omega_n + \alpha \frac{1}{\arctan(\sigma d_{nav})} \tag{5.6}$$

where ω_n is the weight of the Gaussian component (as in (5.5)) that $\vec{\tau}_k$ is generated from, and d_{nav} is the navigation distance from the current robot location to view pose $\vec{\tau}_k$. Parameter α trades off between the probability of finding the target object and the navigation cost. Parameter σ determines how quickly the $\arctan(\sigma d_{nav})$ plateaus.

With $\mathbf{U}_{\text{DS}}$, the object search id **direct** because we are directly considering promising areas represented by the GMMs for the target object.

5.5.2.2 Hybrid Search utility

$\mathbf{U}_{\text{HS}}$ encourages the robot to explore promising areas that could contain the target object and/or any landmark object, while accounting for navigation cost

$$\begin{aligned}\mathbf{U}_{\text{HS}}(\vec{\tau}_k) = \omega_n &+ \alpha \frac{1}{\arctan(\sigma d_{nav})} \\ &+ \beta \max_{j,n} \text{CoOccur}(o, o^j)\omega_n^j\, \mathbf{I}_n^j\end{aligned}$$

where the additional term compared to $\mathbf{U}_{\text{DS}}$ acts to encourage the robot to also explore areas that could contain landmark object o^j which co-occurs with the target object o with probability $\text{CoOccur}(o, o^j)$. Specifically, $\text{CoOccur}(o, o^j) = (1 - \mathcal{B}(R_{target,j} = \textit{Disjoint}))$, and ω_n^j is the weight of the n-th Gaussian component of GMMs fitted to the belief over the location of the landmark object o^j. And $\mathbf{I}_n^j$ is 1 if the n-th Gaussian of object o^j is within the effective observation region at camera pose $\vec{\tau}_k$, otherwise 0.

$\mathbf{U}_{\text{HS}}$ is inspired by the *indirect object search* strategy as studied in [53, 178]. Previous studies

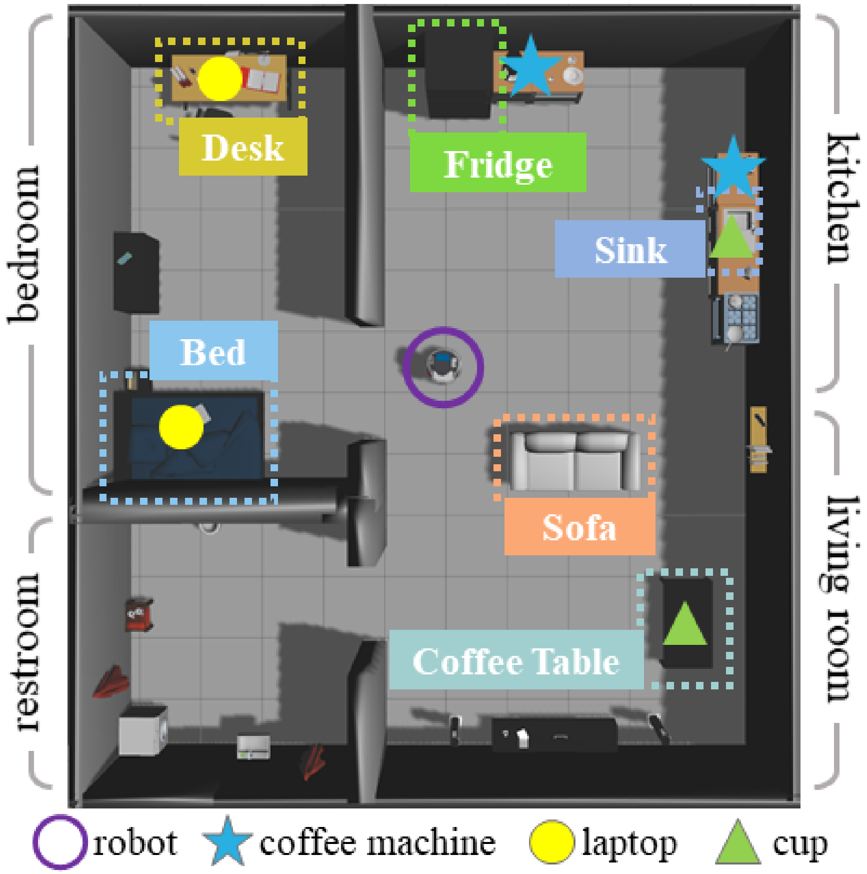


Figure 5.5: Simulation experiments setup in Gazebo: an apartment-like environment with four rooms. There are 6 landmark objects and 3 target objects: *coffee machine, laptop, cup*. Each target object has two equally possible locations.

demonstrated that purposefully looking for an intermediate landmark object helps quickly narrow down the search region for the target object if the landmark object often co-occurs with the target object, thus improving the search efficiency.

With U_{HS}, the object search can be considered **hybrid** because we are considering promising areas represented by GMMs for both the target object (as in direct search) and landmark objects that co-occur with the target object (as in indirect search).

In our experiments, we use a A^* based planner to compute d_{nav}. We empirically set $\alpha = 0.1$, $\beta = 0.4$, and $\sigma = 0.5$ such that $\arctan(\sigma d_{nav})$ plateaus as d_{nav} goes beyond $3m$.

5.6 Experiments

We perform object search tasks in both simulation and real-world environments with a Fetch robot. In the simulation experiments, we quantitatively benchmark various methods, including methods that resemble previous works and our proposed method. In the real-world experiments, we demonstrate qualitatively that the proposed method scales to real-world applications. In both simulation and real-world experiments, the robot accelerates to at most 1m/s and turns at most at 1.7rad/s.

For the simulation experiments, we focus on evaluating the effectiveness of the proposed active object search approach in unfamiliar environment, i.e., only common sense knowledge of inter-object spatial relation is given when modeling the probabilistic inter-object spatial relations, and no long-term or short-term memory is available.

For the real-world experiments, we examined the proposed active object search approach both in unfamiliar and familiar environment. In addition, when long-term and short-term memory become available, we demonstrated the advantage of modeling these types of memories as part of the generalized object permanence, in comparison with object search without modeling these types of memories. Our implementation of the proposed inference algorithm runs at 50 FPS on average (for 5 objects case) on a computer with i7 2.60GHz CPU. And view pose selection step takes on average 0.5 second once robot updates the belief over objects locations. We used a MPEPC based controller proposed by Park et al. [129] to navigate robot from one pose to the next best view pose,

5.6.1 Simulation Experiments in Unfamiliar Environments

The simulation experiments are performed in an apartment-like environment (10mx11m) setup in the Gazebo simulator, as shown in Figure 5.5. The room types and considered landmark objects are annotated in Figure 5.5, along with the placements of target objects. The marginal belief R_{ij} inferred from the factor graph as explained in 5.4.2 is depicted in Figure 5.6.

We set up an object detector in simulation that returns a detection of an object, if the object is in view, not fully occluded, and within the effective observation range. For large objects (e.g. sofa,

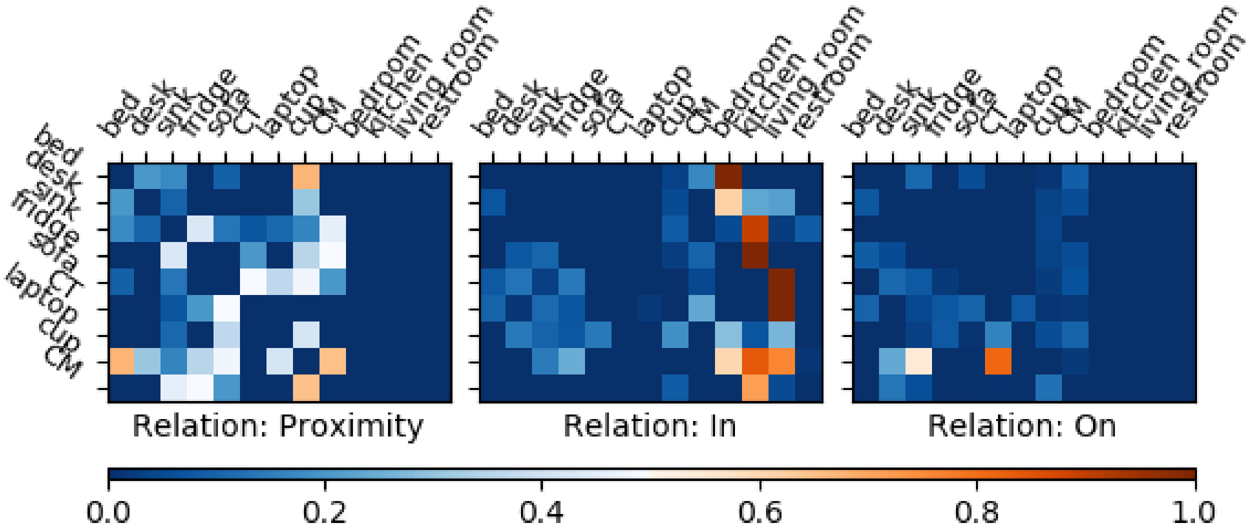


Figure 5.6: Marginal belief on inter-object spatial relations, as well as object-room relations, inferred from the factor graph as explained in Sec. 5.4.2. *CM*: coffee machine, *CT*: coffee table

