Abstract of "Multimodal Human-Robot Interaction With Decision Theory and Mixed Reality" by David Francis Whitney, Ph.D., Brown University, May 2019.

While humans have physically and cognitively evolved to work alongside and communicate with each other, humans and robots cannot intuit each others behavior. We can neither accurately understand the other's state nor anticipate their future actions. This is for two reasons: one, robots lack a theory of mind; two, the physical modalities (i.e. ways of communicating) of human-human interaction do not translate well to human-robot interaction. The aim of this dissertation is to improve human-robot interaction by combining partially observable Markov decision process (POMDP)-based interaction managers with virtual and augmented reality interfaces. POMDP based interaction systems allow the robot to reason about what it does and does not know, and what the human may or may not know. Virtual and augmented reality addresses the problem of bi-directional communication by creating new interaction channels that can replace hard to replicate human-human modalities. Subtle body language cues or facial expressions that a robot cannot do are replaced with 3D visualizations, either in a virtual scene or are superimposed on the real world. Combining these technologies enables untrained humans to effectively direct, and communicate with, robots.

Abstract of "Multimodal Human-Robot Interaction With Decision Theory and Mixed Reality" by David Francis Whitney, Ph.D., Brown University, May 2019.

While humans have physically and cognitively evolved to work alongside and communicate with each other, humans and robots cannot intuit each others behavior. We can neither accurately understand the other's state nor anticipate their future actions. This is for two reasons: one, robots lack a theory of mind; two, the physical modalities (i.e. ways of communicating) of human-human interaction do not translate well to human-robot interaction. The aim of this dissertation is to improve human-robot interaction by combining partially observable Markov decision process (POMDP)-based interaction managers with virtual and augmented reality interfaces. POMDP based interaction systems allow the robot to reason about what it does and does not know, and what the human may or may not know. Virtual and augmented reality addresses the problem of bi-directional communication by creating new interaction channels that can replace hard to replicate human-human modalities. Subtle body language cues or facial expressions that a robot cannot do are replaced with 3D visualizations, either in a virtual scene or are superimposed on the real world. Combining these technologies enables untrained humans to effectively direct, and communicate with, robots.

Multimodal Human-Robot Interaction With Decision Theory and Mixed Reality

by
David Francis Whitney
B.A. & Sc., McGill University, 2014
Sc. M., Brown University, 2016

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2019

This dissertation by David Francis Whitney is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

Stefanie Tellex, Director

Recommended to the Graduate Council

Date _____ _____

George Konidaris, Reader

Date _____ _____

James Tompkin, Reader

Approved by the Graduate Council

Date _____ _____

Andrew G. Campbell
Dean of the Graduate School

Vita

David Whitney was born in New York City, growing up in nearby Tuxedo Park, NY. He attended McGill University, where he studied Cognitive Science, focusing on Neuroscience and Computer Science, and received a B.A.&Sc. in 2014. David joined the Department of Computer Science at Brown University in Providence, Rhode Island in 2014 as a doctoral student. He earned a Sc.M. in 2016 and completed his Ph.D. under the advisement of Dr. Stefanie Tellex.

Acknowledgements

I would like to thank everyone in my life who has helped me, and curse all those who stood in my way.

I would like to thank Aaron, Adam, Adriana, Alborz, Alden, Alex, Alexa, Andrea, Andrew, Andy, Anthony, Arthur, Ben, Birdy, Bradley, Cam, Carl, Carla, Charlie, Chris, Christian, Corieen, Dad, Dahlia, Danaë, Daniel, Danny, Dave, David, Eden, Eli, Emily, Eric, Ericka, Erik, Gaurav, George, Grace, Hanif, Hannah, Henri, Ifrah, Jack, Jackson, Jake, James, Jason, Jay, Jett, John, Jonathan, Jordan, Julianna, Julie, Justin, Kaiyu, Katherine, Kavosh, Kristy, Kyle, Lawson, Lily, Louis, Macgill, Madeline, Margaux, Mario, Matt, Max, Mel, Michael, Miles, Mom, Morgane, Nakul, Nathaniel, Nicole, Omi, Pauli, Pitr, Preston, Qianqian, Roma, Sam, Samir, Sarah, Stephanie, Stefanie, Steve, Thao, Thomas, Vanya, Veena, Vikram, Will, Xinyu, Zaliqa, and Zoë.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In this chapter I elaborate upon and provide an explanation of the context that motivates my thesis statement:

> Combining partially observable Markov decision process (POMDP)-based interaction managers with mixed reality bidirectional communication channels improves the safety and efficiency of human-robot collaboration.

For humans and robots to collaborate efficiently, we need to be able to communicate effortlessly with them. More specifically, we need to be able to convey information quickly to the robot and to have the robot convey information back in order to reach a productive mutual understanding. Currently speech is a well known and widely used communication channel for facilitating human-robot collaboration, which makes sense. Speech can be rapid, intuitive, and (perhaps most importantly) is the most common method for communication between humans. Home agents like the Amazon Alexa, which have become commonplace household items, demonstrate natural language as a primary communication channel. To an extent these devices have been successful in offering a comfortable user experience. As a conversational agent, however, Alexa remains decidedly primitive.

Alexa is simple because the robots skills are hand-coded rule based systems. Essentially Alexas skills are brittle and inflexible when confused. Moreover this system is expensive in terms of man-hours required, and does not scale to multi-turn interactions well. Consequently 70% of Alexa skills are one-shot [2], or single turn. In order for us to collaborate with our robots, we will need something better.

Operating beyond an overly simple (though easy to use) robot like Alexa, we are also compelled to incorporate more nuanced forms of communication because humans use more than speech. Humans also use body language (which has been found to be the strongest signal in human interaction [1]), deictic gestures, and any number of related modalities.

So why do robots not use these channels for communication? We can break the issue in opposing directions, human to robot communication, and robot to human communication. For human to robot communication, the robot must combine heterogeneous information sources that emit data at

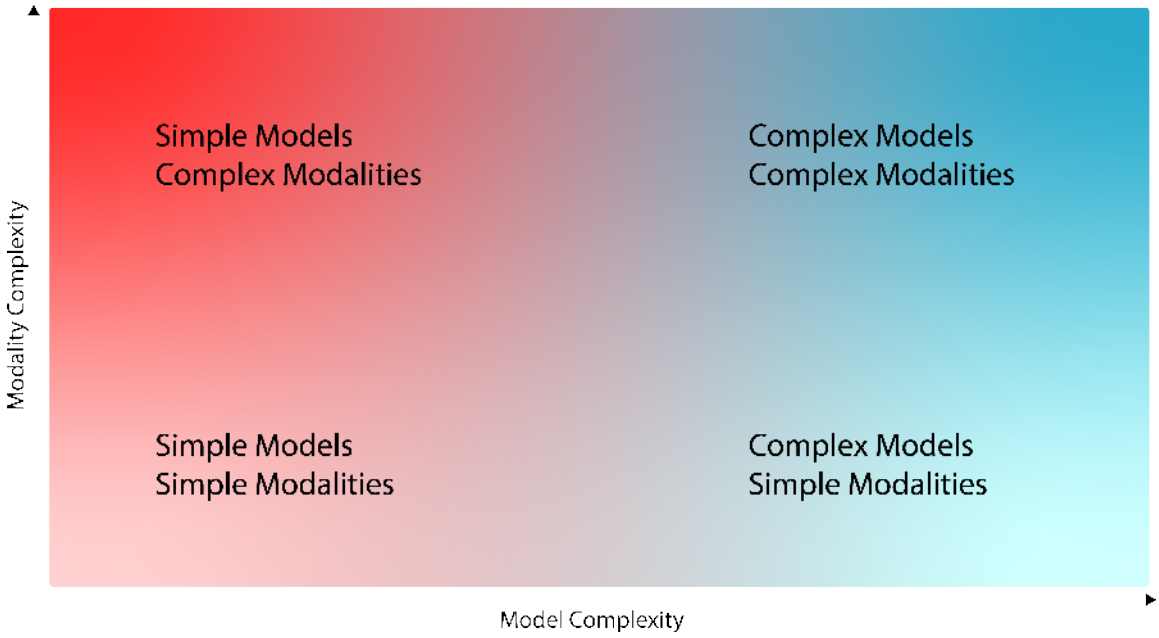


Figure 1.1: Overview of human-robot interaction problem space tackled in this thesis.

different rates. For robot to human communication, we run into a physical problem — most robots lack the physical capability of expressing along those modalities.

Our issues therefore are twofold. We need sophisticated models that can generate complex policy decisions, and need to incoporate many modalities. For a graphical representation of this problem space, see Figure 1.1.

To address these issues, this thesis proposes a family of Partially Observable Markov Decision Processes (POMDPs) which incorporate multimodal interaction and dynamic policy adjustment. Our initial models use speech and gesture, while later systems incorporate mixed reality head-mounted displays, which allow for novel and accurate bidirectional communication between the human and robot.

1.1 Thesis Organization

Each chapter in this thesis builds to our goal of improving human-robot collaboration by extending the capabilities of our model or the modalities it can process. For a graphical representation, see Figure 1.2.

Chapter 3: Interpreting Multimodal Referring Expressions in Real Time

This chapter presents the first component needed to create our POMDP model. It describes state estimation, a probabilistic model for inferring human intent by fusing multiple observation sources into a single probability distribution. This is necessary because humans communicate about objects

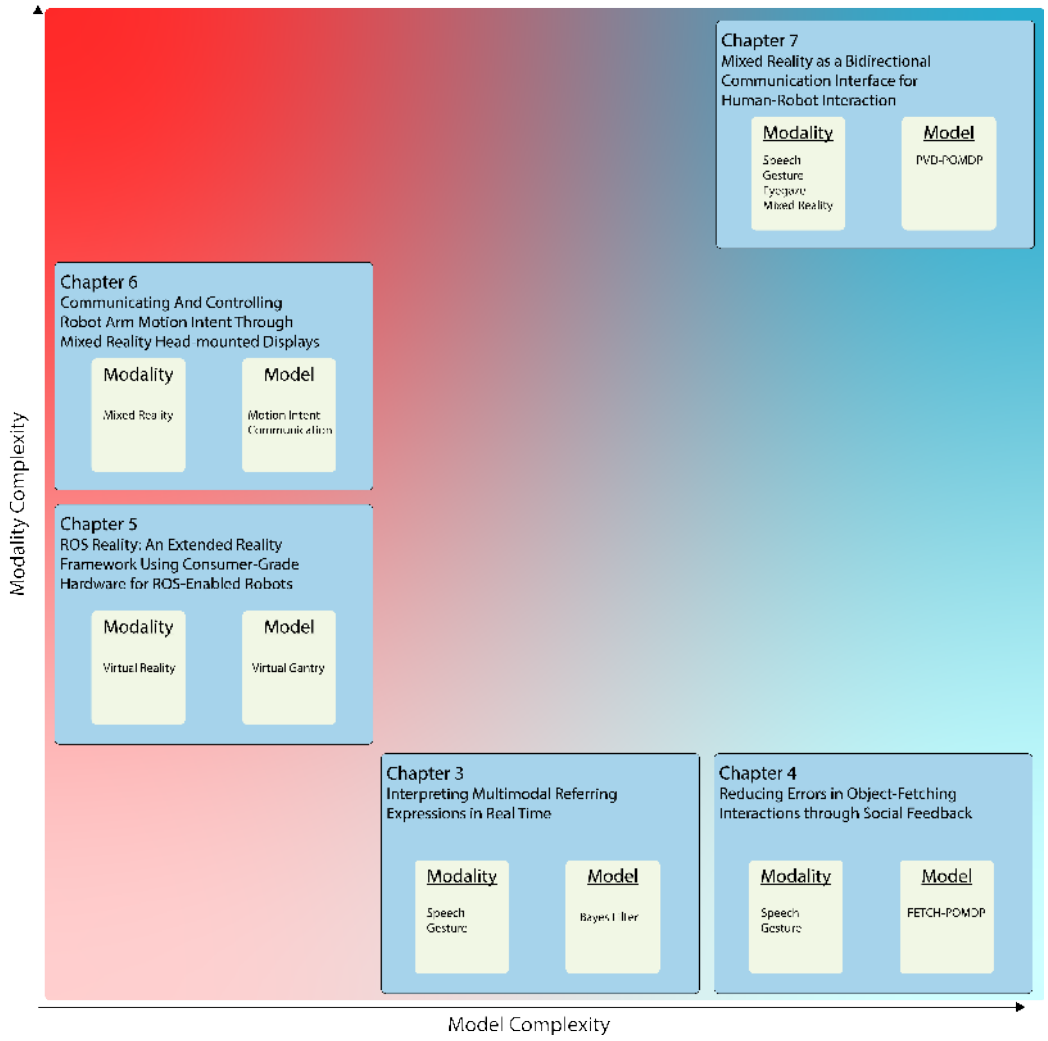


Figure 1.2: Overview of the results of this thesis in the relevant problem space.

using language, gesture, and context, fusing information from multiple modalities over time. Robots must interpret this communication to collaborate with humans on shared tasks. Doing so quickly allows the robot to incorporate the relative timing of words and gestures into the understanding process. To address this problem, we define a multimodal Bayes filter for interpreting referential expressions to objects. Our approach outputs a distribution over the referent object at 14Hz, updating dynamically as it observes new spoken words and gestures. We collect a new dataset of people referring to one of four objects in a tabletop setting and demonstrate that our approach infers the correct object with 90% accuracy. Additionally, we augment and improve our filter in a simulated home kitchen domain by learning contextual knowledge in an unsupervised manner from existing written text, increasing our maximum accuracy to 96%, even increasing in the number of objects from four to seventy.

Chapter 4: Reducing Errors in Object-Fetching Interactions through Social Feedback

This chapter expands on the last by adding robotic agency to the domain. The Bayes filter of Chapter 3 is used as the state estimation portion of a POMDP. This allows the robot to no longer only passively observe the humans observations, but take actions of its own, asking questions to resolve ambiguity. Like the previous chapter, we focus on the object fetching domain. Fetching items is difficult for a social robot: it requires interpreting a person's language and gesture and using these noisy observations to infer what item to deliver. Asking questions would help the robot be faster and more accurate in its task, but existing approaches either do not ask questions or rely on fixed question-asking policies. We propose a model that makes assumptions about cooperation between agents to perform richer signal extraction from observations. This work defines a mathematical framework that allows a robot to increase the speed and accuracy of its ability to interpret a person's requests by reasoning about its own uncertainty and processing implicit information (implicatures). We formalize the item-delivery domain as a Partially Observable Markov Decision Process (POMDP) and approximately solve this POMDP in real time. Our model improves the speed and accuracy of fetching tasks by asking relevant clarifying questions only when necessary. To measure our model's improvements, we conducted a user study with 16 participants. Our method achieved greater accuracy and a faster interaction time compared to state-of-the-art baselines: it is 2.17 seconds faster (25% faster) than a state-of-the-art baseline and 2.1% more accurate.

Chapter 5: Human Robot Interaction via Virtual Reality

This chapter describes ROS Reality, the first open source system that connects any ROS-enabled robot to virtual reality devices. This chapter describes the system, and our experiments to test its efficacy for teleoperation. ROS Reality will be used in later chapters to connect mixed reality devices to our robots, allowing for a much richer set of communication modalities.

Virtual reality (VR) systems let users intuitively interact with 3D environments and have been used extensively for robotic teleoperation tasks. While more immersive than their 2D counterparts, early VR systems were expensive and required specialized hardware. Fortunately, there has been a recent proliferation of affordable consumer-grade VR systems. Our group has designed a VR teleoperation package for the Robot Operating System (ROS), ROS Reality, the first open-source, over-the-Internet teleoperation interface between any ROS-enabled robot and any Unity-compatible VR headset.

We first evaluated our interface on a cup-stacking manipulation task with 18 participants, comparing the relative effectiveness of a keyboard and mouse interface, virtual reality camera control, and positional hand tracking. Our system reduces task completion time from 153 seconds ($\pm$44) to 53 seconds ($\pm$37) (a reduction of 66%), while improving subjective assessments of system usability and workload. Additionally, we show the effectiveness of our system over long distances, successfully completing a cup stacking task from over 40 miles away. Second, we completed a study where

expert human users controlled a Baxter robot via ROS Reality to complete 24 dexterous manipulation tasks, and compared the results to the same users controlling the robot via direct kinesthetic handling. Of the 24 tasks, 16 could be completed via kinesthetic control, and of those 16, 14 could be completed via ROS Reality, a success rate of 87.5%.

Chapter 6: Communicating And Controlling Robot Arm Motion Intent Through Mixed Reality Head-mounted Displays

Now that we have a system that allows a robot to connect to a mixed reality head-mounted display, we investigate using mixed reality as a communication tool for human-robot interaction. This chapter uses the knowledge of effective mixed reality communication in our POMDP model, focusing on motion intent communication. Efficient motion intent communication is necessary for safe and collaborative work environments with co-located humans and robots. Humans efficiently communicate their motion intent to other humans through gestures, gaze, and other non-verbal cues. However, robots often have incredible difficulty using these methods. Existing approaches for robot motion intent communication rely on 2D displays, which require the human to continually pause their work to check a visualization. In this chapter, I propose a mixed reality head-mounted display visualization of the intended robot motion over the wearer's real-world view of the robot and its environment. We described its implementation, which connects a ROS-enabled robot to the HoloLens using ROS Reality, using MoveIt [2] for motion planning, and using Unity to render the visualization. To evaluate the effectiveness of this system against a 2D display visualization and against no visualization, we asked 32 participants to label various arm trajectories as either colliding or non-colliding with blocks arranged on a table. We found a 16% increase in accuracy with a 62% decrease in the time required to complete the task using the next best system. These results demonstrate that a mixed-reality HMD allows a human to determine where the robot is going to move more quickly and accurately than existing methods.

Chapter 7: Mixed Reality as a Bidirectional Communication Interface for Human-Robot Interaction

This chapter presents our POMDP model in its fullest form. It combines multimodal observations, mixed reality communication channels, and dynamic question asking via a flexible interaction policy. We tested the hypothesis that virtual deictic gestures are better for human-robot interaction than physical behaviors. To test this hypothesis, we proposed the Physio-Virtual Deixis Partially Observable Markov Decision Process (PVD-POMDP), which interprets multimodal observations (speech, eye gaze, and pointing gestures) from the human and decides when and how to ask questions (either via physical or virtual deictic gestures) to recover from failure and cope with sensor noise. We conducted a between-subjects user study with 83 participants distributed across three conditions of robot communication: no feedback control, physical feedback, and MR feedback. We tested performance of each condition with objective measures (accuracy, time) and evaluated user

experience with subjective measures (usability, trust, workload). Mixed reality feedback was 10% more accurate than the physical condition with a speedup of 160%. We also found that the feedback conditions significantly outperformed the no feedback condition in all subjective metrics.

Taken together, these results show that incorporating new modalities, when combined with decision theoretic models, improve human-robot collaboration and safety.

Chapter 2

Background and Technical Approach

In this chapter, I define the terms and concepts central to my thesis, and my general technical approach to modeling human-robot collaboration.

2.1 Object Fetching Domain

The main domain of this research is item fetching. I define the problem as a human trying to request items from a robot. The items are spread across a table, and the set is known to both human and robot. The robot's goal is to determine which item the human wants as quickly and accurately as possible, based on different information sources from the user, such as their speech, gesture, or eye gaze. See Figure 2.1 for an example interaction.

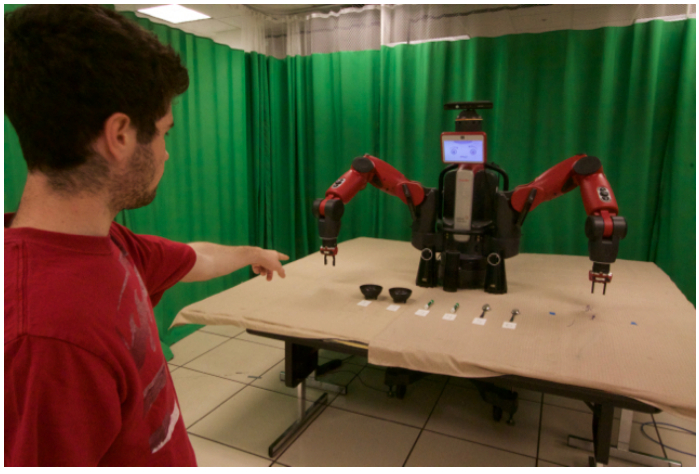


Figure 2.1: An example interaction in our object fetching domain.

2.2 Bayes Filter

A Bayes filter [3], also known as recursive Bayesian estimation, is a probabilistic approach to estimating the current value of a hidden state at time t, $x_t \in \mathcal{X}$, given a history of observations $z_{0:t} \in \mathcal{Z}$. For example, in our domain, you could imagine a scenario where the robot want to know what item a human wants (the hidden state), and at each timestep the robot observes the human's speech and gesture (the observations). For each possible state, the robot will calculate its probability:

$$p(x_t|z_{0:t}). \tag{2.1}$$

To estimate this distribution, we alternate performing a time update and a measurement update. The time update recalculates the belief that the user is referring to a specific object given previous information:

$$p(x_t|z_{0:t-1}) = \sum_{x_{t-1} \in \mathcal{X}} p(x_t|x_{t-1}) \times p(x_{t-1}|z_{0:t-1}). \tag{2.2}$$

The time update includes the transition probability from the previous state to the current state.

The measurement update combines the previous belief with the newest observation to update each belief state:

$$p(x_t|z_{0:t}) = \frac{p(z_t|x_t) \times p(x_t|z_{0:t-1})}{p(z_t|z_{0:t-1})} \tag{2.3}$$

$$\propto p(z_t|x_t) \times p(x_t|z_{0:t-1}) \tag{2.4}$$

These two updates are performed for each timestep.

Bayes filters will appear in Chapters 3, 4, and 7.

2.3 Partially Observable Markov Decision Process

A Markov Decision Process (MDP) [4] is a decision problem formalism in which an agent observes the state of the environment and takes actions in discrete time steps. It is defined by the tuple $\langle S, A, R, T \rangle$, where S is the set of environment states, A is the set of actions, $R(s, a)$ is a reward function that specifies how much instantaneous reward is received for taking action a in state s, and $T(s, a, s')$ is the transition function that defines the probability of the environment transitioning to state s' after the agent takes action a in state s, $p(s'|s, a)$. The goal of the agent is to find an action in any given state (a policy) that maximizes the sum of expected future rewards. In an MDP, it is assumed the agent knows the true state at each timestep. For many problems, this assumption is invalid. Partially Observable Markov Decision Processes (POMDPs) [5] extend MDPs to describe the case when the agent can only indirectly observe the underlying state at each time step from a set

of observations Ω. These observations are modeled as conditionally dependent on the true hidden state by the observation function $O(o, s)$, which defines the probability that the agent will observe observation o in state s, $p(o|s)$.

2.4 Multimodal Social Feedback POMDPs

The main contributions of this thesis are the family of POMDPs which I have named the Multimodal Social Feedback POMDPs, or MSF POMDPs. These POMDPs solve the problem of object grounding. The observations are social signals the human emits, such as speech, pose, and/or eyegaze. The robot will have a terminal action, which signals the robot made a final decision on the object the human was referencing, and also has a set of social feedback actions, like pointing or looking, which are information gathering actions.

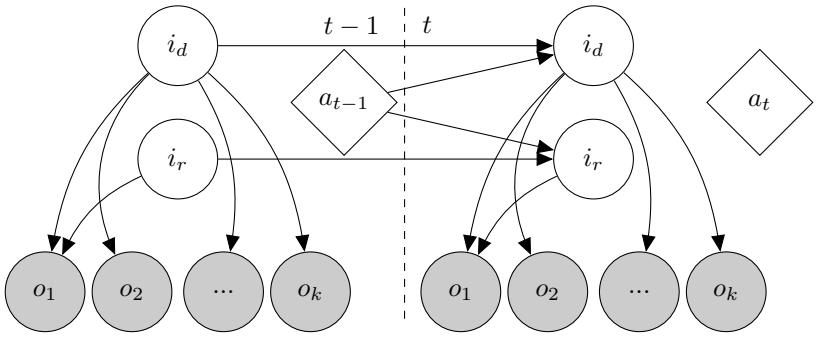


Figure 2.2: A graphical model of a general MSF-POMDP. Hidden variables are white, observed variables are gray. The desired item is i_d, the last item referenced is i_r, the observations are o_1 through o_k and the robot's action is a.

Formally, an MSF-POMDP is specified by the tuple $\langle I, S, A, R, T, O \rangle$.

- I: The list of all items the user could ask for.
- S: $i_d \in I$ is the human's desired item which is hidden. For convenience, we also include the last item the robot referenced (or `null` if none): $i_r \in I \cup \{\texttt{null}\}$.
- A: We categorize actions as social feedback and physical actions. The physical actions consist of a wait action and a parameterized pick(i) action. The wait action merely advances the time-step by one. A pick(i) action finalizes the robot's selection of item i as the user's desired object. The social feedback actions consist of parameterized reference(i) actions. When the robot chooses reference item i, it will indicate item i via some communication modality.
- R: $R(s, a)$: The reward function specifies a large positive reward for picking the correct item, a large negative reward for picking an incorrect item, and smaller negative rewards for wait and the different reference actions.

- T: $T(s, a, s') \equiv p(s' \mid s, a)$ The desired item i_d stays constant until the robot uses a pick action, at which point the interaction terminates. i_d is initially null, and changes to whatever item the robot last referenced.
- O: $O(s, o) \equiv p(o \mid s)$ Observations $o_{1:k}$ will consist of different social signals from the human, such as language, gaze, or gesture.

Chapter 4 describes an instance of an MSF-POMDP, the FEedback-To-Collaborative-Handoff POMDP (FETCH POMDP), and Chapter 7 describes another MSF-POMDP, the Physio-Virtual Deixis POMDP (PVD POMDP).

2.5 Solving POMDPs

While it is intractable to exactly solve all but the simplest POMDPs, approximate solvers can calculate non-optimal solutions in a reasonable amount of time [6].

2.6 Virtual and Mixed Reality

Virtual and mixed reality are different methods of combining digital information with the real world. In virtual reality systems, a user feels completely immersed in a virtual environment. A screen (or other display mechanism) replaces a user's entire field of view with digital imagery, and their head or body is tracked so that their movements match their movements in the virtual world. Virtual reality devices have existed in various forms since the 1960s, but recently entered a renaissance with the release of cheap, high-quality, consumer grade devices like the Oculus Rift and HTC Vive.

Mixed reality, in comparison, is a more complicated term. In 1994, Milgram and Kishino [7] introduced the reality-virtual reality continuum in order to formally define the taxonomy of interfaces that combine the real and virtual (see Fig. 2.3). On one extreme is reality, and on the other is virtuality (aka virtual reality). In between, there are classifications like augmented reality, where digital imagery is superimposed on the real world (e.g. Pokemon Go), and augmented virtuality, where real life objects are superimposed into the virtual world. Milgram and Kishino [7] coined the term mixed reality to refer to all interfaces that fall between the two extremes. Recently, however, the definition of the term has begun to shift. When the Microsoft HoloLens was released in 2015, Microsoft referred to it as a mixed reality headset. According to Microsoft, augmented reality did not capture the full capabilities of their device, which could not just superimpose digital imagery on top the real world, but actually enable the digital content to interact with the real world (e.g. a digital ball could roll off the edge of a table and be occluded) [8]. Magic Leap, another headset maker, adopted a similar stance and stated their devices were distinctly different than augmented reality [9].

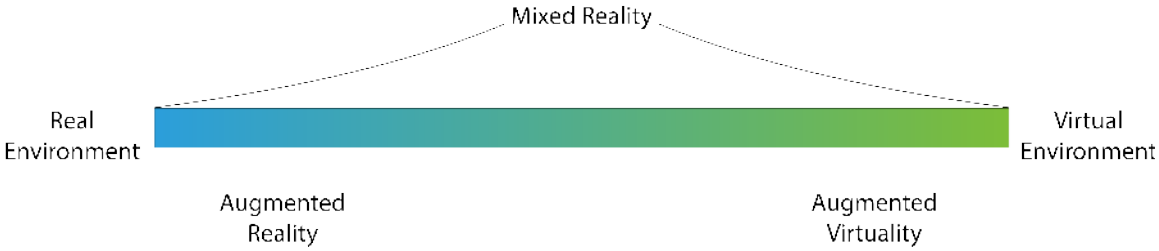


Figure 2.3: The Reality-Virtuality Continuum, as defined by Milgram and Kishino [7]. Note that in their definition, mixed reality comprises all interfaces between a pure real environment and a pure real environment.

Understandably, this has led to confusion. Some people still use the continuum definition, some refer to AR and MR interchangeably, and some use the 'AR+' meaning. For a full analysis of the situation, please see "What is Mixed Reality?', by Speicher et al. [10]. In it, the authors conduct ten expert interviews and a literature review of 68 papers in an attempt to define mixed reality only to conclude: "there is no single definition of MR and it is highly unrealistic to expect one to appear in the future". With all this in mind, I adopt the following definition of mixed reality: Mixed reality, describes interfaces more virtual than augmented reality, and more real than augmented virtuality. The HoloLens and Magic Leap One are mixed reality devices, and interfaces that use them are mixed reality interfaces.

2.7 Virtual and Mixed Reality for HRI

Virtual reality lends itself well to teleoperation interfaces. Chapter 5 describes a virtual reality based teleoperation interface in which the user puts on the VR equipment to share a virtual lab space with the robot. In this space, the human can move around a virtual copy of the robot and reconfigure its end effectors. As they do, the real robots joints match that of the virtual one.

Mixed reality also allows for intuitive communication between a human and robot. The robot can convey information by adding information to the real world without distracting the user from it, and the human can use the sensors in the headset to convey information back. Chapters 6 and 7 describe different mixed reality based human-robot interfaces. Chapter 6 describes a system to communicate motion intent between a human and robot, and Chapter 7 describes a multimodal social feedback POMDP that uses mixed reality as both an observation source and communication channel. These systems combine our theoretical framework with a high-bandwidth bidirectional communication interface, improving the speed and accuracy of achieving mutual understanding.

Chapter 3

Interpreting Multimodal Referring Expressions in Real Time

This chapter presents a probabilistic framework for understanding multimodal referring expressions in real time, which forms the basis of the state estimator for the POMDP described in Chapter 2, especially Section 2.2. The chapter comes from work published at ICRA 2016 [11].

3.1 Introduction

For humans and robots to collaborate in complex tasks, robots must understand people's references to objects in the external world. People provide these signals continuously using language and gesture, and exploit contextual background information to disambiguate requests. Cognitive science experiments have shown that highly successful teams rarely make explicit requests from one another and instead infer the correct actions as needs arise [12]. Responding quickly and incorporating the relative timing of speech and gesture is critical for accurate understanding in human-human interaction [13].

To provide a foundation for these capabilities, we propose a Bayes filtering approach for interpreting multimodal information from language and gesture [3]. Our framework relies on a factored observation probability that fuses information from language and gesture in real time to continuously estimate the object a person is referring to in the real world.

Our approach can also use contextual information, such as the knowledge of which ingredients or tools are likely to be used together, along with language and gesture to disambiguate requests.

In this chapter we focus on a home kitchen domain, generating contextual information in an unsupervised manner by processing an online repository of recipes. Recipes provide semi-structured data that can be automatically mined for contextual information and then combined with language and gesture to interpret a request.

We evaluate our model in simulation and in the real world. In simulation, we use Amazon Mechanical Turk to collect referring expressions and then test those expressions against our system in a simulated kitchen of seventy items. In the real world, we run two separate experiments. In the first, users refer to several objects on a table, switching focus on a fixed schedule. In the second, users interact with a real robot, asking it to hand over one of several items on a table, and only switch once the robot has completed the hand-off.

3.2 Related Work

Language understanding for robots is a very active research topic. While batched interpretation is highly applicable in written communication [14–17], in recent years continuous interpretation has proved more appropriate to real-time domains. Here we will focus largely on works that provide continuous language interpretation.

Kennington and Schlangen [18] created a discriminative model of incremental reference resolution. The authors use wizarded[1] trials of reference resolution to collect training sentences, which they use to train a logistic regression model. Their work is able to correctly resolve sentences, but requires data collection and hand-crafting features. We found our more simple model to be sufficient once combined with gesture in our domain.

Funakoshi et al. [19] also created a model of incremental reference resolution. Like us, their model is based on a Bayesian network design, and can consider different domains for words. In "Bring me the red box, and the blue one, too," their model would understand that "one" refers to the general concept of box. That work focused more on depth in a single modality, where our goal was breadth across multiple modalities.

The majority of gesture systems today focus on gesture recognition [20] which is a classification task that does not require the location or orientation of the gesture [21, 22]. Often, this recognition is performed in batch, and has a slightly different goal, namely to identify times in a video clip in which certain gestures occur. Many approaches to recognition use discriminative models [23, 24], which have been shown to be more accurate than their generative counterparts. The regression-like nature of pointing, however, makes a discriminative approach more difficult. Pointing is not a classification problem, as the goal is a real-valued set of numbers, namely the coordinates in space (x,y,z) the user is pointing to. Our solution is to extend a cone from the wrist of the user, with objects closer to the center ray of the cone considered more likely targets. This approach has been successful before, as shown by Schauerte et al. [25], who extend a cone with a Gaussian distribution from the tip of the arm to identify an object from a still image of a person pointing.

In collaborative robotics, Foster et al. [26] used a rule-based state estimator to deliver drinks from fixed positions behind the bar to multiple users based on their speech and torso position. Similarly, we combine input from multiple modalities, but our work uses a probabilistic approach that smoothly incorporates dynamic gestures such as pointing. Bohus et al. [27] have worked with

[1] Wizarded experiments refer to experiments where a human secretly controls the robot, a la The Wizard of Oz.

a robotic guide that directs users searching for specific rooms in a building. Their model has prior knowledge about the location of the desired rooms, whereas our does not know the location of the desired objects. This system combines the modalities of head direction and speech, but not any other form of gesture.

Matuszek et al. [28] present a multimodal framework for interpreting references to tabletop objects using language and gesture. Our approach similarly focuses on tabletop objects but integrates language and gesture continuously. Additionally, their work has the user sitting at a table, meaning their pointing gestures occur several inches from the referent. In our work, the user stands several feet from the table, making pointing gestures harder to parse.

3.3 Technical Approach

Our aim is to estimate a distribution over the set of objects that a person could refer to, given language and gesture inputs. We frame the problem as a Bayes filter [3], where the hidden state, $x \in \mathcal{X}$, is the object in the scene that the person is currently referencing. The robot observes the person's actions and speech, $\mathcal{Z}$, and at each time step estimates a distribution over the current state, x_t. For more information on Bayes filters, see Sec. 2.2.

3.3.1 Observation Model

The observation model calculates the probability of the observation given the state. Each observation is a tuple describing the user's arm position and speech, $\langle l, r, s \rangle$ where:

- l represents a vector from the elbow (l_o) to the wrist (l_v) of the left arm.
- r represents a vector from the elbow (r_o) to the wrist (r_v) of the right arm.
- s represents the observed speech from the user, consisting of a list of words.

We have an observation model of the form:

$$p(z_t|x_t) = p(l, r, s|x_t). \tag{3.1}$$

We factor our observations assuming that each modality is independent of the others given the state. Namely, we are assuming that if we know the true object, the probabilities of the user pointing at that object with their left hand or right hand are independent:

$$p(z_t|x_t) = p(l|x_t) \times p(r|x_t) \times p(s|x_t). \tag{3.2}$$

The following sections describe how we model each type of input from the person.

Gesture

We model pointing gestures as a vector through three dimensional space. First, we calculate a gesture vector using the skeleton pose returned by the skeleton tracker software NITE [29]. We compute a vector from the elbow to the wrist, then project this vector so that the origin is at the wrist. Next, we calculate the angle between the gesture vector and the vector from the elbow to the center of each object, and then use the PDF of a Gaussian to determine the weight that should be assigned to that object. We define a function $\mathrm{A}(o, p, x)$ as the angle between the point p and the center of mass of object x with the given origin, o. Then

$$p(l|x_t) \propto \mathcal{N}(\mu_l = 0, \sigma_l)[A(l_o, l_v, x_t)] \tag{3.3}$$

$$p(r|x_t) \propto \mathcal{N}(\mu_r = 0, \sigma_r)[A(r_o, r_v, x_t)]. \tag{3.4}$$

While gesture remains a continuous input throughout the entire interaction, many gestures have little or no meaning, such as scratching your nose or crossing your arms. To allow for these without overloading the model with noise, we treat any gesture observation that is greater than some angle θ away from all objects as applying uniform probability to all objects. Mathematically:

$$p(l|x_t) \propto \begin{cases} \frac{1}{|\mathcal{X}|} & \text{if } A(l_o, l_v, x') > \theta \\ & \forall x' \in \mathcal{X} \\ \mathcal{N}(\mu_l = 0, \sigma_l)[A(l_o, l_v, x_t)] & \text{otherwise.} \end{cases} \tag{3.5}$$

The observation for the right arm is calculated in the same way.

Speech

We model speech with a unigram model by taking each word w in a given speech input s and calculating the probability that, given the state x_t, that word would have been spoken:

$$p(s|x_t) = \prod_{w \in s} p(w|x_t). \tag{3.6}$$

To account for words that do not appear in the corpus, we incorporate a uniform epsilon probability for all words that would otherwise have zero probability and then normalize the distribution. When no words are spoken, we assume a null word which has a uniform distribution over the objects. This effect means that spoken words cause a discrete bump in probability according to the language model, which subsequently decays over time as the null word indicates each object equally.

3.3.2 Transition Model

Context is incorporated in our transition model using learned knowledge of related ingredients. In our home kitchen domain, the user requests ingredients for a recipe. Therefore the desired ingredient is the hidden state, and transitions are nonuniform. Recipes generally use ingredients in similar orders.

For example, dry ingredients are used in sequence, or peanut butter follows white bread and grape jelly. With this knowledge we estimate transition probabilities. In other domains, estimates will be more difficult to generate, so we also developed a context-free transition model.

Modeling Non-Contextual Information

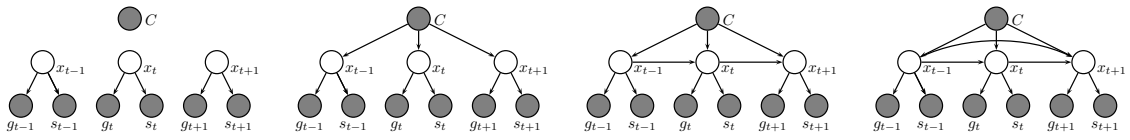
When contextual information is not available, we assume that a person is likely to continue referring to the same object, and at each timestep has some probability, c, of transitioning to the same state:

$$p(x_t|x_{t-1}) = \begin{cases} c & \text{if } x_t = x_{t-1} \\ \frac{1-c}{|\mathcal{X}|-1} & \text{otherwise.} \end{cases} \tag{3.7}$$

This assumption means that the robot's certainty slowly decays over time, in the absence of corroborating information, eventually converging to a uniform distribution. It enables our framework to integrate past language and gesture information but also quickly adapt to new, conflicting information because it assumes the person can switch objects.

Modeling Contextual Information

To model contextual information, we assume that the next object that a person requests depends on the previous object, as well as the information the robot can observe from language and gesture. We empirically calculate transition probabilities by applying language modeling techniques to a large corpus of recipes, C. We consider each recipe as a document, $d_0 \dots d_n \in C$ which contains an ordered list of ingredients, $d_i^0 \dots d_i^k$. We treat this list as an ordered list of states in our model and use it to calculate transition probabilities by mining co-occurrence statistics. Figure 3.1 provides the graphical models for the four approaches we compare in this chapter, using increasing amounts of context to interpret language and gesture.