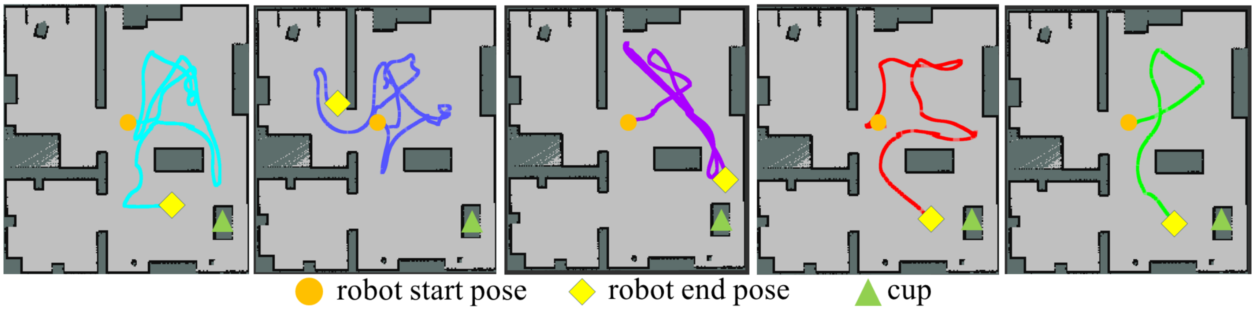


Figure 5.7: Examples of search paths generated by each method while searching for *cup*. Methods from left to right: UDS, IDS-Known-Static, IDS-Known-Dynamic, IDS-Unknown, IHS-Unknown. (Best viewed in color).

bed, fridge), mid-sized objects (e.g. desk, table, sink), and small objects (e.g. cup, laptop, coffee machine), we assume an effective observation range of 5, 4m, 2.5m respectively.

We benchmark following methods:

- **UDS**: Uninformed direct search (Eq.5.6). The robot does not account for the spatial relations between the target and landmark objects (omitting Eq. 5.3 in *SLiM*). This baseline represents a naive approach for object search.
- **IDS-Known-Static**: Informed direct search (Eq.5.6) with a known prior on landmark object locations. The robot assumes that landmark objects are static at the locations provided by

Target Object	Metrics	UDS	IDS known, static	IDS known, dynamic	IDS unknown	IHS unknown
Coffee Machine	Views	7.83	6.17	4.67	6.33	**3.67**
	Search Time (s)	107	76	60	75	**50**
	Search Path (m)	8.68	6.70	5.80	6.74	**4.93**
	Success Rate	**1.0**	**1.0**	**1.0**	**1.0**	**1.0**
Laptop	Views	11.00	12.50	7.17	5.67	**4.17**
	Search Time (s)	197	222	124	91	**78**
	Search Path (m)	28.27	26.86	13.13	**7.69**	8.40
	Success Rate	0.83	0.50	**1.00**	**1.00**	**1.00**
Cup	Views	13.17	14.50	12.67	11.83	**9.00**
	Search Time (s)	184	229	189	185	**139**
	Search Path (m)	22.64	29.81	23.40	19.68	**13.91**
	Success Rate	0.83	0.33	0.83	0.83	**1.00**

Table 5.1: Benchmark results for object search in simulation experiments. Among methods that reached 100% success rate, IHS unknown successfully found target objects within the smallest number of views and least search time.

the prior. This method resembles previous works [78, 90, 167].

- **IDS-Known-Dynamic**: Informed direct search (Eq.5.6) with a known prior on landmark object locations. This is similar to IDS-Known-Static except that the robot does not assume the landmark objects to remain at the locations expressed in the prior.

- **IDS-Unknown**: Informed direct search (Eq.5.6) without prior on landmark object locations. The particles for landmark objects are initialized uniformly across the environment. This method resembles previous works [103, 12].

- **IHS-Unknown**: Informed hybrid search (Eq.5.5.2.2) without prior on landmark object locations.

All methods except for UDS are using the full *SLiM* model. We assume that an occupancy-grid map of the environment is given. We also assume that the room types are accurately recognized across the environment. IDS-Known-* methods are provided with a noisy prior on landmark object locations which differ from the actual locations, to emulate the common cases where perfect knowledge about landmark locations is not available. For all methods, the particles for the target object are initialized uniformly across the environment.

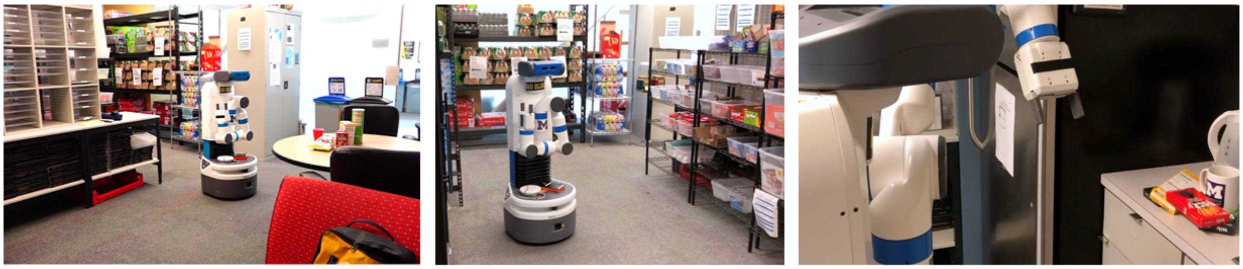


Figure 5.8: An unfamiliar environment that consists of kitchen and a living room.

For each target object, we run 6 trials per method. In each trial, the robot starts at the same location, depicted in Figure 5.5. The object search is terminated if (1) the belief over the target object location has converged or (2) the maximum search time of 5mins has been exceeded. A trial is successful if the robot finds the target object before timeout.

The benchmark result is as shown in Table 5.1. Examples of the resulting search path from each method are depicted in Figure 5.7. Per target object, we measure the average search success rate of each method. Among the successful trials, we measure the average number of view poses, average time taken, average distance travelled. As we can see, UDS is not as effective because it is not making use of the spatial relations between the target objects and landmark objects in the environment. Given a noisy prior on landmark object locations, IDS-Known-Dynamic outperforms IDS-Known-Static because it accounts for the uncertainty of the landmark object locations, whereas IDS-Known-Static is misled by the noisy prior.

Given no prior information, IHS-unknown outperforms IDS-unknown because it encourages the robot to explore promising regions that contain the target and/or useful landmark objects, whereas IDS-unknown only considers promising regions that contain the target object. With IHS-unknown, the robot benefits from finding landmark objects which help narrow down the search region for the target object.

5.6.2 Real-World Experiments in Unfamiliar Environments

The real-world experiment is executed in an environment (8mx8m) that consists of a kitchen and a living room, as shown in Figure 5.8. The target object is a cup. Landmark objects are a table,

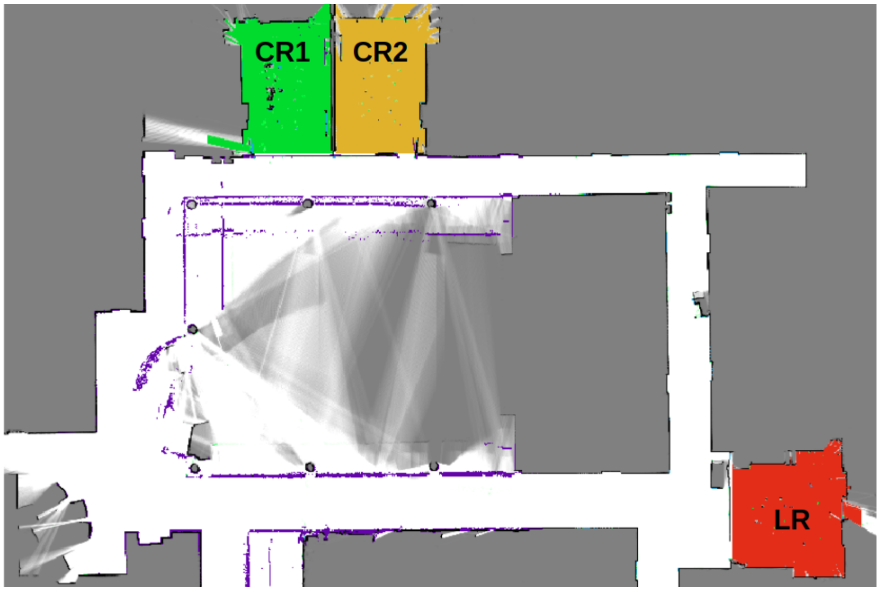


Figure 5.9: Map of our environment. Green: Conference Room 1 (CR1); Yellow: Conference Room 2 (CR2); Red: Lounge Room (LR).

a sofa, a coffee machine and a sink. We run 10 trials of object search using IHS-Unknown. The average success rate is 0.7 (7 out 10 trials), the average number of view poses is 4.86, the average search time is 103s, and the average search path travelled is 8.32m. The failure cases were due to false negative detection of the cup. Examples of real-world experiments with a Fetch robot is included in this video[1].