(a) Uninformed Transitions (no dependency on corpus or previous states).

(b) Unigram model (dependency on corpus, but not previous states).

(c) Bigram model (dependency on corpus as well as one previous state).

(d) Trigram model (dependency on corpus as well as two previous states).

Figure 3.1: Different models for our approach, using increasing amounts of context. Shaded variables are observed.

Our first approach uses a unigram estimator based on individual ingredient frequencies, but does not take the history of past states into account [30–32]. For example, one of the most frequent ingredients is salt, occurring 3.15% of the time in our dataset. The unigram model makes it more likely the robot will fetch the salt but does not incorporate information about previous ingredients.

To compute this model, we count the number of times we observe state x_t compared to the total number of observed states ($x_{0:n}$). Formally:

$$p(x_t|x_{t-1}) = \frac{|\{\forall d_i^k | d_i^k = x_t \in C\}|}{|\{\forall d_i^k \in C\}|}. \tag{3.8}$$

This model gives higher probabilities to more common ingredients, and does not consider past states. A pure unigram estimator would always predict salt as the most likely ingredient.

To incorporate more context we use a bigram model to incorporate one previous state to inform the robot's decision. Formally, we model the probability of the next state, x_t given the previous state, x_{t-1} by counting bigram co-occurrence statistics in the corpus:

$$p(x_t|x_{t-1}) = \frac{|\{\forall d_i^k, d_i^{k+1} \in C | d_i^k = x_{t-1} \wedge d_i^{k+1} = x_t\}|}{|\{\forall d_i^k, d_i^{k+1} \in C\}|}. \tag{3.9}$$

The graphical model for the bigram approach appears in Figure 3.1c. Similarly, we can use two previous states to create a trigram model:

$$p(x_t|x_{t-1}, x_{t-2}) = \\ \frac{|\{\forall d_i^k, d_i^{k+1}, d_i^{k+2} | d_i^k = x_{t-2} \wedge d_i^{k+1} = x_{t-1} \wedge d_i^{k+2} = x_t\}|}{|\{\forall d_i^k, d_i^{k+1}, d_i^{k+2} \in C\}|}. \tag{3.10}$$

The graphical model for the trigram approach appears in Figure 3.1d. While increasing the size of the history adds contextual information, it causes issues with sparseness and compute time, with diminishing returns on accuracy. In our research we found a plateauing of accuracy after trigrams.

Training

Our corpus consists of 42,212 recipes collected from `www.allrecipes.com`. We chose the website for its large collection, varied cuisine, and most importantly, ingredient ordering. All ingredients are listed in the order they are used in the recipe. Each recipe includes a title, the ingredients, the steps, and an end of recipe tag.

The following algorithm applies equations (3.9) and (3.10) to our corpus.

- Given the previously used ingredient (or past two ingredients), for each recipe, iterate through the list of ingredients.
- If a match is found between the input and the current ingredient(s), record the next ingredient in the recipe.
- After scanning all recipes, return the list of ingredients used after the given input, ranked by the number of times they occurred.
- A numerical probability can be constructed by dividing each count by the sum of the counts.

Examples of the top ten unigrams, bigrams and trigrams appear in Table 3.1.

Unigrams	Bigrams	Trigrams
salt	salt, black pepper	all-purpose flour, baking powder, salt
salt	all-purpose flour, baking powder	all-purpose flour, baking soda, salt
white sugar	baking soda, salt	baking powder, baking soda, salt
butter	baking powder, salt	white sugar, eggs, vanilla extract
all-purpose flour	butter, white sugar	all-purpose flour, baking powder, baking soda
water	white sugar, eggs	butter, white sugar, eggs
eggs	all-purpose flour, salt	eggs, vanilla extract, all-purpose flour
garlic	onion, garlic	all-purpose flour, white sugar, baking powder
olive oil	all-purpose flour, baking soda	vanilla extract, all-purpose flour, baking powder
vanilla extract	eggs, vanilla extract	olive oil, onion, garlic

Table 3.1: Top ten ingredient unigrams, bigrams, and trigrams from our training procedure.

3.3.3 Model Parameters

We tuned model hyperparameters by hand. We generated the language model from hand-crafted data combined with the results of our pilot studies. After our initial tuning, we fixed model parameters, and results reported in the chapter all use the same fixed set of parameters. We expect that as we add larger sets of objects, a language model trained using data from Amazon Mechanical Turk or other corpora will be necessary to increase robustness over a larger set of objects.

Our experiments had the following parameters: the uniform transition probability, c, was 0.9995. We set this parameter to give an object that has 100% confidence an approximately 10% drop in confidence per second with all null observations. Standard deviation for the Gaussian used to model probability of gesture, σ_l, σ_r, and σ_h was 1.0 radians. We found that this standard deviation allowed for accurate pointing without skewing the probabilities during an arm swing. The language model consisted of 16 unique words, containing common descriptors for the objects such as "bowl," "spoon," "metal," "shiny," etc. It also included words that were commonly misinterpreted by the speech recognition system, such as "bull" when the user was requesting a bowl.

3.3.4 Algorithm

Algorithm 1 shows pseudocode for our approach, generating a belief distribution over the possible current states $bel(x_t)$, while Figure 3.2 shows an example of the system's execution. The person's speech is ambiguous, and the system initially infers an approximately bimodal distribution between the two bowls. The robot does not hand over any object, which elicits a disambiguating response from the person, who points at the appropriate object. The model incorporates information from language and infers the person is referring to the blue bowl.

Although in this example we are demonstrating the approach at two specific timesteps, the system updates its distribution at 14Hz, enabling it to fuse language and gesture as it occurs and quickly updating in response to new input from the person, verbal or nonverbal. Our approach runs on an 8 2.4 GHz Intel Cores that also performed all perceptual and network processing. This system is used in conjunction with the Baxter Robot and a Kinect V1.

Input: $bel(\mathcal{X}_{t-1}), z_t$
Output: $bel(\mathcal{X}_t)$
for $x_t \in \mathcal{X}_t$ **do**
  $\bar{bel}(x_t) = \sum_{x_{t-1} \in \mathcal{X}_{t-1}} p(x_t|x_{t-1}) * bel(x_{t-1})$
  if not is_null_gesture(l)
    $\bar{bel}(x_t) = p(l|x_t) * \bar{bel}(x_t)$
  if not is_null_gesture(r)
    $\bar{bel}(x_t) = p(r|x_t) * \bar{bel}(x_t)$
  for $w \in s$ **do**
    $\bar{bel}(x_t) = p(w|x_t) * \bar{bel}(x_t)$
  end
  $bel(x_t) = \bar{bel}(x_t)$
end

Algorithm 1: Interactive Bayes Filtering Algorithm

3.4 Evaluation

We evaluated our model through several methods. We first ran simulated trials in our home kitchen domain to detect the efficacy of using contextual information in specific domains. We then ran a system comprehension user study without contextual information to ensure the system's reliability in interpreting referring expressions in a closed environment. Finally, to both show the effectiveness of our model in the real world, as well as demonstrate the ways in which social feedback can play into the model in the future, we ran real world experiments with a robot using this system interacting with human users asking for common kitchenware.[2]

3.4.1 Simulation Results

Next we assess our model's accuracy at inferring ingredients based on a person's requests. Context is most valuable when there are many possible objects that the robot could hand to the person, and we wanted to evaluate our model on a large set of recipes and varied natural language input so we conducted this evaluation using Amazon Mechanical Turk data along with simulated gesture input.

As the number of ingredients the robot interacts with increases, it needs more information to pick the correct one. For example, in a small kitchen there may only be white sugar. The request "hand me the sugar" is unambiguous and the robot easily identifies the correct ingredient. A larger kitchen may have white sugar, brown sugar, and powdered sugar. The request has now become ambiguous, and contextual information becomes necessary to infer the correct object.

[2]Unfortunately, we were unable to test our contextual model in real world. Our contextual simulation study had 70 items in the pantry, and we did not have access to a system that could identify and interact with 70 items at once.

(a) Results using Gesture without Language

Model	d = 3	d = 5	d = 10	d = ∞
Uniform	23.41% ± 1.73%	15.49% ± 1.46%	8.84% ± 1.15%	0.67% ± 0.329%
Unigram	34.82% ± 1.94%	27.74% ± 1.83%	19.21% ± 1.60%	5.43% ± 0.92%
Bigram	**42.74% ± 2.01%**	**35.73% ± 1.94%**	**28.23% ± 1.83%**	12.68% ± 1.34%
Trigram	41.04% ± 1.99%	32.50% ± 1.91%	27.38% ± 1.81%	**12.74% ± 1.35%**

(b) Results Using Gesture with Ambiguous Language Requests

Model	d = 3	d = 5	d = 10	d = ∞
Uniform	74.39% ± 1.78%	70.91% ± 1.84%	67.13% ± 1.91%	47.99% ± 2.03%
Unigram	75.61% ± 1.74%	72.56% ± 1.81%	70.61% ± 1.84%	52.74% ± 2.03%
Bigram	**77.80% ± 1.69%**	**76.22% ± 1.73%**	72.56% ± 1.81%	53.11% ± 2.03%
Trigram	77.38% ± 1.69%	75.12% ± 1.76%	**72.68% ± 1.81%**	**53.72% ± 2.03%**

(c) Results Using Gesture with Ambiguous Language Requests

Model	d = 3	d = 5	d = 10	d = ∞
Uniform	94.63% ± 0.92%	93.96% ± 0.97%	93.41% ± 1.00%	87.50% ± 1.35%
Unigram	95.12% ± 0.87%	94.27% ± 0.94%	**94.39% ± 0.94%**	**89.09% ± 1.27%**
Bigram	**95.67% ± 0.82%**	95.00% ± 0.89%	94.27% ± 0.94%	88.66% ± 1.28%
Trigram	95.55% ± 0.84%	**94.70% ± 0.90%**	**94.39% ± 0.94%**	88.41% ± 1.30%

Table 3.2: Simulated context, language, and gesture results. Errors bounds represent 90% confidence interval.

We presented a series of photos to AMT workers. Each photo contained all the ingredients necessary for a recipe in a kitchen setting. The workers typed requests to the robot. Each worker typed two requests for each ingredient: an ambiguous request, and an unambiguous request. Once the data was collected, the requests were fed as simulated speech to our system. We assessed accuracy by recording whether the system inferred the correct ingredient for each request. We collected a total of 1640 commands over 5 recipes.

Our system had a simulated 'pantry' of objects. The set of ingredients were taken from the cookbook *How to Cook Everything*, under the sections "Kitchen Basics", "Everyday Herbs", and "Everyday Spices" [33]. The ingredients are described as staple ingredients.

Each ingredient in the pantry had several words associated with it. These words were the singular and plural forms of the ingredient's name, and allowed for the observation update to link speech to specific ingredients. For instance, lemon had two words associated with it: lemon and lemons.

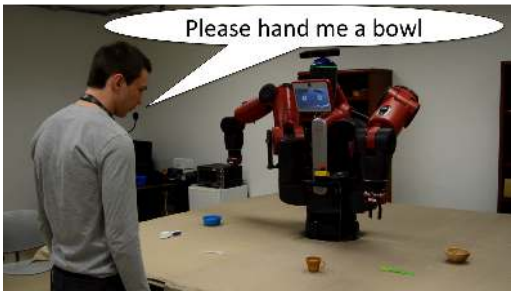
Due to the difficulty of collecting multimodal data for our large dataset, we augmented our system with simulated gesture. We created gesture observations by assuming that a person produced pointing gestures which identified a subset of ingredients, one of which was the one they were asking the robot to fetch. To simulate different amounts of ambiguity in gesture, we varied the size of the cluster, d, between 3, 5, 10, and ∞; here ∞ corresponds to using language only and no gesture.

Tables 3.2a, 3.2b, and 3.2c show an evaluation of our system using uniform, unigram, bigram and trigram models. We report the model's accuracy at identifying the correct object to fetch for each request after the person's natural language input using 90% confidence intervals. First we observe that more specific gestures (with smaller cluster size d) leads to higher model performance. This result is unsurprising because the system has access to significantly more information when augmented with simulated gesture.

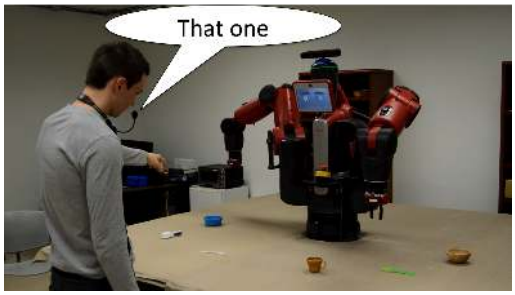
As a high-level trend, we observed a significant increase in performance comparing uniform to unigram. In our unigram model, the robot generates a prior distribution based on common ingredients learned from text, but does not consider objects previously used in the recipe. This model lets us infer the correct action for ambiguous commands such as "fetch the sugar," which most often refers to white sugar rather than brown. This result demonstrates improved performance using information from text in all conditions, but does not integrate contextual information.

Third, we observed a further improvement using the bigram model and trigram model, which use the previous state as context. This performance gain is present under all language conditions, but is increased when commands are ambiguous and decreased for unambiguous commands. Table 3.2c, which uses unambiguous language, shows good performance by all models, including the uniform model which uses no information from text, and a very small positive effect from context. By contrast, Table 3.2a shows results using gestures only, with increasing amounts of ambiguity; here there is a very large improvement from context, going from 23% correct with uniform to 42.7% with the bigram model. In this scenario, gesture provides a strong signal but also contains a large amount of noise; combining this information with context from previous requests significantly improves accuracy.

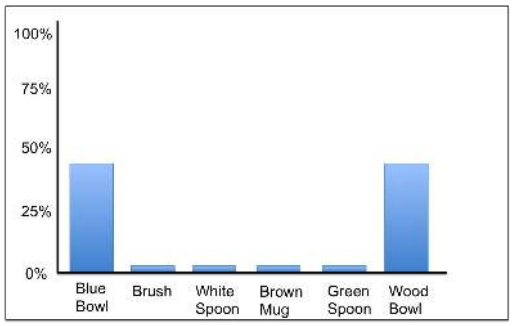
Finally, Table 3.2b shows a modest improvement from context. We expect to see a larger gain with more ambiguous language. In our data, many requests were ambiguous because of spatial language not capable of being understood by our approach. For example, a request such as "Please hand me the onion beside the garlic" would be ambiguous to our system because it cannot process spatial referring expressions. This provides an opportunity for context to disambiguate, but since both ingredients are used similarly, the contextual models cannot determine what the user desires. In our data, many such examples occurred because images showed all ingredients for the same recipe. Despite the limitations of the language data collected on AMT, we still observed a modest improvement from context in this type of language. For instance, in one trial a user requested soy sauce by stating "get me the soy sauce it is next to the garlic." The unigram model estimated the user wanted garlic, as garlic is used more often than soy sauce, but the bigram model looked at the last used ingredient, coconut milk, and calculated soy sauce was used more often than garlic in that context. The trigram model plateaued relative to the bigram model, most likely due to issues of sparsity in the training data.



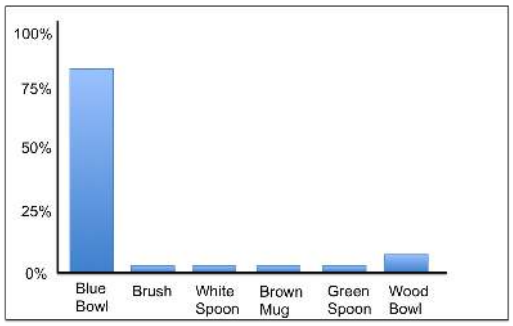
(a) Ambiguous speech.



(b) Clarification with gesture.



(c) State estimate during ambiguous speech.



(d) State estimate after clarification.

Figure 3.2: After an ambiguous spoken request (a), the model has a uniform distribution between two objects (c). The robot responds by indicating confusion. Clarification with gesture (b) causes a probabilistic update leaving the model highly confident it has inferred the correct object (d). The robot responds by smiling and handing the user the object they referenced.

3.4.2 System Comprehension User Study Results

Our real-world experiments measured our algorithm's performance when a person referred to an object visually and with gesture. The subject stood in front of a table with four objects placed approximately one foot apart, forming four corners of a square. We instructed subjects to ask for the indicated object in the most natural way possible, using whatever combination of gesture and language they felt appropriate. We indicated the object to refer to using a laser pointer, and we periodically shifted to a different object on a predetermined schedule. They wore a microphone and we used the HTML5 Speech Recognition package in conjunction with Google Chrome to recognize speech. This package reported incremental output as recognition proceeds, and we performed a model update each time a new word was perceived. We used 13 subjects each participating in five trials for a total of 65 trials.

Results showing the percent of the time the estimated most likely object was the true object appear in Table 3.3 with 95% confidence intervals. During a typical trial, the model starts out approximately uniform or unimodal on the previous object (we did not reset the model between trials.) As the subject points and talks, the model quickly converges to the correct object. Our first set of results give a sense of how quickly the model converges.

Table 3.3: System Comprehension User Study Results

Model	Accuracy (± 95% confidence interval)	Accuracy at end of interaction
Random	25%	25%
Language only	32.4% ± 10%	46.15%
Gesture only	73.12% ± 9%	80.0%
Multimodal (Language and Gesture)	**81.99% ± 5.5%**	**90.77%**

To assess overall accuracy, we also report the system's accuracy at the end of a trial in Table 3.3. Multimodal accuracy with language and gesture is more than 90%, demonstrating that our approach is able to quickly and accurately interpret unscripted language and gesture produced by a person.

The difference in accuracy between gesture alone and the multimodal output is not as large as one might expect. This is in part caused by the small delay in speech recognition software as opposed to the instantaneous gesture input. Additionally, many subjects leaned towards ambiguous speech, such as "hand me that" while pointing, causing the speech accuracy for those trials to be 0%. There were some users, however, who relied on an equal mix of both, and showed large leaps in accuracy between arms and multimodal. The most extreme example is of a user who, over their five trials, achieved only 45.5% accuracy with gesture alone and 42.2% with speech alone, yet reached 85.7% multimodal accuracy, only 2 percentage points away from the sum of the two probabilities, showing the ease at which alternating speech and gesture can give accurate results overall. While a combination of ambiguous speech and gesture such as "that spoon" followed by a gesture would be more accurate than just a gesture, we found that most test subjects either spoke with complete ambiguity or none, using phrases either of the form "hand me that thing" or "hand me the silver spoon". Therefore we were unable to fully test this hypothesis.

3.4.3 System Interaction User Study Results

After successfully demonstrating our system in a closed environment, we ran trials involving a human user interacting with a robot. Whenever the system placed more than 70% confidence in any single object, the robot handed the person that object. We ran 40 trials, each with four objects on a table, two on each side of the robot. Users were instructed to pick an object and continue requesting it until the robot handed them the correct object.

In 80% of the trials the robot handed over the correct object on the first try. In 65% of the trials the robot handed over the desired object after a single referring expression. These trials had an average latency of 1.2 seconds between the end of the referring expression and the beginning of the robot's reaction. On average, it took 15.8 seconds from the end of the referring expression to the time the user received the object they had requested.

In the remaining 35% of the trials that the robot did not correctly infer the desired object from the first referring expression, 15% were failures where the robot simply did not respond to the first referring expression and 20% were failures where the robot handed the user an object other than the

one they requested. The former failure type can be largely attributed to rapid gestures and speech that were missed by our system. Mistranscription also played a role, but less of one. The latter failure type appears largely due to some quirks of NITE, in which the generated skeleton is actually superimposed slightly above the actual location on the body. As a result, the calculated vector came closer to the object behind the desired one, causing a failure.