5.6.3 Real-World Experiments in Familiar Environments

To evaluate our method in familiar environments where the robot had accumulated past observations of surrounding environments, we perform multiple experiments on searching for various target objects at a floor level in a building. All our experiments are performed using a *fetch* robot in an environment with pre-mapped environment in 2D occupancy grid as shown in Figure 5.9. The environment involves three rooms, which are a lounge room (LR) and two conference rooms (CR1, CR2), as shown in Figure 5.10. The conference rooms are located besides each other, the aerial distance between them and the lounge measures 27 meters. We used

[1] https://youtu.be/uWWJ5aV6ScE

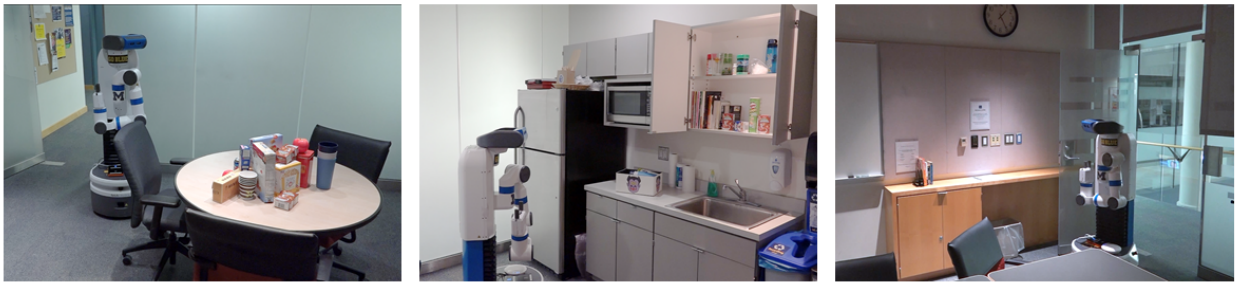


Figure 5.10: A familiar environment that consists of a lounge room and conference rooms.

$\mu = 12, w_{LT} = 50, w_{CC} = 50, w_{ST} = 150$ across our experiments. A video of our experiments is available here[2].

5.6.3.1 Experiment Setup

In our experiments, we task the robot with finding one of five target objects as listed below. The objects are placed throughout the environment according to manually designed underlying location distributions that is hidden from our system. Different objects engage different levels of uncertainty in their location distributions, making some objects almost stationary and others highly uncertain. Here is a brief overview of the manually designed underlying location distributions:

- **Pringles can**: Located exclusively in a cabinet in LR
- **Popcorn box**: Exclusively on table in LR.
- **Coke can**: Always located in LR. Often located on table (usually in a box on table). Less often in box on the counter by the sink. Rarely directly on the counter.
- **DVD**: Equally distributed between counter tops in CR1 and CR2.
- **TV Remote**: Mostly located on table in LR, rest of the time on counter tops in either CR1 or CR2.

Aside from the target objects, the table in the lounge is moved locally frequently as well, as tends to happen in a shared space like a lounge. To collect past observations of the environment

[2] https://youtu.be/h2lqP8fewJk

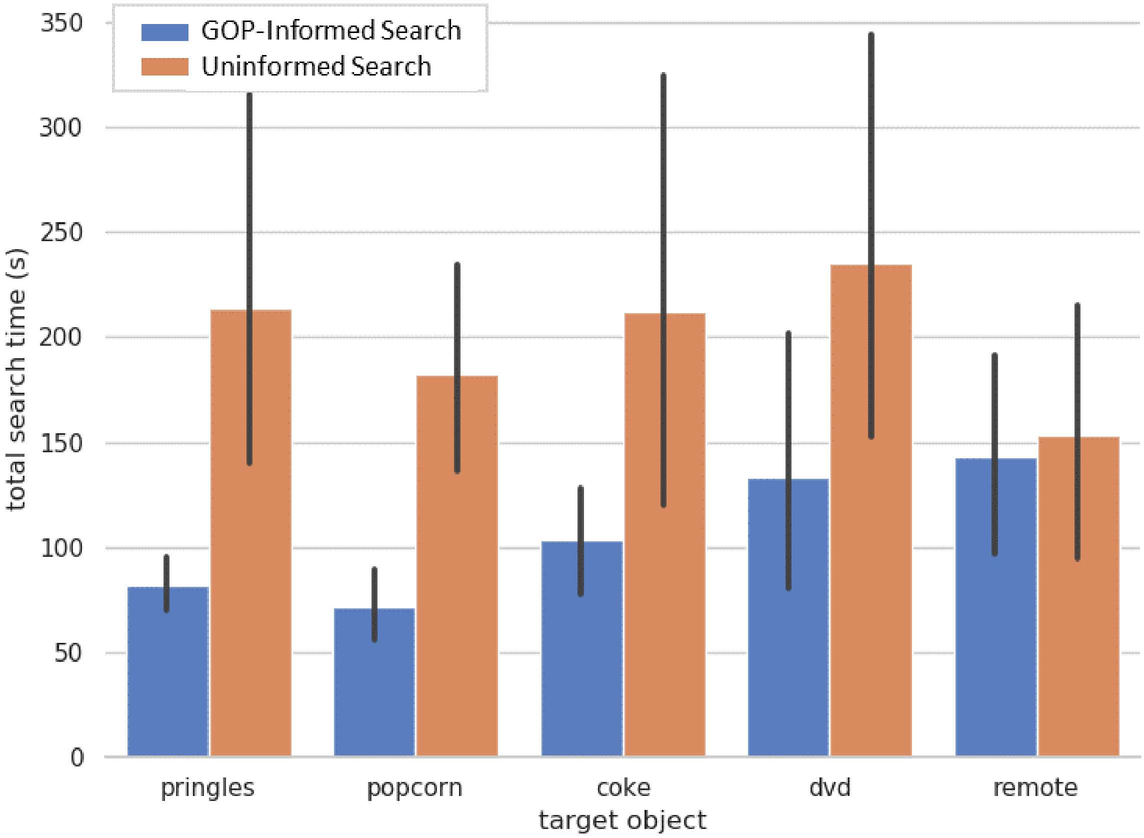


Figure 5.11: *GOP-Informed Search* v.s. *Uninformed Search*. *y*–axis: average search time, with 95% confidence interval marked. *x*–axis: target objects (arranged in the order of increasing uncertainty of their underlying distributions from left to right).

as used in our method, we drove the robot around the floor to visit CR1, CR2, LR multiple times as the objects vary their locations based on their underlying distributions. The robot detects and localizes the objects during these multiple visits based on [189], and derives the spatial relations from their estimated 6 DOF poses based on simple geometric heuristics.

We carried out three sets of experiments to respectively highlight the benefits of incorporating *long term memory*, *short-term memory* and scene graph structure during search. For each set of experiments, we benchmark the search performance in terms of search time of our GOP-Informed method against different baselines. All baselines use the same perception and control component as our main method. We evaluate the observation controller qualitatively over the experiments.

5.6.3.2 GOP-Informed Search v.s. Uninformed Search

In this set of experiments, we examine the benefit of incorporating *long term memory* during search. The baseline method *Uninformed Search* models the possible metric locations of target objects with GMMs purely based on common contextual relations. Because all rooms can potentially contain the target object, as a result, the baseline method has equally weighted GMMs spread across all rooms. At search time, the baseline method initially randomly selects a Gaussian component in the GMMs for active perception, and then continues to perceive other Gaussian components based on distance until the target object is found.

We carried out search experiments for all 5 target objects respectively. For each target object, 10 individual search trials were conducted for our method and the baseline. For each trial, target objects were placed in the environment following the underlying distribution as explained in 5.6.3.1. The average search time for all target objects is as shown in Figure 5.11. As we can see, *GOP-Informed Search* outperforms the *Uninformed Search* for all target objects. For target objects with low uncertainty in their underlying distributions (e.g. *pringle*, *popcorn*), our method significantly benefits from modeling the *long term memory* that captures frequent patterns in the scene structure based on past observations. As the uncertainty of the underlying distributions of object locations increases, the average search time of our method increases and approaches the baseline's performance for target objects such as *remote*.

5.6.3.3 GOP-Informed Search v.s. GOP-Informed Search w/o Short-Term Memory

In this set of experiments, we look into the benefit of incorporating *short-term memory* during search. The baseline method *GOP-Informed Search w/o ST* is the same as *GOP-Informed Search* except that no *short-term memory* is not being modeled.

We carried out search experiments with the target object being *coke*. To examine the benefit of modeling the *short-term memory*, we let the robot observe one of the following inter-object relations around 5 minutes before the search starts: A. *on(coke, counter)*, B. *on(box, counter)*. Case A reflects a direct observation of the relation of *coke* and other object, case B reflects an

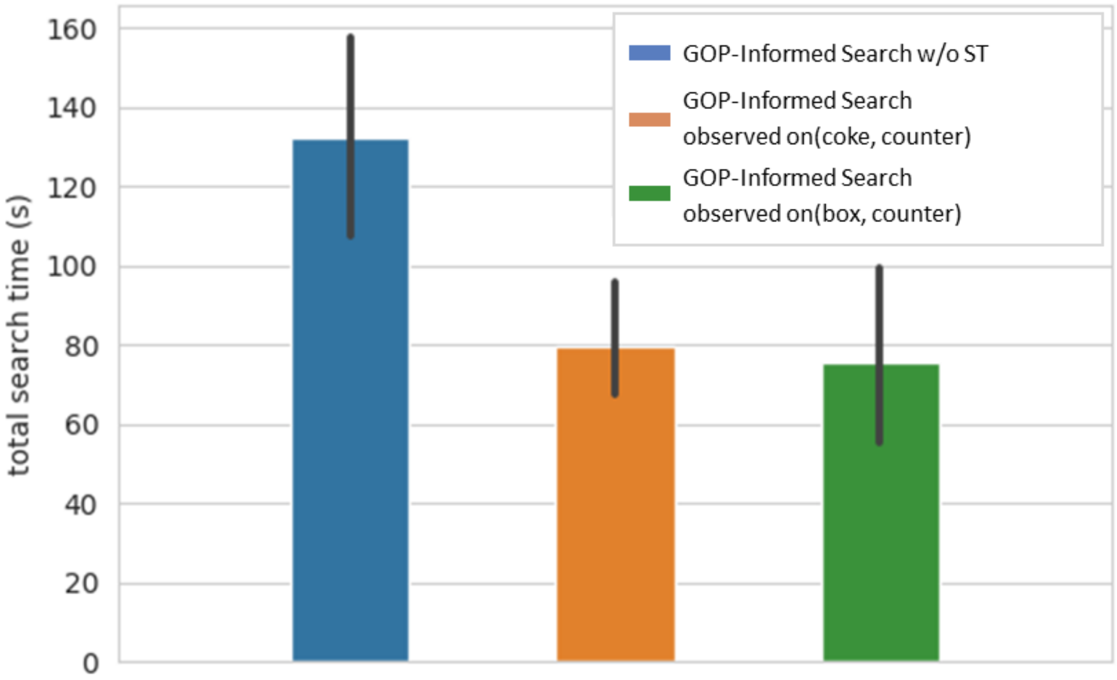
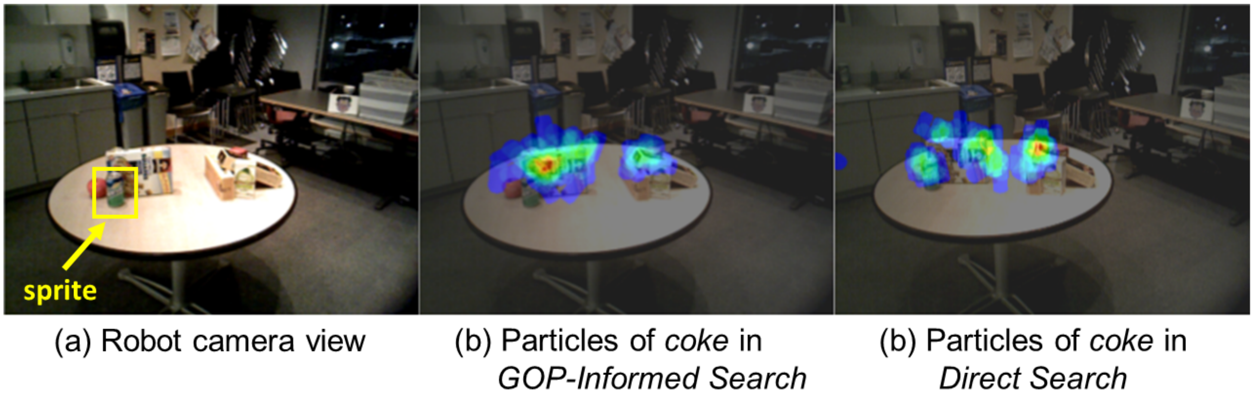


Figure 5.12: *GOP-Informed Search* v.s. *GOP-Informed Search w/o ST*. *y*–axis: average search time, with 95% confidence interval marked.

observation of an object that is highly correlated with *coke* (*coke* is often located in the box).

We conducted 10 individual search trials for *GOP-Informed Search w/o ST* and *GOP-Informed Search* under case A and B. For each trial, we sample the places to put *coke* based on a decaying probability (in Equation 5.4) of *coke* (or *box*) staying on the counter, with a ground truth value of $\mu = 16$ which is hidden from our system. If *coke* (or *box*) is sampled to be moved from the counter, then the new location of *coke* is sampled based on objects underlying distribution as explained in 5.6.3.1 and the co-occurrences of *coke* and *box*.

The average search time is as shown in Figure 5.12. As we can see, when directly observed a certain relation between the target object and other objects (case A), *GOP-Informed Search* outperforms *GOP-Informed Search w/o ST*. The baseline method was mainly driven by *long-term memory* and *common contextual relation*, leading the robot to perceive the table which is usually where *coke* is semantically located. Even when the robot did not directly observed *coke*, but observed *box* to be on counter, *GOP-Informed Search* still significantly outperforms the baseline method *GOP-*



(b) Particles of *coke* in *GOP-Informed Search*
(b) Particles of *coke* in *Direct Search*

Figure 5.13: When searching for *coke*, robot observes *sprite* that co-occur often with *coke*. *GOP-Informed Search* weighs the particles around the *sprite* more (warmer color means more particles) compared to the *direct search* baseline. Best viewed in color.

Informed Search w/o ST. This is because the proposed factor graph captures the strong relation *in(coke, box)* and propagates the information of *on(box, counter)* to *on(coke, counter)* through the scene consistency factor.

5.6.3.4 GOP-Informed Search v.s. Direct Search

In this set of experiments, we seek to examine the benefit of using scene graph structure to guide the search process. The baseline method *Direct Search* does not make sure of any notion of scene graph structure, instead it directly metrically models the distribution of the target object locations. The metrical distribution of target object locations are approximated as GMMs from the past observations. As a result, the baseline method directly tries to localize the target object in the environment, without first localizing its parent objects.

We carried out search experiments for target object being *coke*. 10 individual trials were conducted for *GOP-Informed Search* and *Direct Search*. The target object was placed in the environment following the underlying distribution as explained in 5.6.3.1. The average search time for our method is 154.16s ± 33.83s, and *Direct Search* method achieved 201.43s ± 119.73s. The reason why *GOP-Informed Search* outperforms the baseline *Direct Search* with less standard deviation is mainly because of following factors:

- *GOP-Informed Search* narrows down the search region for the target object by first localizing its parent objects in the inferred scene graph. When the environment is cluttered, and the variance of the object metric location distribution is large, the robot might end up spending extra efforts observing other occupied regions rather than observing the parent object area that is more likely to have the target object.

- *GOP-Informed Search* also considers possible neighbor objects that could co-occur with target objects as part of the inferred scene graph. During the search, object poses particles around the neighbor objects are weighted more than others. This is because that we account for co-occurrences as part of the inter-object spatial relations. Thus the updated belief of object locations motivates the robot to actively perceive the corresponding area with higher confidence. The benefit is as highlighted in Figure 5.13.