3.5 Conclusion

We have demonstrated a Bayes filtering approach to interpreting object references. Our approach incorporated learned contextual dependencies, and ran in real time. This chapter demonstrates steps toward continuous language understanding and more effective human-robot interaction. However it assumes that the robot is a passive observer. In the next chapter, this Bayes filtering approach will be integrated into a decision theoretic framework that allows for robot question asking and multimodal observations.

Chapter 4

Reducing Errors in Object-Fetching Interactions through Social Feedback

This chapter presents the FEedback-To Collaborative-Handoff (FETCH) POMDP, a decision theoretic model for robot item fetching, my first example of a multimodal social feedback POMDP, as described in Section 2.4. This model was first described in Whitney et al. [34] at ICRA 2017.

4.1 Introduction

Object retrieval tasks are common in life, and are representative of tasks expected of a social robot. Humans use both speech and pointing gestures to refer to specific objects. A mechanic repairing a car, for instance, may point and ask the robot to fetch a specific tool from the shelf. There will be times, however, where the robot will fail to understand, either due to errors in interpreting the person or from a genuinely ambiguous command. It would be beneficial if the robot could communicate its lack of understanding back to the human, asking questions when necessary.

The difficulty in this task lies in the uncertainty behind the meaning of natural language and gesture. Speech-to-text software often introduces transcription errors, and human body trackers perform far worse than human level. These problems lead to ambiguities for the robot. When the robot is uncertain, we want it to ask questions, but when confident, we want it to hand the item without bothering the user. Therefore we must intelligently choose between information gathering actions and reward gathering actions. A POMDP is a natural framework for making these choices.

Existing approaches for object fetching use batch-mode language understanding to map human language commands to robot action sequences [35]. These systems do not allow for the robot to ask questions and cannot clarify ambiguity. In non-robotic domains, others have considered systems that explicitly modeled the beliefs of other agents, laying the groundwork for question asking [36].

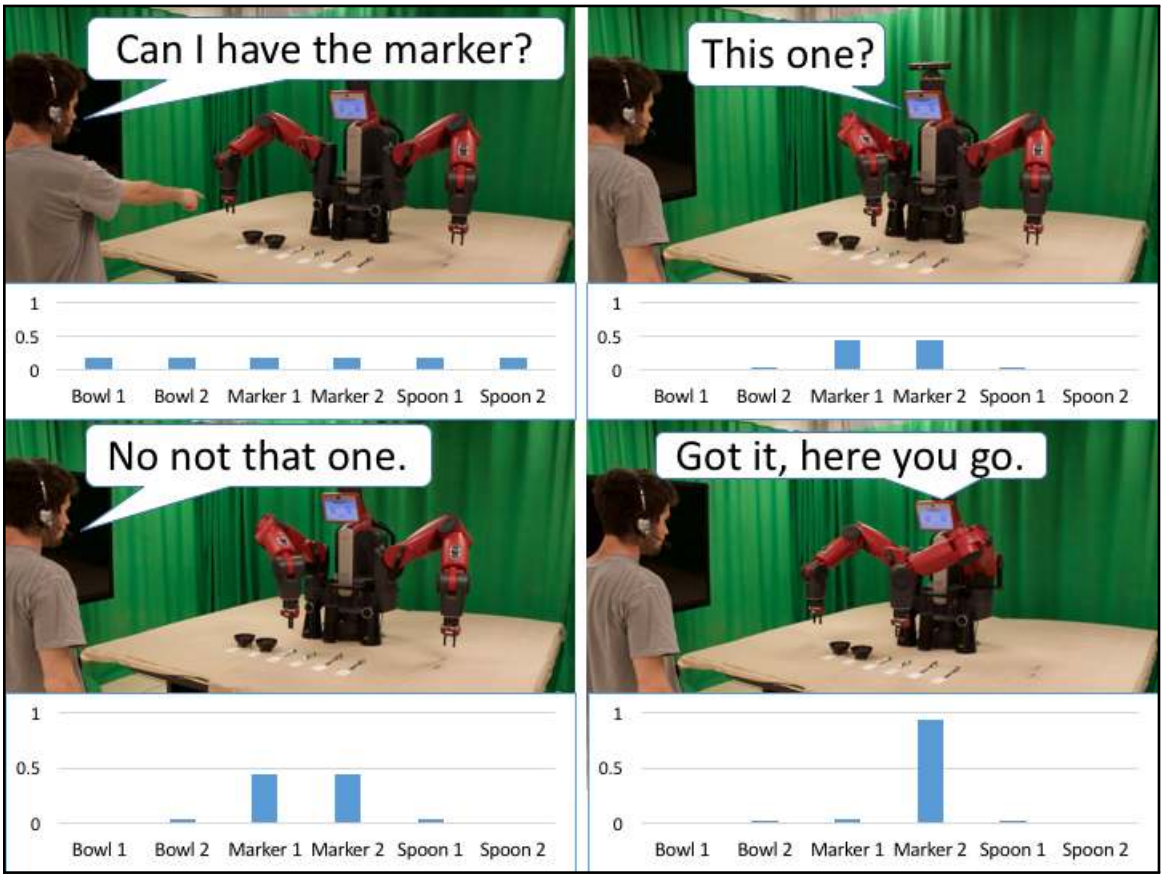


Figure 4.1: Demonstration of our FETCH-POMDP model correctly fetching item for user. Note the robot's understanding of implicit information between panels three and four. This reasoning is not hard-coded into our system, but results from the solution of our POMDP.

Williams and Young [37] created a Speech Dialog System that allows agents to model the beliefs of others in order to ask questions based on phone-based communication. Because phone lines are very noisy, that system had a fixed question asking routine it followed after choosing the question subject. In proximate human-robot collaboration, speech is less noisy.

To achieve a framework for the item-fetching domain that intelligently asks questions as well as extracts implicit information, we define the FEedback To Collaborative Handoff Partially Observable Markov Decision Process (FETCH-POMDP). Our system determines a human's desired item by interpreting natural language and pointing gestures, and can ask clarifying questions when confused.

Our model can understand implicit meaning in the humans actions, known as implicatures. Implicatures are the inferences a listener makes when assuming that the speaker is acting cooperatively. For example, in Figure 4.1, the implicature is that the speaker wanted the other marker because there is only one marker left, the speaker said they wanted a marker, and the speaker is not being deceitful about what they desire. This assumption of cooperation allows the robot to gather more information from the speakers utterance, making the interaction more efficient.

We evaluate the speed and accuracy of our FETCH-POMDP model through a real-world user study, comparing it to two state-of-the-art baselines. We had 16 users request items from our robot with either an ambiguous or unambiguous item configuration. FETCH-POMDP was the most accurate method in an ambiguous environment, and the fastest in an unambiguous environment.

4.2 Related Work

Early works in robot-question asking realized the potential of question asking to increase acccuracy but were limited to rule based approaches [38].

Common methods of natural language processing treat speech as a serialized process and infer utterances through batch-mode approaches [17, 39, 40]. These methods typically do not take into account situational context or other agents' beliefs to correct failures. Our work aims to make robotic inference of human desires an interactive process. An interactive decision process allows for certain language utterances to mean different things to the robot depending on the current situation, creating richer communication channels between the agents.

In the learning from demonstration domain, researchers such as Cakmak and Thomaz [41] have investigated what questions are useful for learning new skills. Our work is concerned with completing a known task, and focus on when to ask questions as opposed to what type of question to ask.

Vogel et al. [42] researched how implicatures allow agents to communicate more information than what is in an utterance, allowing quicker and smoother interactions. Implicatures arise in Decentralized POMDPs (Dec-POMDP) when agents model the state of other agents to maximize joint utility [42]. Due to the fact that the agent keeps a model of the desired object the human has in mind in the FETCH-POMDP, implicatures naturally arise during the interactions.

POMDPs are used in many approaches for solving decision problems where the environment is noisy and not perfectly observable. For example, Hoey et al. [43] created a decision making system from a POMDP for a robot helping dementia patients wash their hands, where the agent must infer the human's actions and psychological state through noisy hand and towel tracking. Since agents keep track of states and personal histories internally, POMDPs are a natural choice for modeling multi-agent settings [44, 45]. Gmytrasiewicz and Doshi [36] used a POMDP to handle a multi-agent setting more interactively than typical approaches. By augmenting the state space to include a limited construct of other agent's beliefs, each agent can reason over the states and actions of the other agents while solving for the optimal policy. Gmytrasiewicz and Doshi [36] prove how modeling the interaction as an Interactive-POMDP (I-POMDP) allow agents to independently compute optimal policies. However, Gmytrasiewicz and Doshi [36] state that an I-POMDP's belief-depth modeling of agents has to be limited because it is impossible to solve the infinite-recursive chain of beliefs. Our approach, in contrast, makes the simplifying assumption that only the last item referenced matters, rather than a full inference of the other agent's belief, enabling real time inference.

Williams and Young [37] modeled a spoken dialogue system as a POMDP, showing how POMDPs are a strong statistical model for determining an optimal policy between two speaking agents. Rather that requiring fixed heuristic for inferring observations such as confidence scoring, automated planning of long-run interactions, or parallel state hypotheses of the world, modeling the system as a POMDP allows a statistical approach to frame the optimal decisions. The authors limit the scope of their trials to speech-related communication tasks. Our work differs by implementing a POMDP model on a robot agent to perform the item-delivery task with a human using both speech and gesture. Furthermore, Williams and Young [37] always ask questions, regardless of context, and use the POMDP to have a policy on which questions to ask, while our work allows the robot to decide whether to ask questions at all.

Chai et al. [46] created a probabilistic model for human-robot interaction that allows a human to inform a robot of objects in the environment using natural language, and the robot to ask for clarification using both speech and gesture. The question-asking policy is fixed. Our work differs in that the FETCH-POMDP generates its own policy based on its observations.

Wu et al. [47] addressed the item-fetching domain by formalizing a POMDP that allowed a robot agent to model the user's beliefs to calculate a policy based on multiple noisy communication modalities. However, Wu et al. [47]'s state space was very large, preventing quick inference and real-time calculation of policy.

Doshi and Roy [48] implemented a POMDP model to understand natural language in order to infer noisy communication and ambiguous word choice. By modeling the dialog manager as a POMDP, Doshi and Roy [48] balances between question-asking for ambiguity clarification with action-taking to fulfill the human's request. However, their state factorization does not include a representation of the human's belief's, which prevents their model from inferring implicatures. Our work naturally infers implicit information from observations and the extra modality of pointing.

4.3 Technical Approach

Imagine a person carrying out a task, such as assembling a piece of furniture or cooking a meal. To complete the task, they need something, such as a screwdriver or a whisk. They use language and gesture to inform the robot of which item they need. The robot observes their language l and gesture g and must select the correct item i as quickly and accurately as possible.

Because of noise in speech and gesture observations, the robot will not be able to infer i from the initial speech and gesture of the human. We therefore want the robot to ask questions when, and only when, it is confused, so as to be accurate while not bothering the human unnecessarily. We must balance between information gathering actions, like asking questions, and goal inducing actions, like fetching. Therefore, we model this problem as a POMDP.

We define a novel model, the FEedback-To-Collaborative-Handoff Partially Observable Markov Decision Process (FETCH-POMDP), to solve our object fetching problem by intelligently selecting when to provide feedback based on its belief state.

4.3.1 FETCH-POMDP Definition

Solving POMDPs is very challenging; to make progress, we must define a model with specific state representations and independence assumptions that enables efficient inference. For more information on the general form of POMDPs, see Sec. 2.3

Our POMDP model for the item-delivery task is called the FEedback-To-Collobarative-Handoff POMDP, or FETCH-POMDP. The model is specified by components $\langle I, S, A, R, T, O \rangle$.

- I is a list of all items on the table, which we assume are known and fixed. Each item $i \in I$ has a known (x, y, z) location on the table, and a set of associated words $i.\texttt{vocab}$ that may be used to refer to itself.

- S: $i_d \in I$ is the human's desired item which is hidden. For convenience, we also include the last item the robot asked about (or `null` if none): $i_r \in I \cup \{\texttt{null}\}$. Note that i_r is known and hence the state (i_d, i_r) is mixed observable [49].

- A: We categorize actions as social feedback and physical actions. The physical actions consist of a wait action and a parametrized pick(i) action. The wait action merely advances the time-step by one. A pick(i) action finalizes the robot's selection of item i as the user's desired object, and the interaction terminates. The social feedback actions consist of a parametrized point(i) action. When the robot chooses to point at an item i, the robot moves its end effector in a pointing motion above item i, and asks "this one?" Because both the pick(i) and point(i) are parametrized by the items on the table, there are $2|I| + 1$ total actions available at any time.

- $R(s, a)$: We provide a large positive reward for picking the correct item, a large negative reward for picking an incorrect item, and smaller negative rewards for wait and point. The costs of the different actions were initially set to correspond to the number of seconds it would take to complete said action, and were tuned from there using both simulated trials and a small pilot study, tuned to result in the shortest interaction time and highest accuracy, regardless of social feedback paradigm.

a	s	$R(s, a)$
pick(i)	$i = i_d$	$+10$
pick(i)	$i \neq i_d$	-12.5
point(i)	$*$	-6
wait	$*$	-1

- $T(s, a, s') \equiv p(s' \mid s, a)$: Our transition function is deterministic. We assume that i_d, the desired object, remains fixed. We also assume that after the robot asks about item i, i_r changes deterministically from `null` to i.[1] Littman [51] and others [52] have shown that deterministic

[1] We could model the transition of i_r as being stochastic, to capture the possibility of the human not understanding the robot's question. This is very important in domains where the only method of communication is noisy, e.g. a phone-line [50]. In a domain like ours, where the human can both see and hear the robot with high fidelity, we were able to design our robot actions so the human understood the robot's question with near perfect accuracy.

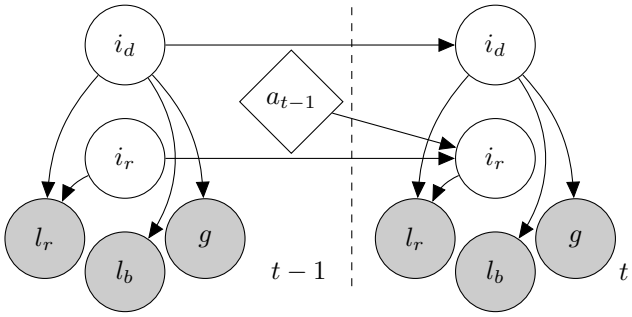


Figure 4.2: A graphical model of our FETCH-POMDP. Hidden variables are white, observed variables are gray.

POMDPs retain much of their expressive power compare to stochastic POMDPs. The focus of our model is to estimate the value of a hidden variable, not handle stochastic transitions. Therefore the complexity in our problem arises primarily from our observation function.

- $O(s, o) \equiv p\,(o \,|\, s)$: Observations consist of the human's language, l and gesture, g. To define the POMDP the robot needs a model of $p(o|s) = p(l, g|s)$. Most of the complexity of our model is captured in this observation model which is defined in the next section.

4.3.2 Observation Model

Users may produce speech and gestures, which we consider as observations in our model. Each observation $o \in \Omega$ is a tuple of language l, and gesture g.

- Language: Let l be the string of words the user has said.[2] We split l into two portions: The response utterance l_r consisting of positive/negative response words, and the base utterance l_b consisting of all other words. Either of these two strings may be empty (ϵ). To determine which words the user spoke are part of l_r, we compare each word in l to a list of positive and negative responses. The positive responses r_p are the words { 'yes', 'yeah', 'sure', 'yup' } and the negative responses r_n are { 'no', 'nope', 'other', 'not' }.

- Gesture: g is the pointing vector, measured from the user's head to the user's wrist.[3]. If no pointing is detected, g has value `null`

The entire observation calculation is given as follows:

$$p\,(o \,|\, s) = p\,(l_b, l_r, g \,|\, i_d, i_r)\,. \tag{4.1}$$

We assume the three observation components are conditionally independent given the state.

$$p\,(o \,|\, s) = p\,(l_b \,|\, i_d, i_r)\;\; p\,(l_r \,|\, i_d, i_r)\;\; p\,(g \,|\, i_d, i_r)\,. \tag{4.2}$$

[2] l is obtained by transcribing microphone input using CMU Pocketsphinx [53], a speech-to-text software.

[3] g is obtained using a Microsoft Kinect and OpenNI's skeleton tracker software [29].

As can be seen in our graphical model (Fig. 4.2), l_b and g do not depend on i_r, as the response of the human is captured in l_r. Therefore

$$p\left(o \mid s\right) = p\left(l_b \mid i_d\right)\ p\left(l_r \mid i_d, i_r\right)\ p\left(g \mid i_d\right). \tag{4.3}$$

We will now describe each portion of Equation 4.3.

Language Component

The probability of the base utterance is $p\left(l_b \mid i_d\right)$. It is calculated according to a smoothed unigram speech model. This unigram model, also called a bag-of-words model, considers the probability of each word independently. Each utterance l_b is broken down into its individual words $w \in l$:

$$p\left(l_b \mid i_d\right) = \begin{cases} p_l\ \prod_{w \in l_b} p\left(w \mid i_d\right)\ , & l_b \neq \epsilon \\ 1 - p_l\ , & l_b = \epsilon. \end{cases} \tag{4.4}$$

The probability of each word (within the product term) is:

$$p\left(w \mid i_d\right) = \frac{\mathbb{I}[w \in i_d.\texttt{vocab}] + \alpha}{|i_d.\texttt{vocab}| + \alpha\ |\texttt{words}|}\ , \tag{4.5}$$

where $\mathbb{I}[w \in i_d.\texttt{vocab}]$ is one if w appears in the vocabulary of i_d, and zero otherwise. $|i_d.\texttt{vocab}|$ is the number of words in the vocabulary of i_d. $|\texttt{words}|$ is the total size of the vocabulary. α is the smoothing parameter, which guarantees the probability of a word can never be zero. Also, p_l is the probability an utterance is made. We empirically chose $\alpha = 0.2$ and $p_l = 0.95$ based on simulation trials and the small pilot study.