5.7 Conclusion

We present an effective active object search approach through the introduction of GOP and the *SLiM* model. We model GOP through a factor graph that accounts for long-term, short-term memory and common sense knowledge on inter-object spatial relations. *SLiM* simultaneously maintains the belief over target object locations as well as landmark object locations, while accounting for the probabilistic inter-object spatial relations between all object pairs modeled as GOP. Further, we propose a hybrid search strategy that draws insights from both direct and indirect object search. With quantitative experiments in simulation, we demonstrate the benefit of accounting for uncertainty in landmark object locations when a noisy prior on their locations is given. When no prior on landmark objects is given, we demonstrate that the proposed hybrid search strategy outperforms a direct search strategy by encouraging the robot to explore areas that are promising and contain not only the target object but also landmark objects. We also show the proposed object search approach operating in real-world experiments.

CHAPTER 6

Discussion and Conclusion

6.1 Conclusion

This dissertation introduces Semantic Robot Programming (*SRP*) as a declarative approach to the problem of robot programming from workspace demonstrations. *SRP* enables an intuitive modality of interaction for end users to program robots by directly demonstrating goal scenes.

By bridging semantic mapping with robot PbD, we have enabled end users to program robots under perceptual uncertainty in cluttered scenes. With *SRP*, we show a Fetch robot successfully reproducing user intended goals with generalization across various initial state of the world. *CT-Map* further enhances the ability of robots to learn a task at a large scale by semantically mapping room scaled environments from streaming observations. We demonstrated *CT-Map* outperforming state-of-the-art neural network based object detectors and commonly adopted 3D registration method for localizing objects in a clutter. By modeling GOP and efficiently maintaining the belief of objects via *SLiM*, robots are able to reason about objects that are not being directly observed. When performing a user intended task at large scale, we have enabled robots to search for objects needed for the task much more efficiently than state-of-the-art methods.

In sum, this dissertation has coined semantic mapping and robot PbD as semantic robot programming, and provided a declarative approach to robot PbD that is generalizable to different initial world states, robust to perceptual uncertainty in clutter, and efficient for tasks at large scale. This dissertation provides interesting future pathways to pursue towards ultimate interactive task learning in robot PbD.

6.1.1 Future Works

There are many future directions that are worth further investigation. First, only one user demonstration of the goal scene is provided and the robot will reproduce a task that overfits to unintentional scene structures in the demonstration. For example, the demonstrated goal scene might contain non-task related objects. Secondly, the robot only reproduces the axiomatic spatial relations between objects as user demonstrated, regardless of the exact relative metric poses between objects in the demonstration. However, the user might desire specific relative poses between objects. For example, the user would desire the fork to be placed on the left side of a plate, and the knife to be on the right side of the plate, instead of any poses of the fork or the knife that satisfies the proximity relation to the plate. To address above issue, the robot can take in multiple user demonstrations and determine 1) task related objects; 2) the correct expressiveness level (metric or axiomatic) of a particular edge between two object nodes in the scene graph. Interacting with the user for feedback can also help disambiguate the goal of the task.

To extend the task domain beyond tasks that are only concerned about inter-object spatial relations, such as cooking tasks, we need to 1) extend the scene graph representation to incorporate other object states such as temperature, fill-level of a container object, and visual appearances in addition to object poses; 2) extend the sensor modality to observe the object states from various channels; 3) extend the action library beyond pick and place actions with affordance templates [58]. We have started exploring a new way to represent objects called Affordance Coordinate Frame (ACF) [186] to generalize robotic manipulation to novel object instances. Instead of representing one object with its 3D geometry model and pose, we propose to represent an object with its parts, and compatibility between parts. Multiple ACFs can be registered to one object part. Manipulation policies can be defined with respect to each ACF for robot to make use of the corresponding affordance of the object.

As the robot performs a user demonstrated task, it is important to monitor the progress of the task and react to unexpected events. We can develop interactive perception [19] approaches to monitor the scene state, and also action prediction models for detecting unexpected events, such that the robot can robustly perform the task based on both reactive controller and re-planning when needed.

Another particular interesting direction is to generalize a user demonstrated goal across different object instances. For example, given multiple demonstrations of organizing a coffee table in a living room with different object instances involved, the robot should be able to perform the same organization task when object instances different from the ones in demonstrations are involved. As a way to approach this problem, a similarity measurement between scene graphs can be developed for user demonstrated task. The robot can then auto-propose scene graphs constructed from the object instances present at execution time, and choose one scene graph for performing the task based on the similarity measurements between the proposed and the demonstrated scene graphs.

BIBLIOGRAPHY

[1] P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the twenty-first international conference on Machine learning*, page 1. ACM, 2004.

[2] N. Abdo, H. Kretzschmar, L. Spinello, and C. Stachniss. Learning manipulation actions from a few demonstrations. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 1268–1275. IEEE, 2013.

[3] S. R. Ahmadzadeh, A. Paikan, F. Mastrogiovanni, L. Natale, P. Kormushev, and D. G. Caldwell. Learning symbolic representations of actions from human demonstrations. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3801–3808. IEEE, 2015.

[4] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz. Keyframe-based learning from demonstration. *International Journal of Social Robotics*, 4(4):343–355, 2012.

[5] B. Akgun and A. Thomaz. Simultaneously learning actions and goals from demonstration. *Autonomous Robots*, 40(2):211–227, 2016.

[6] A. Aldoma, Z.-C. Marton, F. Tombari, W. Wohlkinger, C. Potthast, B. Zeisl, R. B. Rusu, S. Gedikli, and M. Vincze. Point cloud library. *IEEE Robotics & Automation Magazine*, 1070(9932/12), 2012.

[7] A. Aldoma, F. Tombari, R. B. Rusu, and M. Vincze. Our-cvfh–oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. In *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*, pages 113–122. Springer, 2012.

[8] S. Alexandrova, Z. Tatlock, and M. Cakmak. Roboflow: A flow-based visual programming language for mobile manipulation tasks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 5537–5544. IEEE, 2015.

[9] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5):469–483, 2009.

[10] A. Aydemir, A. Pronobis, M. Göbelbecker, and P. Jensfelt. Active visual object search in unknown environments using uncertain semantics. *IEEE Transactions on Robotics*, 29(4):986–1002, 2013.

[11] A. Aydemir, K. Sjöö, J. Folkesson, A. Pronobis, and P. Jensfelt. Search in the real world: Active visual object search based on spatial relations. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 2818–2824. IEEE, 2011.

[12] A. Aydemir, K. Sjöö, and P. Jensfelt. Object search on a mobile robot using relational spatial information. In *Proceedings of International Conference on Intelligent Autonomous Systems*, pages 111–120, 2010.

[13] S. Y. Bao, M. Bagra, Y.-W. Chao, and S. Savarese. Semantic structure from motion with points, regions, and objects. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2703–2710. IEEE, 2012.

[14] P. Beeson, J. Modayil, and B. Kuipers. Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy. *The International Journal of Robotics Research*, 29(4):428–459, 2010.

[15] M. Beetz, M. Tenorth, and J. Winkler. Open-ease. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1983–1990. IEEE, 2015.

[16] G. Biggs and B. MacDonald. A survey of robot programming systems. In *Proceedings of the Australasian conference on robotics and automation*, pages 1–3, 2003.

[17] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. *Handbook of Robotics*, chapter Robot programming by demonstration. Springer, Secaucus, NJ, USA, 2008.

[18] A. Billard, S. Calinon, R. Dillmann, and S. Schaal. Robot programming by demonstration. In *Springer handbook of robotics*, pages 1371–1394. Springer, 2008.

[19] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[20] N. Bore, P. Jensfelt, and J. Folkesson. Multiple object detection, tracking and long-term dynamics learning in large 3d maps. *arXiv preprint arXiv:1801.09292*, 2018.

[21] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic slam. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1722–1729. IEEE, 2017.

[22] S. Brandi, O. Kroemer, and J. Peters. Generalizing pouring actions between objects using warped parameters. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 616–621. IEEE, 2014.

[23] R. Brooks. A robust layered control system for a mobile robot. *IEEE journal on robotics and automation*, 2(1):14–23, 1986.

[24] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.

[25] J. Butterfield, S. Osentoski, G. Jay, and O. C. Jenkins. Learning from demonstration using a multi-valued function regressor for time-series data. In *2010 10th IEEE-RAS International Conference on Humanoid Robots*, pages 328–333, Dec 2010.

[26] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298, 2007.

[27] D. J. Cannon. Point-and-direct telerobotics: Object level strategic supervisory control in unstructured interactive human-machine system environments. 1992.