Next we consider the probability of the response, $p\left(l_r \mid i_d, i_r\right)$. We make another conditional independence assumption, so that each word u in l_r is independent.

$$p\left(l_r \mid i_d, i_r\right) = \begin{cases} p_l\ \prod_{u \in l_r} p\left(u \mid i_d, i_r\right)\ , & l_r \neq \epsilon \\ 1 - p_l\ , & l_r = \epsilon. \end{cases} \tag{4.6}$$

To calculate $p(u|s)$, we must consider three possibilities for the state: $i_r = i_d$, $i_r \neq i_d$, and $i_r = \texttt{null}$. If $i_r = i_d$, then it is very likely that the user will respond with a positive utterance, and very unlikely that they will respond with a negative utterance. If $i_r \neq i_d$, then the opposite is true. If $i_r = \texttt{null}$, then no question has been asked, so both types of responses are equally likely. The mathematical representation of $p\left(u \mid s\right)$ is governed by the following conditional probability table:

Table 4.1: Conditional Probability Table for $p\left(u \mid i_d, i_r\right)$

	$u \in r_p$	$u \in r_n$
$i_r = i_d$	0.99	0.01
$i_r \neq i_d$ and $i_r \neq$ null	0.01	0.99
$i_r =$ null	0.5	0.5

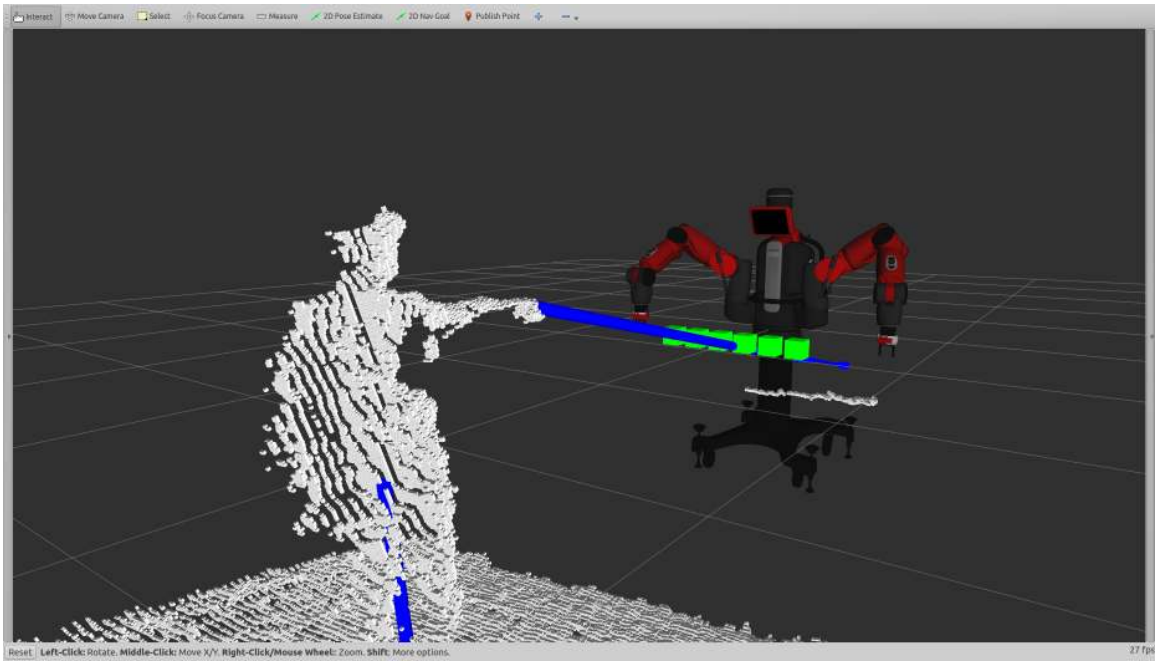


Figure 4.3: Visualization of a user pointing at item. The blue vectors represent the calculated pointing vectors from each arm. The left arm is down at the user's side, and the right arm is pointing at item four.

The 0.99 and 0.01 values correspond to our assumption that the human is cooperating with the robot and will respond truthfully to questions. The 0.5 values come from the fact that if no question has been asked, either response type is equally likely.

Gesture Component

Gesture is measured as a pointing vector starting at the head of the user and moving through the user's wrist. (see Fig. 4.3). We assume a user points directly at their desired item i_d, with a Gaussian noise term on the angle with mean zero and standard deviation σ. Let θ_{i_d} be the angle between the observed pointing vector and an ideal pointing vector directly pointing at i_d. From our pilot study, we determined $\sigma = 0.15$ radians and $p_g = 0.1$ resulted in the fastest interaction time and highest accuracy.

$$p\left(g \mid i_d\right) = \begin{cases} p_g \, \mathcal{N}\left(\theta_{i_d}\,;\, 0, \sigma^2\right) & g \neq \texttt{null} \\ 1 - p_g & g = \texttt{null}. \end{cases} \tag{4.7}$$

To determine if a gesture was made, we created a threshold for our gesture function. If $\theta_i > 0.3$ radians for all objects i on the table, then we considered the user to not currently be pointing, and $g = \texttt{null}$.

4.3.3 Solving the POMDP

Our observation space for language is countably infinite, and our observation space for gesture is continuous. This observation space makes solving the POMDP challenging; we solve it using an approximate solver, sparse sampling [54], on the resulting belief MDP for the POMDP. All POMDPs can be converted into a corresponding belief MDP, which is an MDP where every belief in the original POMDP is a state. The state space of the belief MDP is therefore continuous [5]. The solution to the belief MDP is identical to that of the original POMDP [5].

Sparse sampling finds an approximate solution to the MDP by constructing a probabilistic decision tree of finite depth d, where each node is a state-action pair, and chooses the action whose branch has the highest expected reward. To construct the tree, the algorithm samples a finite number n of observations from each node, and treats these as the total observation space of each node. This type of solver is called a receding horizon planner, because the planner can only consider states up to d actions away. Therefore the solver's accuracy increases as d and n increase. Of course, as d and n increase, runtime also increases.

From our simulations and pilot study, we found $d = 2$ and $n = 10$ lead to appropriate action choices while running quickly enough to enable real-time communication.

Sampling Language

We model the sampled l_b is a single word sampled from the observation function. We do the same for l_r. We constrained the length of the samples to one word to speed up calculations. In our simulations, we did not find this constraint affected performance.

Sampling gesture

Gesture is sampled from the observation function for gesture. The simulated human will directly point at the desired item, with an added noise term sampled from the Gaussian distribution described in 4.3.2.

4.4 Evaluation

The goal for our system is to perform robot-to-human object hand-off as quickly and accurately as possible. We define the speed of the interaction as the time the human begins the request to the time the robot decides to pick an item. We report accuracy as whether the robot decided to hand over the item the human desired.

To evaluate our system, we conducted a user study where users used language and gesture to instruct the robot to hand them a particular item. We had two physical configurations of the items, ambiguous and unambiguous. In each physical layout we tested three robot interaction paradigms. In paradigm one, the robot never gave social feedback. This is equivalent to an improved version of the model from our previous work [11]. In paradigm two, the robot always asked at least one

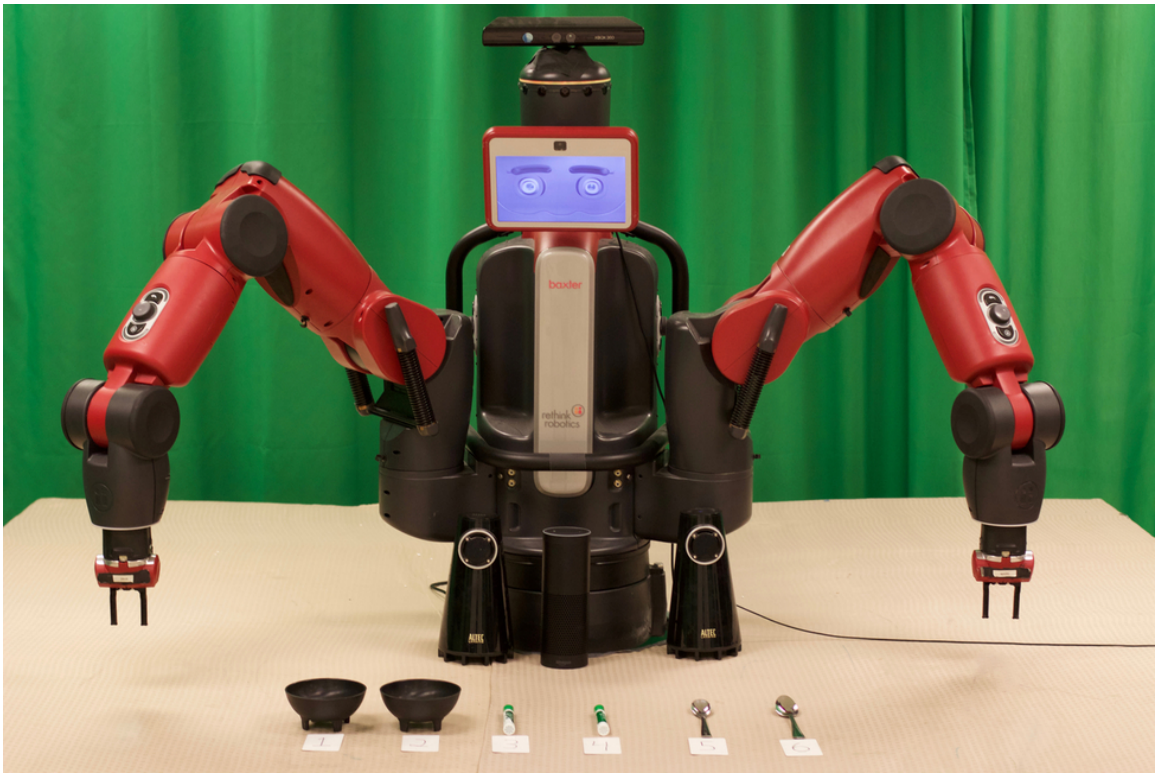


Figure 4.4: The user's view of robot, with items arranged in the ambiguous configuration.

question about the item it considered most likely until it was 95% confident of its answer. This is comparable to the PODMP model described by Young et al. [50], where the system determined what piece of information to ask about via a POMDP solution, but had a fixed question asking routine. In paradigm three, the robot intelligently asked questions according to the found solution of the FETCH-POMDP. We report the speed and accuracy at this task across all combinations of physical layouts and interaction paradigms.

Our motivation for these physical configurations is to test the two ends of the spectrum for needing social feedback. When the environment is unambiguous, the robot should be able to intelligently infer that it does not need to be asking lots of questions, but as the environment becomes ambiguous, the robot will intelligently infer the need to ask questions.

Our evaluation aimed to assess the effectiveness of our autonomous system at increasing the speed and accuracy of our human-robot interaction. Specifically we had the following two hypotheses:

- **H1:** In the unambiguous configuration, our autonomous system will be at least as accurate as the two baselines, faster than always-social feedback, and at least as fast as non-social-feedback.
- **H2:** In the ambiguous configuration, our autonomous system will be more accurate than no-social feedback, and as accurate as always-social-feedback. Our system will be faster than always-social and at least as fast as no-social-feedback.

The dependent variables are the accuracy and elapsed time measures recorded with each trial (see Sec. 4.4.2). The independent variables are what interaction paradigm was used by the robot, and the physical configuration of the items on the table. The null hypothesis for H1 is that all interaction paradigms will have have same accuracy and elapsed time in the unambiguous layout configuration. The null hypothesis for H2 is that all interaction paradigms will have have same accuracy and elapsed time in the ambiguous layout configuration.

4.4.1 Physical Configuration

Each user stood in front of a Baxter robot with six items spread across a table directly in front of the robot, as shown in Figure 4.4. The items were two black plastic bowls, two green expo markers, and two silver metal spoons. The two bowls, two markers, and two spoons are identical except for their locations. The Kinect was mounted on the robot's head.

In the unambiguous layout, the items were spread far apart from one another along a large arc in front of the robot (inter-item distance of 45 cm), and the user stood 1.22 m away from the objects, at the minimum range for the Kinect. The items were spread to cover the entire reachable span of the robot. Identical items were placed far apart from one another, so as to be easily distinguishable using pointing gestures. In the ambiguous layout, the items were in a line at the center of the table, and the user stood 3.2 m away, just inside the Kinect's maximum range of 4 m. Identical items were placed next to each other (i.e. bowls next to bowls and spoons next to spoons) with an inter-item distance of 15 cm, making pointing less effective. Any closer and the robot's pointing action would have become uninterpretable. Half of the users had the items in the ambiguous layout, and half the unambiguous layout.

4.4.2 Experimental Procedure

We want each item to be selected an equal number of times with each interaction paradigm, so we gave each user a fixed list of items to select. The ordering of the list was shuffled. The user requested the item from the robot using natural language and gesture, and was instructed to treat the robot as they would a person. The interaction began following a countdown given from the experimenter, and ended when the robot told the user which item it thought was desired. We had 16 users in total. Each user conducted 54 trials, 18 with no social feedback, 18 with intelligent social feedback, and 18 with always-social-feedback. For each of the interaction paradigms, every item was selected as the desired item 3 times. For each trial, two variables were measured; length of trial and correctness of the robot's prediction.

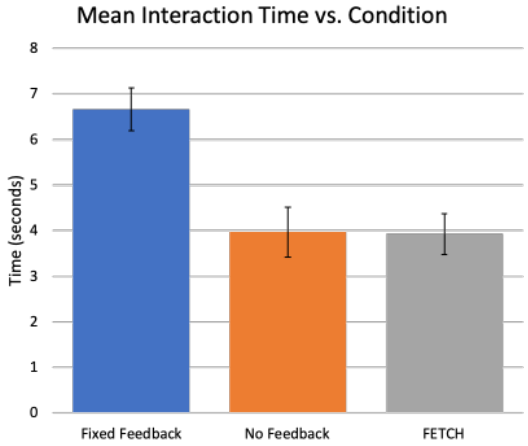
4.4.3 Statistical Analysis

Note that this study partially follows a within-subjects design. All users perform trials with all interaction paradigms, but only perform trials with one of the two item configurations. We would have preferred to conduct a full within-subjects design study, but doubling the trials for each user

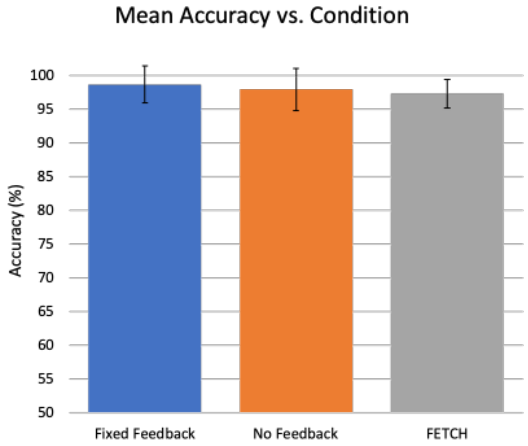
would have meant an average study time of an hour per user, which would have led to user fatigue. Interaction paradigm efficacy was more susceptible to individual differences in pilot studies, so we chose for those variables to be tested within subjects.

Because our interaction paradigms were measured within-subjects, we tested for significance with the Wilcoxon signed-rank test, a non-parametric statistical hypothesis test. It is similar to the paired Student's t-test, but does not assume that the data is normally distributed [55].

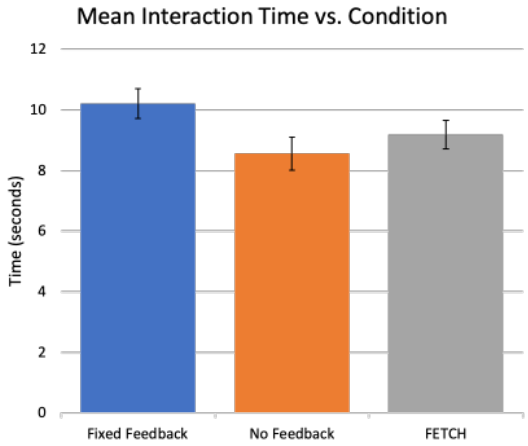
4.4.4 Results



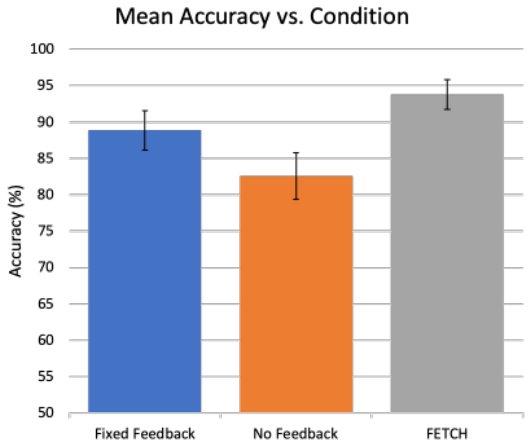
(a) Mean interaction times in the unambiguous configuration.



(b) Mean accuracies in the unambiguous configuration.



(c) Mean interaction times in the ambiguous configuration.



(d) Mean accuracies in the ambiguous configuration.

Figure 4.5: Average interaction time and accuracy for users. Error bars represent standard error of the mean.

Overall all systems were accurate, detecting the correct item with 88.4% accuracy in the ambiguous configuration and with 97.9% accuracy in the unambiguous configuration. Overall mean interaction time was 9.31 s in the ambiguous configuration and 5.86 s in the unambiguous configuration.

The results of our experiments confirmed our hypotheses. In the ambiguous configuration, our model was not significantly slower than no social feedback ($p = 0.06$), with an average difference of 0.59 s, but was significantly faster than always asking social feedback ($p = 0.03$), with an average difference of 1.05 s. There was no significant difference in accuracy between our model and the always asking policy ($p = 0.14$), but our model was significantly more accurate then not asking ($p = 0.003$), with an average improvement of 11.1%.

In the unambiguous configuration, our model was significantly faster than always asking ($p = 3.62 \times 10^{-22}$) with an average difference of 3.28 s, and not significantly faster than not asking ($p = 0.89$). All paradigms in the unambiguous configuration had average accuracies above 97%, with no significant difference between them. See Fig. 4.5 for a graph of these results.

Combining the two physical configurations together, FETCH-POMDP was significantly more accurate than never asking by 5.21% ($p = .014$), while being just as fast (0.03 s faster on average). When combining the physical configurations, FETCH-POMDP was significantly faster than the fixed asking policy by 2.17 s, or 25% faster ($p = 1.7 \times 10^{-17}$), while also being more accurate (2.1% more accurate on average). Each user completed a qualitative survey after performing all the trials. When asked about what they thought the robot understood, all users correctly inferred that the robot understood pointing and basic name descriptions of items. Interestingly, 6 users, or 38%, thought the robot could also understand prepositional phrases such as "to the left of x".

4.5 Discussion

We were surprised that FETCH-POMDP was more accurate than the fixed feedback policy in the ambiguous configuration. We had hypothesized that fixed feedback would be the most accurate, since asking more questions should remove more confusion. We found, however, that asking too many questions risked speech-to-text failures that would confuse the system. One mistake we repeatedly saw during trials, for instance, was misinterpreting the word 'yes' as the word 'hand.' The more questions the system asked, the higher the chance of a transcription error. This is why the fixed feedback policy had a lower average accuracy than FETCH-POMDP in the ambiguous configuration. In the unambiguous configuration, the pointing observations were so much stronger that the fixed feedback model rarely needed to ask more than one question, so transcription error did not noticeably affect accuracy.

Another surprising result was that FETCH-POMDP was on average faster than no feedback in the unambiguous configuration. This is because the system was usually able to infer the correct item from its initial observations, but occasionally would be unsure. With FETCH-POMDP, the robot was able to ask a question, resolve the ambiguity, and pick the desired item. Without social

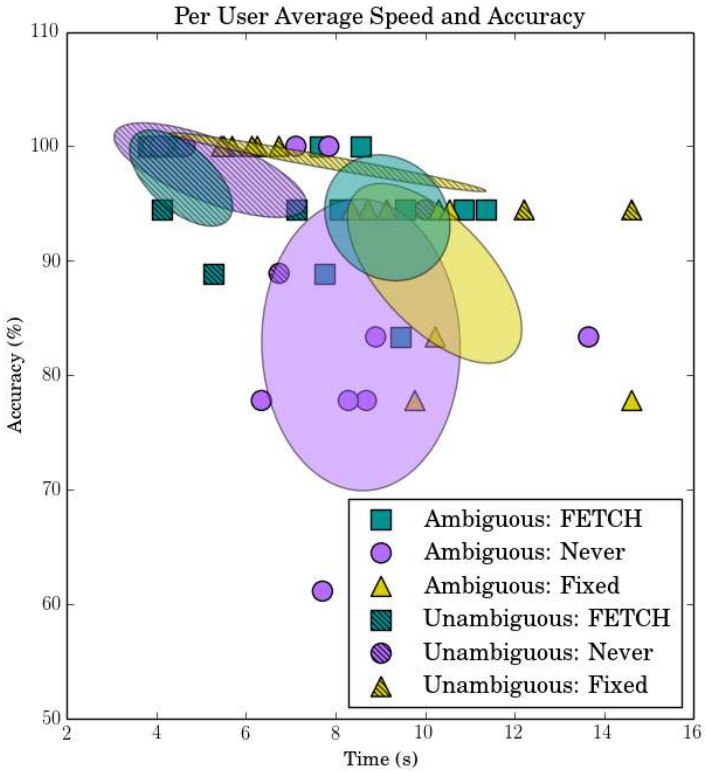


Figure 4.6: Average accuracy and time for each user across each interaction paradigm. Each point represents the average accuracy and trial time for an interaction type for a single user. Ellipses represent Gaussian distribution fitted to points to one standard deviation. Note how the FETCH-POMDP ellipses (shown in green), are farthest to the top and left, with the smallest standard deviations.

feedback, the robot could only wait. The human wouldn't immediately realize the robot needed more observations, so the interaction would come to a standstill. These outlier interactions can be seen in Figure 4.6.

During trials, many users used prepositional phrases in order to describe items, such as "Hand me the spoon to the left of the bowl." Although the language model in this work did not account for referential phrases, the agent was able to use intelligent social feedback to figure out what the human desired. This may explain why many users reported that they thought the robot did understand prepositional phrases, and this result suggests question asking improves the perceived competence of the robot.

4.6 Conclusion

This chapter shows how social feedback improves human robot communication, and how POMDPs are effective methods of generating this feedback. Using multimodal observations to model the hidden state of the human from noisy signals allows for richer state extraction than either modality alone. The FETCH-POMDP's framework allows extensions to make a more sophisticated model of the agent's hidden states. This lends itself to a more general framework that can model agent's mental states in more generalized interactions.

In the next chapters, we will temporarily leave the world of decision theory, and enter into the world of virtual and mixed reality, and see to how to integrate each into a robotic system to achieve efficient and safe human-robot interaction.

Chapter 5

Human Robot Interaction via Virtual Reality

This chapter describes ROS Reality, a set of software packages designed to connect AR and VR devices to a ROS-enabled robot, as well as an over-the-internet virtual reality teleoperation system, as described in Section 2.7. I, along with Eric Rosen, envisioned and constructed the system, which has served as the basis of much of my subsequent work, as well as works from other researchers in my lab [56, 57]. ROS Reality was initially described by Whitney et al. [58] from a user interface standpoint, and was subsequently described by Whitney et al. [59] from a system design standpoint.

5.1 Introduction

Virtual reality (VR) is a compelling interface for robots because it enables fluid interactions in the real physical world, and allows users to specify points and transforms in an intuitive way. VR interfaces can be used for teleoperation, robot teaching, and learning from demonstration, as well as debugging on the robot remotely.

A major benefit of VR systems is that they allow non-expert users to control robots. The mapping of robot manipulators to VR controllers creates an interface in which manipulators act as extensions of the users' hands. This human-robot interface can allow novice users to intuitively perform a variety of dexterous robot manipulation tasks without extensive training. Thus, VR interfaces may also be a means to leverage the proficiency of human users to facilitate robots learning complex, fine-grained manipulation tasks. VR therefore permits non-experts to control robots, and leverage their experience in challenging domains.

However, integrating robots with a VR system is challenging. There is no standard interface to connect ROS [60] to standard virtual reality paradigms, such as Unity, so that it can be used with consumer-grade hardware, such as the HTC Vive. Additionally there is a lack of standardization in terms of tasks and use cases for these systems.

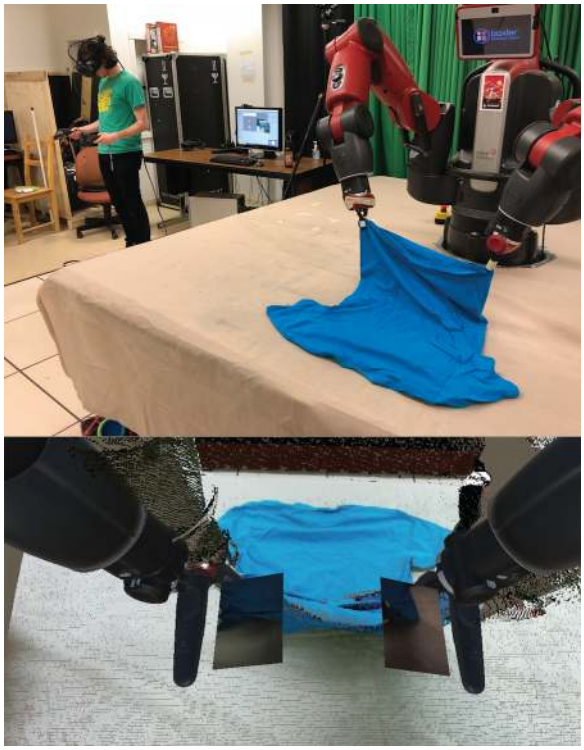


Figure 5.1: Top image: An operator using ROS Reality VR to teleoperate a Baxter to fold a shirt. Bottom image: View of scene from VR headset. Note a point cloud, mesh model of robot, VR controllers, and wrist camera feeds from robot are all visible to the user.

With this in mind, we present ROS Reality. ROS Reality is a VR and Augmented Reality (AR) teleoperation interface using consumer-grade VR and AR hardware with ROS-enabled robots. It allows users to view and control robots over-the-Internet using consumer-grade VR and AR hardware. ROS Reality has served as the technical basis for the VR research in Whitney et al. [58], and for the AR research in Rosen et al. [61]. A VR teleoperation demonstration using ROS Reality is shown in Fig. 5.1. In this chapter, we focus on the VR system architecture and application of ROS Reality.

We detail our consumer-grade VR and AR teleoperation interface, ROS Reality. We discuss how the package allows for a ROS-networked robot, like Baxter from Rethink Robotics, to bilaterally communicate over the Internet with an HTC Vive through the Unity game engine. We also present the results of a pilot study conducted to test the efficacy of using ROS Reality to teleoperate a robot to perform 24 dexterous manipulation tasks. Portions of this work previously appeared in an extended abstract by Rosen et al. [62].

5.2 Related Work

Teleoperation enables robots to complete tasks that would otherwise be too difficult to complete autonomously, such as in the DARPA Robotics Challenge [63], and also allows humans to operate by proxy in environments that would normally place them in harm's way [64]. 2D interfaces for robot teleoperation, especially over the Internet, have become popular in recent years [65]. Monitor and keyboard control schemes have been used to control robots for a variety of classical tasks like motion planning and grasping [66]. Web browsers have proven especially useful in allowing anyone around the world with a computer to teleoperate a robot, broadening the user base [67]. However, 2D monitor interfaces do not reflect the way that humans observe and interact with the 3D world. Our research has shown that a VR interface can address this problem as non-expert users were faster, more efficient, and preferred using a VR interface over a 2D monitor interface for teleoperation [58].

Virtual reality interfaces and gantry systems are intuitive ways to directly map a user's actions to those of the robot they are controlling [58]. For example, the da Vinci Robot System is an immersive haptic telesurgery system that has improved surgical performance for both novice and experienced users [68]. Although powerful, the interface is specific to the surgical domain. Exoskeleton systems are intuitive to control, but are limited to specific robots and is extremely expensive, limiting the potential operator-base compared to web-based interfaces [69].

Recent advancements in graphics have made commercially available VR systems accessible to the gaming community. Systems like the HTC Vive, Oculus Rift, and Google Cardboard offer cheap and portable VR hardware. As a result, researchers have recently begun exploring these VR systems for robot teleoperation. Zhang et al. [70] used an HTC Vive to teleoperate a PR2 and perform imitation learning. Lipton et al. [71] also used a commercially available VR system for performing teleoperation on a Baxter. Our previous work [58] on comparing VR to 2D teleoperation systems also used an HTC Vive for the VR interface, enabled through ROS Reality. By having labs use the same VR systems, results and interfaces are easier to duplicate.

The proliferation of consumer-grade VR systems is very recent, so there has been little research on the efficacy of teleoperation interfaces that use this technology (e.g., [70]). Although task completion depends heavily on the interface type and particular robot, we were interested in exploring what complex tasks could be completed on our open-source software using a common research robot.

Our choice of objects and manipulation tasks to evaluate on was inspired by previous work on robot task benchmarks. Kasper et al. [72] created a program to generate an open-database of over 100 object models for evaluating recognition, localization, and manipulation in service robots. Goldfeder et al. [73] released a collected dataset of items and stable grasps as a means for conducting machine learning and benchmarking grasp planning algorithms. One notable benchmark is the YCB object and model set [74], which is a set of accessible items chosen to include a wide range of common object sizes, shapes, and colors to test a variety of robot manipulation skills using accepted protocols. The YCB dataset has made it easy and cheap for any research lab to evaluate a robot manipulator on general tasks over a large object dataset. Several of the tasks performed in this evaluation come from this dataset.

5.3 ROS Reality

This section first provides a brief synopsis of interacting with a robot in virtual reality, and then a technical description of ROS Reality.[1]

5.3.1 VR as a Teleoperation Interface

There are multiple ways of displaying the robot's state to the user, and mapping the user's input to the robot. We bin these different methods into two main categories: egocentric or robocentric.

In egocentric models, the human is the center of the virtual world, and virtually inhabits the same space as the robot. Lipton et al. [71]'s homunculus work and Zhang et al. [70] are examples of this egocentric mapping. Under these conditions, human users have reported feeling like they 'become the robot' or 'see out of the robot's eyes'.

In a robocentric model, the human and robot share a virtual space, but are not necessarily superimposed on one another. The model we used for evaluating ROS Reality [58] falls into this category. Under this model, the human walks around a virtual model of the robot, and controls its arms by virtually grabbing and dragging them. We call this model a virtual gantry system.

5.3.2 System Overview

An HTC Vive is connected to a computer running the Unity game engine. Unity builds a local copy of our robot based on its URDF with a custom-made URDF parser. Unity connects to a ROS network over the Internet via a Rosbridge WebSocket connection [75]. The pose and wrist cameras of the robot are sent via this WebSocket connection, as well as the color and depth image of a Kinect 2 mounted to the robot's head. The color and depth image are built into a point cloud in Unity via a custom shader. When the user holds down a deadman's switch, the pose of the user's controllers are sent back to the robot, which uses an inverse kinematics solver to move the robot's end effectors to the specified poses. Refer to Fig. 5.2 for a visual overview of the ROS Reality system.

5.3.3 ROS

ROS (Robot Operating System) [60] is a set of tools and libraries to help program robot applications. ROS connects processes of programs, known as nodes, that perform different functions. Nodes communicate by streaming data over channels, or topics, on a local TCP network, known as a ROS network. Nodes create publisher objects to publish data over the network on a topic, or subscriber objects to subscribe to a topic.

ROS Reality launches a Kinect2 ROS node [76], two RGB camera feeds (one for each wrist camera of the robot), a Rosbridge WebSocket server [75], a custom ROS node that converts the full transform (TF) of the robot to a compact string, and another ROS node that listens for target poses from the VR systems, queries the robot's Inverse Kinematic (IK) solver, and moves the robot to the IK solution if found, or reports an IK failure if one is not found.

[1] Full source code is available at `https://github.com/h2r/ros_reality`

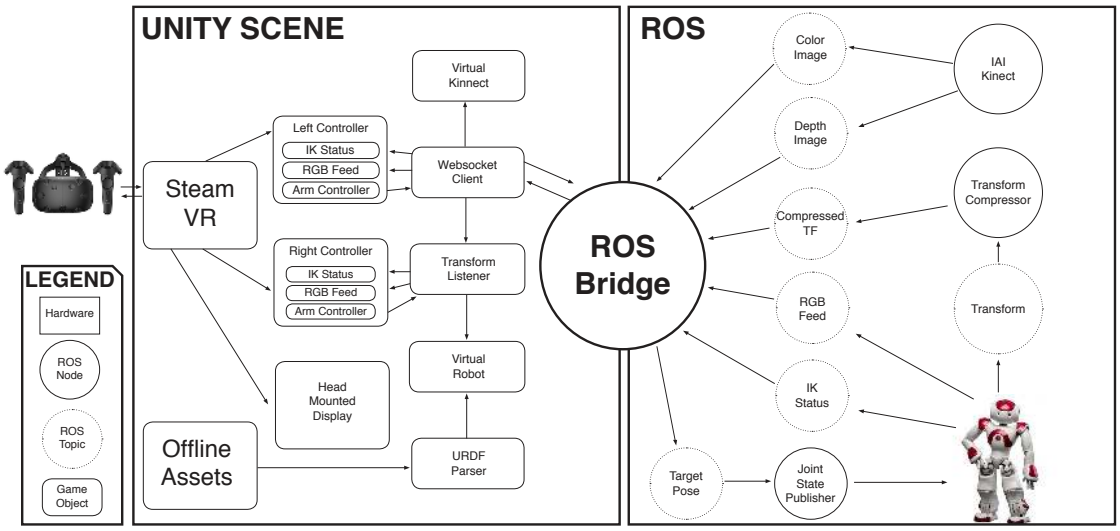


Figure 5.2: The architecture of the ROS Reality system.

5.3.4 HTC Vive

The HTC Vive is a consumer-grade virtual reality system. It has three tracked objects: one head-mounted display (HMD) and two wand controllers. Each device is tracked via two infrared pulse laser emitters, known as lighthouses, allowing for tracking via time-of-flight calculations. Each tracked object is positionally and rotationally tracked with roughly 1-2mm of error. The wand controllers are wireless and the HMD connects to a computer via a USB and HDMI cable. Each controller has a touch-pad, trigger, and two buttons for user input.

The HTC Vive supports several game and physics engines, but the initial (and in our opinion best supported) development platform is Unity.[2] The Vive connects to Unity through a software package called SteamVR.

5.3.5 Unity

Unity is a game engine used for many popular 2D, 3D, and Virtual/Augmented/Mixed Reality applications. It has a built-in physics engine that can handle contact dynamics and material simulation (such as water, sand, or cloth). It supports integration with most common VR (and AR) hardware, and provides a shader language for writing custom GPU shaders.

An open Unity environment is called a scene. In this scene are a collection of the atomic units of Unity, the GameObject. Attached to each GameObject are a set of Components. There are dozens of types of components, but the most important for our purposes is the script. A script is a small C# program that is executed at every rendering frame. The functionality of ROS Reality is implemented via a set of these Unity scripts.

[2]Formally known as Unity3D.

(a) An image of the PR2 robot visualized in Unity from the URDF Parser of ROS Reality.

(b) An image of the Baxter robot visualized in Unity from the URDF Parser of ROS Reality.

Figure 5.3: Robots parsed with the URDF Parser of ROS Reality.

5.3.6 ROS Reality

ROS Reality is a set of programs that allows a user to view and control a ROS-enabled robot over-the-Internet in VR. ROS Reality is composed of a set of C# scripts, described below.

WebSocket Client

This script is a C# implementation of the default Rosbridge client, roslibjs [77]. It supports advertising, subscribing, and publishing to ROS topics. All messages are sent and received in a JSON format, and data is encoded in base64 as per the Rosbridge specification.

URDF Parser

This script parses a Unified Robot Description Format (URDF) file and builds a hierarchy of GameObjects that comprise the robot. URDF is an XML-based specification for representing robot models common to all ROS-enabled robots. URDFs include information about each part of the robot, known as links, and how the links of a robot are connected, known as joints. The URDF Parser creates a GameObject for each link, and connects them according to the joints. Currently, we have successfully tested our URDF parser with a PR2, and Baxter, as seen in Figs. 5.3a and 5.3b.

The virtual robot has physical properties that can be simulated via Unity's physics engine. This allows the robot to interact with other GameObjects, useful for practicing teleoperation interactions in simulated scenarios.

Transform Listener

The Transform Listener subscribes to (a compact representation of) the robot's transform (TF) and moves the virtual robot to the same pose as the real robot. The ROS TF topic has the position and rotation (represented as a quaternion) of each link, which this script reads and applies to each link of the simulated robot.

One difficulty in doing this is that ROS and Unity use different coordinate frames. The Transform Listener therefore first converts the ROS positions and rotations via the following equations.

Positions:

$$x_{unity} = -x_{ros} \tag{5.1}$$

$$y_{unity} = z_{ros} \tag{5.2}$$

$$z_{unity} = -y_{ros}, \tag{5.3}$$

and rotations:

$$qx_{unity} = qx_{ros} \tag{5.4}$$

$$qy_{unity} = -qz_{ros} \tag{5.5}$$

$$qz_{unity} = qy_{ros} \tag{5.6}$$

$$qw_{unity} = qw_{ros}. \tag{5.7}$$

RGB Camera Visualizer

To visualize camera feeds from the robot, this script subscribes to a specified camera topic. When it receives the camera image it converts it from base64 and textures a plane GameObject with the camera feed. The plane GameObject is attached to the user's wand controller, so the user can always see it during manipulation. This script supports images in JPG or PNG formats, but ROS Reality always uses JPG for bandwidth reasons.

Kinect PointCloud Visualizer

The Kinect PointCloud Visualizer script uses a GPU shader to construct a point cloud out of the RGB camera image and raw depth map from a Kinect v2. The script subscribes to the RGB and depth topics of the Kinect and passes them as textures to a custom geometry shader. This shader creates a colored quad for each pair of pixels in the RGB and depth images. The color of the quad is simply the color of the associated RBG pixel. The position must be calculated. Each pixel in the depth image is the distance in millimeters of that pixel from the camera plane, so first we convert from millimeters to meters, and calculate the position of the quad relative to the camera. We then multiply that position by the transformation matrix of the Kinect in the Unity scene to get the world space position of the quad. The world space position is finally multiplied by the view and projection matrices, and passed to the vertex shader.

Arm Controller

This script allows the user to send target end effector coordinates to the robot. When a deadman's switch is held down (the side grip buttons on an HTC Vive) the current position and orientation of the controller are converted from the Unity coordinate frame to the ROS coordinate frame and published over a topic to a node in the ROS network that queries the robot's built in IK solver and moves the robot if a solution is found. Additionally, this script lets the user open and close the gripper with the trigger of the wand controller. This is also accomplished by sending a message over a topic to the robot.

The conversion from Unity to ROS for positions and rotations can be inferred from equations 5.1, 5.2, 5.3, 5.4, 5.5, 5.6, and 5.7:

IK Status Visualizer

This script subscribes to the current status of the robot's IK solver and turns the users wand controller red if the IK solver failed. This lets the user know if the target position they sent to the robot cannot be reached.

5.3.7 Robot

We use a Baxter from Rethink Robotics. Baxter is a robot designed for industrial automation applications. It has a fixed base and display screen head, with two 7 DoF arms and grippers with force sensing that enable it to dexterously manipulate a variety of objects. We attached rubber grips that come in the Baxter toolkit in order to maximize the friction at the end effector.

We have also connected ROS Reality to a simulated PR2 in Gazebo, and have been able to watch the robot move in real time, but have not yet set up the infrastructure to control that robot.

5.4 Long-Distance Teleoperation Trial

In order to test the efficacy of ROS Reality for long-distance teleoperation, we had a human operator control a robot 41 miles away, at a separate university. In this trial, we were able to successfully stack 10 cups back to back on the first attempt, as well as play a short game of chess by picking and placing pieces. The user reported no lag or bandwidth issues. For an image of this trial see Figure 5.4. A video of this demonstration can also be found online.[3]

5.5 Novice User Teleoperation Experiment

Our evaluation assesses the effectiveness of VR camera control and positional hand tracking as teleoperation interfaces. To do so, we asked novice users to teleoperate a Baxter robot to perform a

[3] `https://youtu.be/e3jUbQKciC4`

Figure 5.4: Long distance teleoperation trial. The robot is in Cambridge, MA while the teleoperator is in Providence, RI. The teleoperator was able to stack 10 cups in a row on their first attempt.

cup-stacking task in four ways: directly manipulating the arm, and using three different teleoperation interfaces: keyboard and monitor, positional hand tracking and monitor, and positional hand tracking with VR camera control. We report task completion time as an objective metric, as well as subjective assessments of system usability, likability, and workload.

5.5.1 Task

Each user was given the task of assembling three cups—all located on a table in front of the robot—into a single stack, by controlling a Baxter robot's right arm to first place the blue cup into the green cup, and then the blue-green stack into the yellow cup. The blue and green cups were placed flat on the table, while the yellow cup was propped up at a 45-degree angle. The cups were taken from the group of stacking cups in the YCB Object set [74]. The task is shown in Figure 5.5. During teleoperation, the participants controlled the robot from a computer across the room, and a divider blocked their line of sight.