[28] C. Chao, M. Cakmak, and A. L. Thomaz. Towards grounding concepts for transfer in goal learning from demonstration. In *2011 IEEE International Conference on Development and Learning (ICDL)*, volume 2, pages 1–6. IEEE, 2011.

[29] S. Chernova and M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34(1):1, 2009.

[30] M. J.-Y. Chung, M. Forbes, M. Cakmak, and R. P. Rao. Accelerating imitation learning through crowdsourcing. In *ICRA*, pages 4777–4784, 2014.

[31] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*, pages 241–252. Springer, 2014.

[32] J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel. Towards semantic slam using a monocular camera. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 1277–1284. IEEE, 2011.

[33] M. Colledanchise and P. gren. How behavior trees modularize hybrid control systems and generalize sequential behavior compositions, the subsumption architecture, and decision trees. *IEEE Transactions on Robotics*, 33(2):372–389, April 2017.

[34] C. Crick, S. Osentoski, G. Jay, and O. C. Jenkins. Human and robot perception in large-scale learning from demonstration. In *2011 6th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 339–346, March 2011.

[35] R. Cubek and W. Ertel. Conceptual similarity as a key to high-level robot programming by demonstration. In *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, pages 1–6. VDE, 2012.

[36] H. Dang and P. K. Allen. Robot learning of everyday object manipulations via human demonstration. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1284–1289. IEEE, 2010.

[37] M. P. Deisenroth, G. Neumann, J. Peters, et al. A survey on policy search for robotics. *Foundations and Trends® in Robotics*, 2(1–2):1–142, 2013.

[38] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation (ICRA 1999)*, May 1999.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[40] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins. Efficient nonparametric belief propagation for pose estimation and manipulation of articulated objects. *Science Robotics*, 4(30):eaaw4523, 2019.

[41] R. G. Dromey. From requirements to design: Formalizing the key steps. In *First International Conference onSoftware Engineering and Formal Methods, 2003. Proceedings.*, pages 2–11. IEEE, 2003.

[42] S. Ekvall, P. Jensfelt, and D. Kragic. Integrating active mobile robot object recognition and slam in natural environments. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pages 5792–5797. IEEE, 2006.

[43] S. Ekvall and D. Kragic. Robot learning from demonstration: a task-level planning approach. *International Journal of Advanced Robotic Systems*, 5(3):33, 2008.

[44] J. Elfring, S. Jansen, R. van de Molengraft, and M. Steinbuch. Active object search exploiting probabilistic object–object relations. In *Robot Soccer World Cup*, pages 13–24. Springer, 2013.

[45] P. Espinace, T. Kollar, N. Roy, and A. Soto. Indoor scene recognition by a mobile robot through adaptive object detection. *Robotics and Autonomous Systems*, 61(9):932–947, 2013.

[46] M. A. T. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (3):381–396, 2002.

[47] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3-4):189–208, 1971.

[48] R. E. Fikes and N. J. Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *Artificial intelligence*, 2(3):189–208, 1972.

[49] C. Finn, T. Yu, T. Zhang, P. Abbeel, and S. Levine. One-shot visual imitation learning via meta-learning. *arXiv preprint arXiv:1709.04905*, 2017.

[50] A. Fod, M. J. Matarić, and O. C. Jenkins. Automated derivation of primitives for movement classification. *Auton. Robots*, 12(1):39–54, Jan. 2002.

[51] K. French, S. Wu, T. Pan, Z. Zhou, and O. C. Jenkins. Learning behavior trees from demonstration. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7791–7797, May 2019.

[52] C. Galindo, A. Saffiotti, S. Coradeschi, P. Buschka, J.-A. Fernandez-Madrigal, and J. González. Multi-hierarchical semantic maps for mobile robotics. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 2278–2283. IEEE, 2005.

[53] T. D. Garvey. Perceptual strategies for purposive vision. Tech. Rep. AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025., 1976.

[54] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[55] D. H. Grollman and O. C. Jenkins. Incremental learning of subtasks from unsegmented demonstration. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 261–266. IEEE, 2010.

[56] S. Guadarrama, L. Riano, D. Golland, D. Go, Y. Jia, D. Klein, P. Abbeel, T. Darrell, et al. Grounding spatial relations for human-robot interaction. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1640–1647. IEEE, 2013.

[57] P. Harris. Development of search and object permanence during infancy. *Psychological Bulletin*, 82(3):332, 1975.

[58] S. Hart, P. Dinh, and K. Hambuchen. The affordance template ROS package for robot task programming. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6227–6234. IEEE, 2015.

[59] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, J. Wyatt, D. Hebesberger, T. Kortner, et al. The strands project: Long-term autonomy in everyday environments. *IEEE Robotics & Automation Magazine*, 24(3):146–156, 2017.

[60] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *European conference on computer vision*, pages 30–43. Springer, 2008.

[61] E. Herbst, P. Henry, and D. Fox. Toward online 3-d object segmentation and mapping. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3193–3200. IEEE, 2014.

[62] J. Huang and M. Cakmak. Code3: A system for end-to-end programming of mobile manipulator robots for novices and experts. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, pages 453–462. ACM, 2017.

[63] A. J. Ijspeert, J. Nakanishi, and S. Schaal. Movement imitation with nonlinear dynamical systems in humanoid robots. In *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, volume 2, pages 1398–1403. IEEE, 2002.

[64] M. Isard. Pampas: Real-valued graphical models for computer vision. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003.

[65] O. C. Jenkins and M. J. Matarić. A spatio-temporal extension to Isomap nonlinear dimension reduction. In *The International Conference on Machine Learning (ICML 2004)*, pages 441–448, Banff, Alberta, Canada, Jul 2004.

[66] O. C. Jenkins, M. J. Mataric, S. Weber, et al. Primitive-based movement classification for humanoid imitation. In *Proceedings, First IEEE-RAS International Conference on Humanoid Robotics (Humanoids-2000)*, pages 1–18, 2000.

[67] N. Jetchev, T. Lang, and M. Toussaint. Learning grounded relational symbols from continuous data for abstract reasoning. 2013.

[68] Y. Jiang, M. Lim, and A. Saxena. Learning object arrangements in 3d scenes using human context. *arXiv preprint arXiv:1206.6462*, 2012.

[69] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on pattern analysis and machine intelligence*, 21(5):433–449, 1999.

[70] D. Joho and W. Burgard. Searching for objects: Combining multiple cues to object locations using a maximum entropy model. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 723–728, May 2010.

[71] L. P. Kaelbling and T. Lozano-Pérez. Integrated task and motion planning in belief space. *The International Journal of Robotics Research*, page 0278364913484072, 2013.

[72] C. C. Kemp, C. D. Anderson, H. Nguyen, A. J. Trevor, and Z. Xu. A point-and-click interface for the real world: laser designation of objects for mobile manipulation. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*, pages 241–248. IEEE, 2008.

[73] P. Khandelwal, S. Zhang, J. Sinapov, M. Leonetti, J. Thomason, F. Yang, I. Gori, M. Svetlik, P. Khante, V. Lifschitz, et al. Bwibots: A platform for bridging the gap between ai and human–robot interaction research. *The International Journal of Robotics Research*, 36(5-7):635–659, 2017.

[74] S. M. Khansari-Zadeh and A. Billard. Learning stable nonlinear dynamical systems with gaussian mixture models. *IEEE Transactions on Robotics*, 27(5):943–957, 2011.

[75] O. Khatib. The potential field approach and operational space formulation in robot control. In *Adaptive and Learning Systems*, pages 367–377. Springer, 1986.

[76] J. Kirk, A. Mininger, and J. Laird. Learning task goals interactively with visual demonstrations. *Biologically Inspired Cognitive Architectures*, 18:1–8, 2016.

[77] J. Kober, J. A. Bagnell, and J. Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[78] T. Kollar and N. Roy. Utilizing object-object and object-scene context when planning to find things. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 2168–2173. IEEE, 2009.

[79] J. Z. Kolter, P. Abbeel, and A. Y. Ng. Hierarchical apprenticeship learning with application to quadruped locomotion. In *Advances in Neural Information Processing Systems*, pages 769–776, 2008.

[80] G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot learning from demonstration by constructing skill trees. *The International Journal of Robotics Research*, page 0278364911428653, 2011.

[81] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *Advances in neural information processing systems*, pages 244–252, 2011.

[82] P. Kormushev, S. Calinon, and D. G. Caldwell. Robot motor skill coordination with em-based reinforcement learning. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 3232–3237. IEEE, 2010.

[83] P. Kormushev, D. N. Nenchev, S. Calinon, and D. G. Caldwell. Upper-body kinesthetic teaching of a free-standing humanoid robot. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3970–3975. IEEE, 2011.