This task was designed to be difficult. The cups fit snugly into each other, with a clearance of under two millimeters. The blue and green cups were not secured to the table, and were liable to be knocked over if bumped. The angle of the yellow cup required the operator to rotate the robot arm about two of its axes, a dexterous task that forced the operator to consider the arm's orientation and position simultaneously.

Figure 5.5: Pictures of the cup-stacking task: (left) the initial configuration, (middle) the blue cup stacked in the green, and (right) the blue-green stack into the yellow cup.

5.5.2 Interfaces

Our experiment compared four interfaces:

Direct Manipulation (Direct)

Users physically grabbed the arm by the wrist and moved it in order to complete the task. We chose this interface as the lower bound, best-case baseline for the task. The users were able to directly view the cups and move the arm. An ideal teleoperation system would be as fast and accurate as direct manipulation.

Keyboard and Monitor (KM)

Users viewed the scene using a 1080p 23" computer monitor. The users could move the camera through the scene using a mouse, and control the robot's end effector using a keyboard interface,[4] in a manner typical of software interfaces such as RViz [60] and Gazebo [78].

Positional Hand Tracking with Monitor (PM)

Users view the scene and control the camera as in the keyboard and monitor interface, but control the arm with the positional tracking interface. This interface allows us to study the effect of positional tracking—a relatively new aspect of VR headsets—without virtual reality camera control.

Positional Hand Tracking with Virtual Reality Camera Control (VR)

Users viewed the scene using an HTC Vive virtual reality headset, and controlled the arm using an HTC Vive hand controller. The VR headset allowed the user to move about the scene at will, and the hand controller controlled the gripper using the positional hand tracking technique described in section 5.3.6. This is the complete version of our system.

[4] The `WASD` keys governed horizontal movement, Q and E moved the arm down and up, and R and F opened and closed the grippers. The shift key switched translational movement to rotational.

5.5.3 Experimental Procedure

Users teleoperated the robot to perform the cup-stacking task with each interface. Direct manipulation was always done first to gain familiarity with the robot. Next, they performed the three teleoperation schemes in random order. There are six possible orderings of the teleoperation schemes, and we ensured each was done an equal number of times. We had 18 participants, so each of the six possible orderings were performed by three different users.

After using each interface, participants filled out subjective evaluations for that interface. After using all interfaces, participants filled out a form asking for further subjective measures, such choosing their favorite interface, and basic demographic information.

For each interface, we instructed the user how to move the robot and view the scene. We asked participants to complete the task as quickly as possible. They were then given as many attempts as they liked to complete the task. For each task attempt, the experimenter gave a countdown and then started a stopwatch. The experimenter stopped the stopwatch once all three cups were completely stacked. If the user knocked over a cup or otherwise made the task impossible, an experimenter recorded the time, reset the objects, and restarted the attempt.

5.5.4 Participants

Our evaluation used 18 participants (11 male, 7 female) with ages ranging from 18 to 22 ($M = 19.78$, $SD = 1.17$). Video game usage at peak varied between users from 0 to 30 hours per week ($M = 8.36$, $SD = 8.76$).

5.5.5 Measurements

The independent variable in our experiment was the choice of interface. Our objective dependent variable was the time to completion of the task. For this measure, we took each participant's fastest time for each interface. Five of the eighteen participants were unable to complete the task with the keyboard and monitor interface and two users were unable to complete the task with the positional tracking and monitor interface. For those users, we chose the attempt in which the user was closest to stacking all three cups.

Our subjective dependent variables were user workload as measured by the NASA Task Load Index (NASA-TLX) [79], system usability as measured by the System Usability Scale (SUS) [80, 81], and system likability as measured by several Likert scale questions. Each measure was collected via questionnaires at various points throughout the experiment.

The NASA-TLX is a widely used assessment tool that measures perceived workload of a particular task [79]. It measures global workload across six sub-scales: mental demand, physical demand, temporal demand, effort, frustration, and performance. Participants were asked to provide a rating of their perceived mental workload along each of the six dimensions via a scale ranging from 0 (Low) to 100 (High) for the first five dimensions and 0 (Perfect) to 100 (Failure) for the performance dimension. The weighted measure of paired comparisons among the sub-scales was not included. Research has

suggested that workload scores derived using the weighted sub-scales are nearly identical to those derived using the unweighted sub-scales, and adding the paired comparison ratings is time-consuming and could hinder participant recall of experienced workload [82].

Participants assessed each interface on overall usability by filling out a System Usability Scale (SUS) questionnaire [80, 81]. The SUS questionnaire asks users to rate ten sentences on a 7-point Likert scale ranging from "strongly disagree" to "strongly agree." The sentences cover different aspects of the system, such as complexity, consistency, and cumbersomeness. Like the NASA-TLX, the SUS is measured on a scale from 0 to 100. For the SUS, however, 0 is the worst score, and 100 is the best.

For our final subjective measure, we asked each participant to rate the various interfaces in terms of likability on a Likert scale from 1 to 7 and as a covariate measure, we asked participants how many hours of video-games they played per week at their peak.

5.5.6 Hypotheses

We expected that users would show the best performance (i.e., the fastest completion times, lowest levels of mental workload, highest usability and likability scores) in the Direct Manipulation Interface condition, followed by the Positional Hand Tracking with VR condition, and then the Positional Hand Tracking with Monitor condition. Finally, we posited that the Keyboard and Monitor condition would be associated with the lowest levels of performance.

Specifically, we had 3 hypotheses:

- **H1:** The Direct Manipulation Interface condition will be associated with the best performance of the four conditions, as demonstrated by (a) the fastest task completion times, (b) the lowest levels of mental workload, (c) the highest usability scores, and (d) the highest likability ratings.
- **H2:** The Positional Tracking with Virtual Reality Interface condition will be associated with the best performance out of the teleoperated conditions.
- **H3:** Of the remaining teleoperated conditions, the Positional Hand Tracking with Monitor condition will be associated with better performance than the Keyboard and Monitor condition.

The first hypothesis reflects our idea that Direct Manipulation is the easiest interface for completing the cup stacking task. The remaining hypotheses reflect our thought that using the Vive headset offers superior perception that leads to quicker task completion than looking at a monitor, and that having position/pose-tracking hand controllers will make it faster and more intuitive to control the robot than a keyboard.

5.5.7 Results

To analyze the three hypotheses, four Analyses of Covariance (ANCOVAs) were used to look for significant differences between the conditions on the four dependent measures (i.e., task completion times, NASA-TLX, SUS, and Likability measure). Planned contrasts were conducted to test for

Table 5.1: Results of Novice User Study

(a) Table of means, standard deviations, and significant contrasts between experimental conditions on the time to completion dependent measure.
ANCOVA $F(3, 14) = 37.840$, $p < .001$, partial $\eta^2 = .890$, $N = 18$, LSD Significant between two conditions at $p < .05$

	Time to complete task		
Condition	Mean	SD	Significant Contrast
Direct	8.15	2.68	KM,PM,VR
KM	153.43	44.37	Direct,PM,VR
PM	79.81	39.09	Direct, KM
VR	52.56	37.16	Direct, KM

(b) Table of means, standard deviations, and significant contrasts between experimental conditions on the NASA-TLX dependent measure.
ANCOVA $F(3, 13) = 12.289$, $p < .001$, partial $\eta^2 = .739$ $N = 17$, LSD Significant between two conditions at $p < .05$
*Contrast marginally significant at $p = .058$

	NASA-TLX Measure		
Condition	Mean	SD	Significant Contrast
Direct	29.31	12.54	KM,PM,VR
KM	56.37	13.71	Direct,PM*,VR
PM	51.08	15.90	Direct, KM*, VR
VR	44.95	20.53	Direct, KM

(c) Table of means, standard deviations, and significant contrasts between experimental conditions on the SUS dependent measure.
ANCOVA $F(3, 12) = 6.847$, $p = .006$, partial $\eta^2 = .631$, $N = 16$, LSD Significant between two conditions at $p < .05$
*Contrast marginally significant at $p = .056$

	System Usability Scale		
Condition	Mean	SD	Significant Contrast
Direct	71.25	9.97	KM,PM
KM	37.29	19.13	Direct,PM,VR
PM	55.94	21.01	Direct, KM, VR*
VR	71.46	19.61	KM, PM*

(d) Table of means, standard deviations, and significant contrasts between experimental conditions on the Likability dependent measure.
ANCOVA $F(3, 14) = 24.679$, $p < .001$, partial $\eta^2 = .894$, $N = 18$, LSD Significant between two conditions at $p < .05$

	Likability Measure		
Condition	Mean	SD	Significant Contrast
Direct	5.61	1.61	KM,PM
KM	2.06	1.35	Direct,PM,VR
PM	4.28	1.71	Direct, KM, VR
VR	6.11	1.41	KM, PM

significant differences between individual conditions. Specifically, planned contrasts were conducted to look for significant differences on the dependent measures between the Direct Manipulation condition and each of the teleoperation conditions (i.e., Condition 1 vs. 2, 3, and 4, independently). Planned contrasts were also conducted to look for differences on the dependent measures between the VR condition and each of the other teleoperated conditions (i.e., Condition 4 vs. 2 and 3), and to look for significant differences on the dependent measures between the Positional Hand Tracking with Monitor condition and the Keyboard and Monitor condition (i.e., Condition 3 vs 2).

A one-way repeated measures ANCOVA with task completion times for each experimental condition as the within-subjects variable, and scores on the measure of peak video game hours as the covariate, was used to test for significant differences in mean teleoperation task completion times across the interface conditions. The test revealed that there was a significant difference in mean task completion times across the four interface conditions, Wilks $\Lambda = 0.110$ $F(3, 14) = 37.840$, $p < 0.001$, $\eta^2 = 0.890$. Planned contrasts using the LSD method were conducted to test for significant differences in task completion times between conditions. The means, standard deviations, and statistically significant contrasts between conditions are presented in Table 5.1a and Figure 5.6.

The Direct Manipulation condition resulted in statistically significantly faster task completion times than any of the other conditions. This result supports Hypothesis H1, which stated that the

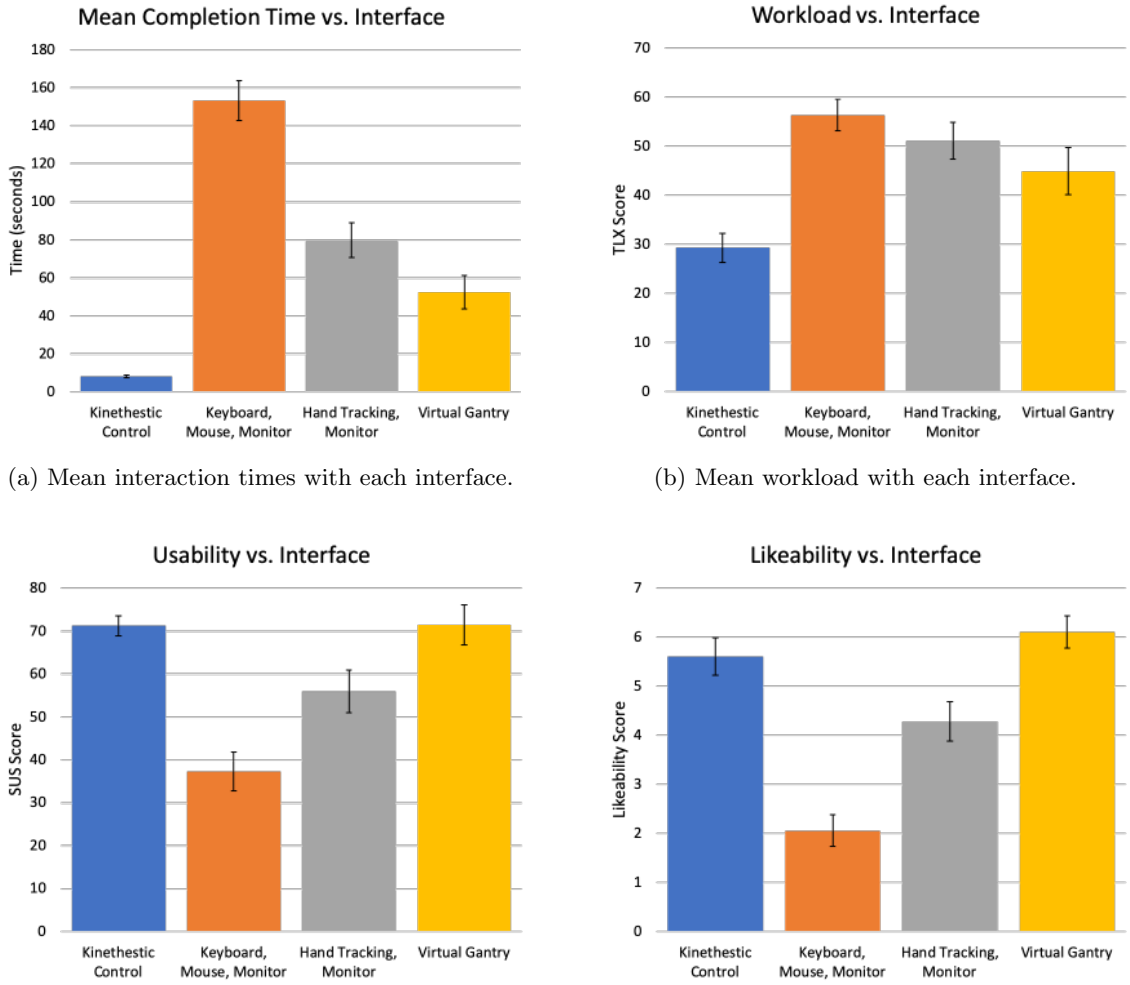
Direct Manipulation condition would be associated with the best performance on the task completion time measure. Further, of the teleoperated conditions, the VR condition was associated with the fastest task completion times. However, the VR condition was only statistically significantly faster than the KM condition, but not the PM condition. These findings only lend partial support for Hypothesis H2, which stated that the VR condition would be associated with significantly faster task completion times than both the PM and KM conditions. Finally, the PM condition was statistically significantly faster than the KM condition, which supports Hypothesis H3.

For the NASA-TLX measure, the Direct Manipulation condition resulted in statistically significantly lower subjective workload scores than any of the other conditions. This result supports Hypothesis H1, which stated that the Direct Manipulation condition would be associated with the lowest levels of workload among the four conditions. Further, of the teleoperated conditions, the VR condition was associated with the lowest levels of subjective workload. However, workload scores in the VR condition were only statistically significantly lower than the KM condition, but not the PM condition. These findings lend only partial support for Hypothesis H2, which stated that the VR condition would be associated with significantly lower workload scores than both the PM and KM conditions. Finally, the difference in workload scores between the PM condition and the KM condition was not statistically significant at the $p = 0.05$ level, instead the difference between the two conditions approached significance at $p = 0.058$. This finding only lends partial support for Hypothesis H3.

A one-way repeated measures ANCOVA with scores on the SUS for each experimental condition as the within-subjects variable, and scores on the measure of peak video game hours as the covariate, was used to test for significant differences in subjective assessments of the usability of each interface across the conditions. The test revealed that there was a significant difference in mean SUS scores across the four interface conditions, Wilks $\Lambda = 0.369$ $F(3, 12) = 6.847$, $p = 0.006$, $\eta^2 = 0.631$. Planned contrasts using the LSD method were conducted to test for significant differences in SUS scores between conditions. The means, standard deviations, and statistically significant contrasts between conditions are presented in Table 5.1c.

The Direct Manipulation condition was associated with higher SUS scores than all of the other conditions except the VR condition. Thus, Hypothesis H1 which stated that the DM condition would be associated with the highest SUS scores of all of the conditions was not supported. Of the teleoperated conditions, however, the VR condition was associated with the highest SUS scores out of any of the conditions, strongly supporting H2. Finally, the difference in SUS scores between the PM condition and the KM condition only approached significance with $p = 0.056$. This finding only lends partial support for Hypothesis H3.

A one-way repeated measures ANCOVA with scores on the Likability measure for each experimental condition as the within-subjects variable, and scores on the measure of peak video game hours as the covariate, was used to test for significant differences in assessments of how much users liked interacting with each interface. The test revealed that there was a significant difference in mean Likability scores across the four interface conditions, Wilks $\Lambda = 0.159$ $F(3, 14) = 24.679$, $p < 0.001$,



(a) Mean interaction times with each interface.

(b) Mean workload with each interface.

(c) Mean usability with each interface.

(d) Mean likeabilities with each interface.

Figure 5.6: Objective and subjective results of the novice user study. Error bars represent standard error of the mean.

$\eta^2 = 0.841$. Planned contrasts using the LSD method were conducted to test for significant differences in Likability scores between conditions. The means, standard deviations, and statistically significant contrasts between conditions are presented in Table 5.1d.

Similar to the SUS results, on the likability measure, the Direct Manipulation condition was associated with higher SUS scores than all of the other conditions except the VR condition. Thus, Hypothesis H1 which stated that the DM condition would be associated with the highest Likability scores across all of the conditions was not supported. Instead, the VR condition had the highest Likability scores in comparison to all the other conditions, again lending strong support for Hypothesis H2. Finally, the difference in Likability scores between the PM and KM condition was statistically significant, where users rated liking interacting with the PM interface more than the KM interface, supporting Hypothesis H3.

5.6 Discussion

Overall, we found that the full VR interface was significantly better in both the objective and subjective metrics compared to the keyboard and monitor interface. It was faster, with an average improvement of 101 seconds (66% improvement), and was rated as having a lower workload and higher usability, as measured by the NASA-TLX and SUS measures, respectively. Additionally, the full VR interface was much more liked, with an average likability score of 6.11 (out of 7) compared to 2.06. This result supports Hypothesis H2 and is encouraging, as it implies that a user performing VR teleoperation tasks would be both faster and happier than if they were using a keyboard and monitor interface.

Interestingly, while the full VR interface was on average faster than the positional hand tracking with monitor interface, it was not significantly so. This implies that the positional hand tracking was more important to the task speed than the VR camera control. The workload was also not significantly different. The system usability, however, was highly significantly different. The full VR interface scored much higher on the SUS test, $M = 71.46$ compared to $M = 55.94$. This implies that although users were able to complete the task with the monitor, they found it more difficult to use than the VR interface, further supporting Hypothesis H2.

As expected, the VR interface was slower than direct manipulation of the arm. Direct manipulation allows the user to see the cups with their own eyes and move the robot with their own hands. The fastest time recorded for direct manipulation was 5.5 seconds, which we believe approaches the physical limit of the task. The workload score was also significantly lower, which may be due to the shorter times the users achieved with direct manipulation. Both the fast time to complete the teleoperation task and the low workload scores strongly supported H1. Surprisingly, however, the VR interface actually had a marginally higher SUS score compared to direct manipulation, $M = 71.46$ to $M = 71.25$. We believe this is because SUS measures the complexity, consistency, and ease of use of a system, not physical effort or objective success.

Participants failed the task when a cup was knocked over or dropped, leading it to roll out of reach of the robot. This happened the most with the keyboard and monitor interface. Five of the eighteen users were never able to complete the task with the keyboard interface. Two users were never able to complete the task with the positionally tracked controller and monitor, and all users completed the task with the VR interface at least once.

5.7 VR Teleoperation Task Feasibility

We considered desirable skills for a manipulator robot to have. Our goal was to answer two questions:

1. Is the robot physically capable of performing certain tasks?
2. If so, can a human teleoperating the robot in VR complete this task?

Because the physical capabilities of the robot depends on the hardware, our specific study used a research Baxter robot. Refer to Section 5.3.7 for more information.

In order to answer these two questions, two authors of this work acted as the expert teleoperators for performing the trials. For question one, we physically moved the robot's arms in real life to complete the task. Direct manipulation of the robot's arms gives users the best perception of the scene, along with direct haptic feedback from the robot and the environment. We used this methodology of Direct Manipulation in our previous VR study as a good measure for task feasibility [58]. For question two, we used our ROS Reality interface mentioned in Section 5.3 to perform VR teleoperation to complete the tasks.