[84] I. Kostavelis and A. Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86–103, 2015.

[85] O. Kroemer, C. Daniel, G. Neumann, H. Van Hoof, and J. Peters. Towards learning hierarchical skills for multi-phase manipulation tasks. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 1503–1510. IEEE, 2015.

[86] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Transactions on information theory*, 47(2):498–519, 2001.

[87] B. Kuipers. The spatial semantic hierarchy. *Artificial intelligence*, 119(1-2):191–233, 2000.

[88] D. Kulić, C. Ott, D. Lee, J. Ishikawa, and Y. Nakamura. Incremental learning of full body motion primitives and their sequencing through human motion observation. *The International Journal of Robotics Research*, 31(3):330–345, 2012.

[89] L. Kunze, M. Beetz, M. Saito, H. Azuma, K. Okada, and M. Inaba. Searching objects in large-scale indoor environments: A decision-theoretic approach. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4385–4390. Citeseer, 2012.

[90] L. Kunze, K. K. Doreswamy, and N. Hawes. Using qualitative spatial relations for indirect object search. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 163–168. IEEE, 2014.

[91] J. E. Laird, K. Gluck, J. Anderson, K. D. Forbus, O. C. Jenkins, C. Lebiere, D. Salvucci, M. Scheutz, A. Thomaz, G. Trafton, R. E. Wray, S. Mohan, and J. R. Kirk. Interactive task learning. *IEEE Intelligent Systems*, 32(4):6–21, 2017.

[92] J. E. Laird, A. Newell, and P. S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial intelligence*, 33, 1987.

[93] A. X. Lee, A. Gupta, H. Lu, S. Levine, and P. Abbeel. Learning from multiple demonstrations using trajectory-aware non-rigid registration with applications to deformable object manipulation. In *IROS*, pages 5265–5272, 2015.

[94] I. Lenz, H. Lee, and A. Saxena. Deep learning for detecting robotic grasps. *The International Journal of Robotics Research*, 34(4-5):705–724, 2015.

[95] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[96] J. K. Li, D. Hsu, and W. S. Lee. Act to see and see to act: Pomdp planning for objects search in clutter. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 5701–5707. IEEE, 2016.

[97] B. Limketkai, D. Fox, and L. Liao. Crf-filters: Discriminative particle filters for sequential state estimation. In *Robotics and Automation (ICRA), 2007 IEEE International Conference on*, pages 3142–3147. IEEE, 2007.

[98] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[99] P. Lindes, A. Mininger, J. R. Kirk, and J. E. Laird. Grounding language for interactive task learning. In *Proceedings of the First Workshop on Language Grounding for Robotics*, pages 1–9, Vancouver, Canada, Aug. 2017. Association for Computational Linguistics.

[100] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[101] Z. Liu, D. Chen, K. M. Wurm, and G. von Wichert. Table-top scene analysis using knowledge-supervised mcmc. *Robotics and Computer-Integrated Manufacturing*, 33:110–123, 2015.

[102] S.-Y. Lo, S. Zhang, and P. Stone. Petlon: Planning efficiently for task-level-optimal navigation. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS 18, page 220228, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[103] P. Loncomilla, J. Ruiz-del Solar, and M. Saavedra. A bayesian based methodology for indirect object search. *Journal of Intelligent & Robotic Systems*, 90(1-2):45–63, 2018.

[104] M. Lorbach, S. Höfer, and O. Brock. Prior-assisted propagation of spatial information for object search. In *Intelligent Robots and Systems (IROS), 2014 IEEE/RSJ International Conference on*, pages 2904–2909. IEEE, 2014.

[105] A. Marzinotto, M. Colledanchise, C. Smith, and P. Ögren. Towards a unified behavior trees framework for robot control. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5420–5427. IEEE, 2014.

[106] M. J. Mataric. Behaviour-based control: Examples from navigation, learning, and group behaviour. *Journal of Experimental & Theoretical Artificial Intelligence*, 9(2-3):323–336, 1997.

[107] J. McCarthy and P. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. L. Webber and N. J. Nilsson, editors, *Readings in Artificial Intelligence*, pages 431 – 450. Morgan Kaufmann, 1981.

[108] J. McCormac, A. Handa, A. Davison, and S. Leutenegger. Semanticfusion: Dense 3d semantic mapping with convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4628–4635. IEEE, 2017.

[109] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet rgb-d: 5m photorealistic images of synthetic indoor trajectories with ground truth. *arXiv preprint arXiv:1612.05079*, 2016.

[110] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL-the planning domain definition language. 1998.

[111] S. J. McKenna and H. Nait-Charif. Tracking human motion using auxiliary particle filters and iterated likelihood weighting. *Image and Vision Computing*, 25(6):852–862, 2007.

[112] D. Meger, M. Muja, S. Helmer, A. Gupta, C. Gamroth, T. Hoffman, M. Baumann, T. Southey, P. Fazli, W. Wohlkinger, et al. Curious george: An integrated visual search platform. In *Computer and Robot Vision, 2010 Canadian Conference on*, pages 107–114. IEEE, 2010.

[113] S. Mohan and J. Laird. Learning goal-oriented hierarchical tasks from situated interactive instruction. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[114] S. Mohan, A. H. Mininger, J. R. Kirk, and J. E. Laird. Acquiring grounded representations of words with situated interactive instruction. In *Advances in Cognitive Systems*. Citeseer, 2012.

[115] J. M. Mooij. libdai: A free and open source c++ library for discrete approximate inference in graphical models. *Journal of Machine Learning Research*, 11(Aug):2169–2173, 2010.

[116] Y. Munakata. Infant perseveration and implications for object permanence theories: A pdp model of the ab task. *Developmental Science*, 1(2):161–184, 1998.

[117] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[118] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, and M. Kawato. Learning from demonstration and adaptation of biped locomotion. *Robotics and Autonomous Systems*, 47(2):79–91, 2004.

[119] V. Narayanan and M. Likhachev. Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances. In *Robotics: Science and Systems*, 2016.

[120] V. Narayanan and M. Likhachev. Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances. In *Proceedings of Robotics: Science and Systems*, AnnArbor, Michigan, June 2016.

[121] V. Narayanan and M. Likhachev. Discriminatively-guided deliberative perception for pose estimation of multiple 3d object instances. In *Robotics: Science and Systems*, June 2016.

[122] A. Y. Ng, S. J. Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[123] H. Nguyen, M. Ciocarlie, K. Hsiao, and C. C. Kemp. Ros commander (rosco): Behavior creation for home robots. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 467–474. IEEE, 2013.

[124] M. N. Nicolescu and M. J. Mataric. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 241–248, 2003.

[125] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5239–5246. IEEE, 2012.

[126] S. Niekum, S. Osentoski, G. Konidaris, S. Chitta, B. Marthi, and A. G. Barto. Learning grounded finite-state representations from unstructured demonstrations. *The International Journal of Robotics Research*, 34(2):131–157, 2015.

[127] A. Paraschos, C. Daniel, J. R. Peters, and G. Neumann. Probabilistic movement primitives. In *Advances in neural information processing systems*, pages 2616–2624, 2013.

[128] D.-H. Park, H. Hoffmann, P. Pastor, and S. Schaal. Movement reproduction and obstacle avoidance with dynamic movement primitives and potential fields. In *Humanoids 2008-8th IEEE-RAS International Conference on Humanoid Robots*, pages 91–98. IEEE, 2008.

[129] J. J. Park, C. Johnson, and B. Kuipers. Robot navigation with model predictive equilibrium point control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4945–4952. IEEE, 2012.

[130] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE, 2009.

[131] C. Paxton, G. D. Hager, L. Bascetta, et al. An incremental approach to learning generalizable robot tasks from human demonstration. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 5616–5621. IEEE, 2015.

[132] C. Paxton, A. Hundt, F. Jonathan, K. Guerin, and G. D. Hager. Costar: Instructing collaborative robots with behavior trees and vision. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 564–571. IEEE, 2017.

[133] S. Pillai and J. Leonard. Monocular slam supported object recognition. *arXiv preprint arXiv:1506.01732*, 2015.

[134] N. Ratliff, B. Ziebart, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa. Inverse optimal heuristic control for imitation learning. In *Artificial Intelligence and Statistics*, pages 424–431, 2009.

[135] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[136] S. Reed and N. De Freitas. Neural programmer-interpreters. *arXiv preprint arXiv:1511.06279*, 2015.

[137] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[138] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2017.

[139] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. IEEE, 2009.

[140] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2155–2162. IEEE, 2010.

[141] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[142] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1352–1359. IEEE, 2013.

[143] C. Sammut, S. Hurst, D. Kedzier, and D. Michie. Learning to fly. In *Machine Learning Proceedings 1992*, pages 385–393. Elsevier, 1992.

[144] M. Scheutz, G. Briggs, R. Cantrell, E. Krause, T. Williams, and R. Veale. Novel mechanisms for natural human-robot interactions in the diarc architecture. In *Workshops at the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[145] M. Scheutz, R. Cantrell, and P. Schermerhorn. Toward humanlike task-based dialogue processing for human robot interaction. *Ai Magazine*, 32(4):77–84, 2011.

[146] M. Scheutz, P. Schermerhorn, J. Kramer, and D. Anderson. First steps toward natural human-like hri. *Autonomous Robots*, 22(4):411–423, 2007.

[147] O. Sener and A. Saxena. rcrf: Recursive belief estimation over crfs in rgb-d activity videos. In *Proceedings of Robotics: Science and Systems*, July 2015.

[148] K. Shubina and J. K. Tsotsos. Visual search for an object in a 3d environment using a mobile robot. *Computer Vision and Image Understanding*, 114(5):535–547, 2010.

[149] D. Silver, J. A. Bagnell, and A. Stentz. Learning from demonstration for autonomous navigation in complex unstructured terrain. *The International Journal of Robotics Research*, 29(12):1565–1592, 2010.

[150] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[151] K. Sjö, D. G. López, C. Paul, P. Jensfelt, and D. Kragic. Object search and localization for an indoor mobile robot. *Journal of Computing and Information Technology*, 17(1):67–80, 2009.

[152] K. Sjöö, A. Aydemir, and P. Jensfelt. Topological spatial relations for active visual search. *Robotics and Autonomous Systems*, 60(9):1093–1107, 2012.

[153] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel. Combined task and motion planning through an extensible planner-independent interface layer. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 639–646, May 2014.

[154] J. Stückler, B. Waldvogel, H. Schulz, and S. Behnke. Dense real-time mapping of object-class semantics from rgb-d video. *Journal of Real-Time Image Processing*, 10(4):599–609, 2015.

[155] I. A. Sucan and S. Chitta. Moveit! *Online Available: http://moveit. ros. org*, 2013.

[156] I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics & Automation Magazine*, 19(4):72–82, December 2012. http://ompl.kavrakilab.org.

[157] E. B. Sudderth, A. T. Ihler, W. T. Freeman, and A. S. Willsky. Nonparametric belief propagation. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2003.

[158] Z. Sui, O. C. Jenkins, and K. Desingh. Axiomatic particle filtering for goal-directed robotic manipulation. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4429–4436. IEEE, 2015.

[159] Z. Sui, Z. Zhou, Z. Zeng, and O. C. Jenkins. Sum: Sequential scene understanding and manipulation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[160] N. Sünderhauf, T. T. Pham, Y. Latif, M. Milford, and I. Reid. Meaningful maps with object-oriented semantic mapping. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 5079–5085. IEEE, 2017.

[161] A. K. Tanwani and S. Calinon. Learning robot manipulation tasks with task-parameterized semitied hidden semi-markov model. *IEEE Robotics and Automation Letters*, 1(1):235–242, 2016.

[162] K. Tateno, F. Tombari, and N. Navab. When 2.5 d is not enough: Simultaneous reconstruction, segmentation and recognition on dense slam. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 2295–2302. IEEE, 2016.

[163] B. Tekin, S. N. Sinha, and P. Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018.

[164] A. ten Pas and R. Platt. Using geometry to detect grasp poses in 3d point clouds. In *Intl Symp. on Robotics Research*, 2015.

[165] A. Ten Pas and R. Platt. Localizing handle-like grasp affordances in 3d point clouds. In *Experimental Robotics*, pages 623–638. Springer, 2016.

[166] M. Tenorth and M. Beetz. Knowrob: A knowledge processing infrastructure for cognition-enabled robots. *The International Journal of Robotics Research*, 32(5):566–590, 2013.

[167] R. Toris and S. Chernova. Temporal persistence modeling for object search. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 3215–3222. IEEE, 2017.

[168] R. Toris, D. Kent, and S. Chernova. Unsupervised learning of multi-hypothesized pick-and-place task templates via crowdsourcing. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4504–4510. IEEE, 2015.

[169] J. Tremblay, T. To, A. Molchanov, S. Tyree, J. Kautz, and S. Birchfield. Synthetically trained neural networks for learning human-readable plans from real-world demonstrations. *arXiv preprint arXiv:1805.07054*, 2018.

[170] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[171] L. Ureche, K. Umezawa, Y. Nakamura, and A. Billard. Task parameterization using continuous constraints extracted from human demonstrations. *IEEE Transactions on Robotics*, 31(ARTICLE), 2015.

[172] H. Veeraraghavan and M. Veloso. Teaching sequential tasks with repetition through demonstration. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 3*, pages 1357–1360. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[173] M. Veloso, J. Biswas, B. Coltin, and S. Rosenthal. Cobots: robust symbiotic autonomous mobile service robots. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 4423–4429. AAAI Press, 2015.

[174] P. Viswanathan, D. Meger, T. Southey, J. J. Little, and A. K. Mackworth. Automated spatial-semantic modeling with applications to place labeling and informed search. In *Computer and Robot Vision, 2009 Canadian Conference on*, pages 284–291. IEEE, 2009.

[175] M. Vochten, T. De Laet, and J. De Schutter. Generalizing demonstrated motions and adaptive motion generation using an invariant rigid body trajectory representation. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 234–241. IEEE, 2016.

[176] M. Vondrak, L. Sigal, J. Hodgins, and O. Jenkins. Video-based 3d motion capture through biped control. *ACM Transactions On Graphics (TOG)*, 31(4):1–12, 2012.

[177] C. Wang, J. Cheng, J. Wang, X. Li, and M. Q.-H. Meng. Efficient object search with belief road map using mobile robot. *IEEE Robotics and Automation Letters*, 3(4):3081–3088, 2018.

[178] L. E. Wixson and D. H. Ballard. Using intermediate objects to improve the efficiency of visual search. *International Journal of Computer Vision*, 12(2-3):209–230, 1994.

[179] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez. Manipulation-based active search for occluded objects. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2814–2819. IEEE, 2013.

[180] Y. Xiang and D. Fox. Da-rnn: Semantic mapping with data associated recurrent neural networks. *arXiv preprint arXiv:1703.03098*, 2017.

[181] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[182] Y. Xiao, S. Katt, A. ten Pas, S. Chen, and C. Amato. Online planning for target object search in clutter under partial observability. In *Robotics and Automation (ICRA), 2019 International Conference on*, pages 8241–8247. IEEE, 2019.

[183] D. Xu, S. Nair, Y. Zhu, J. Gao, A. Garg, L. Fei-Fei, and S. Savarese. Neural task programming: Learning to generalize across hierarchical tasks. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–8. IEEE, 2018.

[184] Y. Yang, Y. Li, C. Fermüller, and Y. Aloimonos. Robot learning manipulation action plans by "watching" unconstrained videos from the world wide web. In *AAAI*, pages 3686–3693, 2015.

[185] T. Yu, P. Abbeel, S. Levine, and C. Finn. One-shot hierarchical imitation learning of compound visuomotor tasks. *arXiv preprint arXiv:1810.11043*, 2018.

[186] Z. Zeng, P. S. Joshi, and O. C. Jenkins. Unsupervised learning of affordance coordinate frame for robotic task generalization. In *ICRA 2019 workshop: 2nd International Workshop on Computational Models of Affordance in Robotics*, Montral, Canada, 2019.

[187] Z. Zeng and B. Kuipers. Object manipulation learning by imitation. *arXiv preprint arXiv:1603.00964*, 2016.

[188] Z. Zeng, A. Röfer, and O. C. Jenkins. Semantic linking maps for active visual object search. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.

[189] Z. Zeng, Y. Zhou, O. C. Jenkins, and K. Desingh. Semantic mapping with simultaneous object detection and localization. In *Intelligent Robots and Systems (IROS), 2018 IEEE/RSJ International Conference on*, pages 911–918. IEEE, 2018.

[190] Z. Zeng, Z. Zhou, Z. Sui, and O. C. Jenkins. Semantic robot programming for goal-directed manipulation in cluttered scenes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7462–7469. IEEE, 2018.

[191] L. Zhang and J. C. Trinkle. The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing. In *2012 IEEE international conference on robotics and automation (ICRA)*, pages 3805–3812. IEEE, 2012.

[192] Z. Zhao and X. Chen. Semantic mapping for object category and structural class. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 724–729. IEEE, 2014.

[193] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014.