Baxter's 7 DoF arms are equipped with parallel electrical grippers at the end effector, such that Baxter is effectively able to grip, push, pull, and rotate objects. However, Baxter's ability to grip objects is limited by the nature of its parallel electrical grippers, as well as the ability of its arms to exert push and pull forces.

We derived a set of 24 tasks by choosing different common manipulation tasks that could be relevant for manipulator robots in a variety of domains (e.g., home personal assistant uses, socially assistive applications), while simultaneously attempting to pick tasks that we believed possible to implement on Baxter. Two groups of tasks were chosen, such that one half of the tasks could be completed using one manipulator and the other half required use of both manipulators at the same time. In addition, tasks were chosen to represent an array of different movements (i.e., grip, push, pull, and rotate).

For each task performed via direction control and via ROS Reality, we performed a maximum of 5 attempts to complete the task. A given task was deemed feasible if we were able to complete it at least once. We report our results for the tasks in Table 5.2 and Fig. 5.7.

5.7.1 Discussion

Overall, VR control of the Baxter robot via ROS Reality was a success. For single manipulator tasks, eight out of twelve were achieved through direct manipulation, with seven of those eight tasks achieved through VR. For the two manipulator tasks, eight out of twelve were also achieved through direct manipulation, and again, seven of those eight tasks were completed through VR.

In general, the direct kinesthetic manipulation of the robot permitted the easiest, fastest completion of tasks for the tasks that proved physically possible for the robot. This is unsurprising, given the familiar nature of guiding a human on how to physically move to perform a task, as well as getting to directly observe the robot's workspace. The users found VR most useful when the task required complex movements of the robot's joints. During direct manipulation, resistance in the robot's manipulators forced the operators to use two hands to move the robot's limb. This meant the operator could effectively only move one manipulator at a time, and had to extend a fair amount of force to move the manipulator to a complex position. By contrast, in VR the user does not have to constantly parametrize joint angles of the robot, instead specifying end effector pose and having the robot calculate and navigate the correct trajectory.

Our trials revealed that force exerted by the robot was a limiting factor in whether or not tasks could be completed in the first place, with the robot unable to generate sufficient force for certain

Table 5.2: Task Feasibility Evaluations

Task	Description	Task Number	Direct?	VR?
Block Stacking	Stack ten 3x3cm wood blocks in a column.	1	Yes	No
Unscrew Bottle	Unscrew the cap to a bottle.	2	No	-
Uncap Marker	Remove cap from an Expo marker.	3	Yes	Yes
Hinge Board	Open all six latches on Melissa and Doug Latches Wooden Activity Board.	4	No	-
Stir Pot	Stir a wooden spoon in a metal pot.	5	Yes	Yes
Push Spacebar	Push the spacebar button on a keyboard.	6	Yes	Yes
Checker Piece	Pick and place a checker piece on a board.	7	Yes	Yes
Squeeze Purell	Squeeze out Purell from the bottle.	8	Yes	Yes
Connect 4 Piece	Insert a Connect 4 piece into the slot.	9	Yes	Yes
Toss Ball	Toss a juggling ball up and catch it in the same hand.	10	No	-
Use Fork	Get a piece of food onto a plastic fork.	11	Yes	Yes
Unzip Zipper	Unzip a loose zipper.	12	No	-
Open Chips	Open a plastic bag of chips.	13	No	-
Carry plate	Carry a plate with an item on it from one location to another.	14	Yes	Yes
Open Glass Bottle	Use a bottle opener to open a glass bottle.	15	No	-
Peel Potato	Use a peeler to peel the skin of a potato.	16	Yes	No
Uncap Marker	Remove cap from an Expo marker.	17	Yes	Yes
Dust Pan	Use a dust pan to sweep small blocks.	18	Yes	Yes
Fold Shirt	Fold a T-shirt.	19	Yes	Yes
Handover Expo	Handover a pen from one manipulator to the other.	20	Yes	Yes
Open Box	Open a shoebox.	21	Yes	Yes
Tap a Paradiddle	Tap to the rhythm of paradiddle.	22	Yes	Yes
Toss Ball	Toss a ball from one manipulator and catch in other.	23	No	-
Tie Shoelace	Tie a shoelace into a knot.	24	No	-

Table 5.3: List of tasks and performances. One manipulator tasks are above the line, while two manipulator tasks are below the line.

tasks (e.g., toss and catch ball, open chips). However, manipulation tasks that did not require substantial force were largely successful. Rotation did not pose a major obstacle to task completion. Manipulation tasks requiring dexterous grasping were also limited by the parallel electrical grippers on the robot used in our evaluation but are likely to be much easier for robots with higher DoF end effectors. Finally, the robot's built-in collision detection system prevented completion of the two manipulator task of opening the bag of chips, with the system preventing the robot's arms from coming close enough to grip the bag of potato chips on both sides of the bag.

5.8 Future Research

The system described in this chapter serves as the foundation for a host of future research. For instance, Learning from Demonstration (LfD) is a popular approach to teach robots complex manipulation tasks because it utilizes human expertise, judgment, and decision making. However, obtaining demonstrations from human participants in laboratory studies is both time and resource intensive. One approach to addressing this problem has been to develop algorithms that require

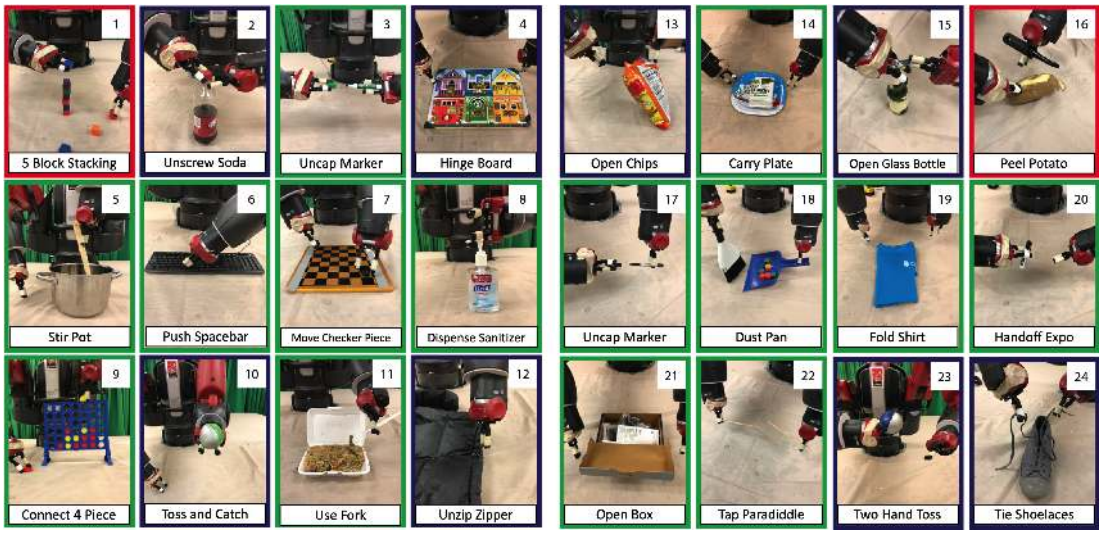


Figure 5.7: The results of the VR task attempts. Green outline indicates task was completed both kinesthetically and in VR, blue indicates the task could not be completed kinesthetically, and red outlines indicates that the task could be completed kinesthetically, but not in VR.

fewer and fewer demonstrations from humans; which is challenging, and will inevitably require at least one, if not more, demonstrations by human users with physical robots. Even with this solution, at some point for most tasks, users will have to interact with robots to help train them. However, using VR as a mechanism to gather LfD data at scale is a promising alternative. Learning complex tasks from task experts can be challenging for autonomous systems, with VR sidestepping this issue by enabling users to directly control systems while leveraging the benefits of the system. Demonstrations could be provided to virtual robots by users accessed over the Internet in a crowdsourcing paradigm, completing tasks at scale, and thus addressing participant and resource limitations that currently plague extant LfD training methods. VR coupled with ROS Reality has the potential to offer a cost and time-effective solution to this challenge.

5.9 Conclusion

Virtual reality is becoming increasingly available to everyday users, as hardware platforms steadily decrease in cost. These systems also represent an intuitive interface for controlling robots. In this chapter we offer an open-source VR teleoperation package, ROS Reality, that makes any ROS-enabled robot controllable by any Unity-compatible VR headset. This work also identifies and tests robot manipulation tasks using ROS Reality with a consumer-grade VR headset.

In the next chapter, we will adapt ROS Reality for use with mixed reality hardware, and design a system to intuitively communication motion intent from the robot to the human.

Chapter 6

Communicating And Controlling Robot Arm Motion Intent Through Mixed Reality Head-mounted Displays

This chapter presents a method for communicating and controlling robot motion via a mixed reality interface. It presents the first ideas of bidirectional human robot communication as described in Section 2.7. It comes from two previous works: a conference paper [61], which was subsequently invited for an expanded journal publication [83].

6.1 Introduction

Industrial robots excel at performing precise, accurate, strenuous, and repetitive tasks, which makes them ideal for activities like car assembly. A major drawback of these robots is that humans are unable to easily predict their motions, which forces most industrial robots to be isolated from human workers and restricts human-robot collaboration. This is especially true in a fluid working environment without rigidly-defined tasks, or where robots move autonomously. Although the intended robot motion is defined ahead of time through motion planning, efficiently conveying the intended motion to a human is difficult. Human-robot collaboration requires robots to communicate to humans in ways that are intuitive and efficient [84]; yet, the motion intention inference problem leads to many safety and efficiency issues for humans working around robots [85].

This problem has inspired research into how robots might effectively communicate intent to humans. Current interfaces for communicating robot intent have limitations in expressing motion plans within a shared workspace. Humanoid robots can try to mimic the gestures and social cues

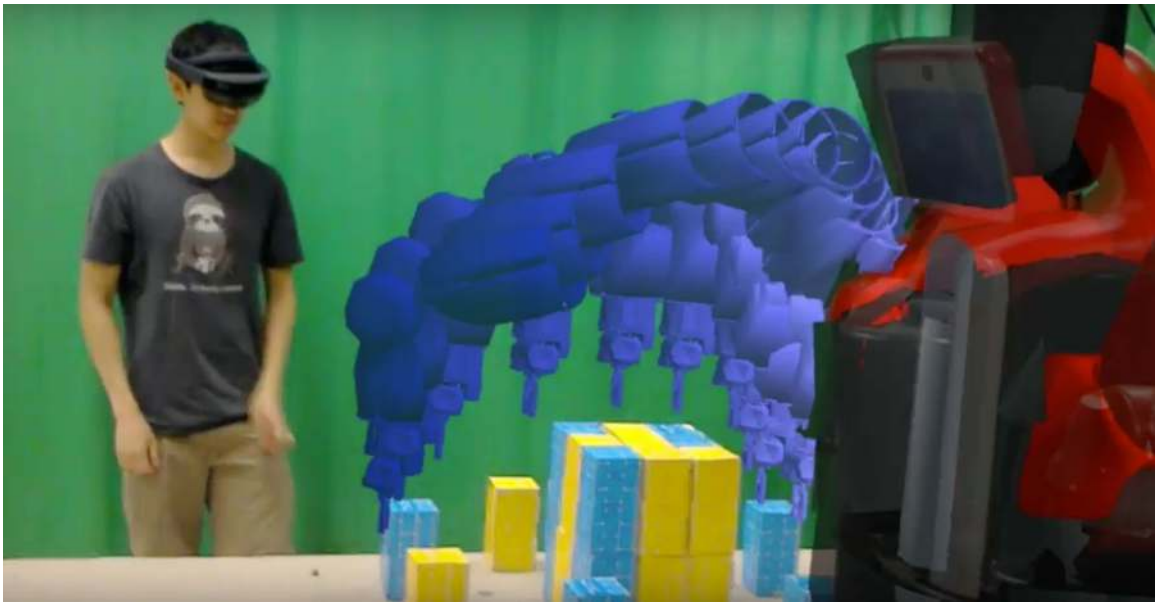


Figure 6.1: An image captured directly from the MR Headset of a user viewing a robot trajectory.

that humans use with each other, but many robots are not humanoid. The motion robots intend to make can also be visualized on a 2D display near the robot. This requires the human to take their attention away from the robot's physical space to observe the display, which could be dangerous. Additionally, a 2D projection of a 3D motion plan can take time for a human to understand, requiring interaction to inspect different points of view.

Natural communication might be achieved when humans can see a robot's future motion in the real world from their own point of view, via a head-mounted display [86, 87]. This could increase safety and efficiency as the human no longer needs to divert their attention. Further, as the 3D motion plan would be overlaid in 3D space, human users would not need to make sense of 2D projections of 3D objects.

We tested this idea with a system that enables humans to view robot intended motion via 3D graphics on a mixed reality (MR) head-mounted display (HMD)—the Microsoft HoloLens. This allows a participant to visualize the robot arm motion in the real workspace before it moves, preventing collisions with the human or with objects (Fig. 6.1). [1]

In addition to visualizing robot motion intent, it is important for the robot to be able to *replan* an intended trajectory based on human response, i.e., when the user notices that the planned robot trajectory will collide with objects in the environment. Using MoveIt [2], we allow a user to command the robot to plan new trajectories with the same start and end points, and so visualize and choose from different robot motion trajectories.

[1] As there is no existing open source HoloLens ROS integration for the robotics community, we have released our code: `https://github.com/h2r/Holobot`. This integrates HoloLens with the widely-used Unity game engine, provides a URDF parser to quickly import robots into Unity, and network code to send messages between the robot and HoloLens.

We experimentally compare our system to both a 2D display interface and a control condition with no visualization (Fig. 6.4). In a within-subjects-design study, 32 participants used all three system variants to classify arm motion plans of a Rethink Robotics Baxter as either colliding or not colliding with blocks on a table. Our MR system reduced task completion time by 7.4 seconds on average (a reduction of 38%), increased precision by 11% percent on average, and increased accuracy by 15% percent on average, compared to the next best system (2D display). Additionally, we improved subjective assessments of system usability (System Usability Scale) [80] and mental workload (NASA Task Load Index) [79]. This experiment shows the promise of mixed-reality HMDs to further human-robot collaboration.

6.2 Related Work

Humans use many non-verbal cues to communicate motion intent. There is much work in approximating these cues in humanoid robots, focusing especially on gestures [88] and gaze [89], as well as related work on non-verbal communication with non-humanoid robots [90]. However, robots often lack the faculty or subtlety to physically reproduce human non-verbal cues—especially robots that are not of human form. One alternative is to use animation and animated storytelling techniques, such as forming suggestive poses or generating initial movements [91]. This increases legibility; the ability to infer the robot's goal through its directed motion [92]. However, these methods still lack the ability to transparently communicate complex paths and motions. Further, tasks involving close proximity teamwork may require more detailed knowledge of how the robot will act both before and during the motion, such as in collaborative furniture assembly [86] and co-located teleoperation [93].

Verbal communication has also been shown to be an effective way to have robots communicate their high-level intent [94]. However, while speech is useful for quickly expressing abstract actions such as "I will rotate the table", it is difficult to communicate low-level actions such as what joint-angles the robot will assume throughout the planned motion. Not only is it cumbersome for the robot to explicitly state all of the relevant information for describing a high degree-of-freedom (DoF) arm motion, it is too much to expect humans to be able to easily interpret such speech because humans do not typically talk in this manner.

Other related works have used turn and display indicators on the robot to communicate navigational intent [95–97]. These techniques were found to improve human trust and confidence in robot actions, but did not express high detail in the motion plan [98, 99].

We can also use 2D displays to visualize the robot's future motions within its environment through systems like RViz [100, 101]. These require the human operator to switch focus from the real world environment to the visualization display [102]. This may lead the operator to expend more time understanding the robot state and environment rather than collaborating with the robot [103, 104].

6.2.1 Augmented and Mixed Reality for Human-robot Collaboration

We can adapt the real-world environment around the human-robot collaboration to help indicate robot intent. One way is to combine light projectors with object tracking software to build a general-purpose augmented environment. This has been used to convey shared work spaces, robot navigational intention, and safety information [105–107]. However, building special purpose environments is time consuming and expensive, with a requirement for controlled lighting conditions. Further, they exhibit occlusions of the augmenting light from objects in the environment, and limit the number of people able to see perspective-correct graphics.

Hand-held tablets can allow participants to view a mixed reality of 3D graphics overlaid onto a camera feed of the real world [108]. These types of approaches mediate the issue of diverted attention which 2D displays suffer. However, they limit the ability of the operator to use their hands while working, and there is a mismatch in perspective between the eyes of the human and the camera in the tablet.

Optical head-mounted displays can overlay 3D graphics on top of the real world from the point of view of the human. This has been hypothesized to be a natural and transparent means of robot intent communication, for instance, with the overlay of future robot poses [86, 87]. Hopefully such a system would reduce human-robot collaborative task time and produce fewer errors. The recent introduction of the Microsoft HoloLens has made off-the-shelf implementations of such a visualization possible. Previously, the HoloLens and other MR interfaces have been used in human-human collaboration, such as communicating with remote companions and playing adversarial games [109–111]. However, mixed reality as a tool to communicate robot motion intent for human-robot collaboration is nascent. Contemporary work investigates the use of mixed reality for communicating drone paths [112], but there is a lack of work dealing with multi-jointed, high degree of freedom robots. This inspired us to test the hypothesis that an MR HMD which allows participants to see visual overlays on top of real-world environment in human-robot collaborative tasks is more performant than existing approaches.

6.2.2 3D Spatial Reasoning in VR Displays

Since the HoloLens and similar devices are so new, there is little direct evidence to support their efficacy in robot intent communication. However, hypotheses may be informed from literature in the parallel technology of virtual reality (VR) which, in a similar way to mixed reality (MR), provides head tracked stereo display of 3D graphics to create immersion. In VR, 3D spatial reasoning gains have been tested [113]. Pausch et al. [114] found that head-tracked displays outperform stationary displays for a visual search task. Ware and Franck [115] found a head-tracked stereo display 3 times less erroneous than a 2D display for visually assessing graph connectivity. Slater et al. [116] measured performance gains in Tri-D chess for first-person perspective VR HMDs over third-person perspective 2D displays (like RViz). Ruddle et al. [117] found navigation through a 3D virtual building was faster using HMDs over 2D displays, though with no accuracy increase.

Not all experiments in this area favor large-format VR. Many prior works compare immersive head-tracked CAVE displays against desktop and 'fishtank VR' displays, and often smaller higher-resolution displays induce greater performance thanks to faster visual scanning [118, 119]. Santos et al. [120]. reviewed all HMD to 2D display comparisons in the literature until 2009, and found their results broadly conflicting. Then, they conducted their own comparison for 3D navigation: on average, the desktop setup was better than the VR HMDs.

In general, the relationship between VR display and task performance is one with many confounding factors. The benefits over traditional 2D desktop displays are task dependent, and no clear prescriptive guidelines exist for which techniques to employ to gain what benefit. As such, while we may assume that a mixed reality interface for viewing 3D would be better, the evidence from the VR literature tells us that the issue may be more complex.

6.3 Technical Approach

Communicating and controlling robot motion intent requires us to join our robot control system (ROS with ROS Reality) to a motion planner (MoveIt), and to visualize the result on a mixed reality head-mounted display (HoloLens with Unity) in a shared robot/headset coordinate system.

6.3.1 ROS and ROS Reality

ROS and ROS Reality have been covered in Chapter 4, particularly Sections 5.3.3 and 5.3.6. To see how ROS and ROS Reality were used for this system, see Fig. 6.2).

6.3.2 MoveIt

For robot motion planning, we use the MoveIt [2] software package, the most common motion planning software for ROS-enabled robots. Users are able to programmatically specify start and goal robot transforms to MoveIt from a ROS Node. With this, we need to both send desired pose information from the HoloLens to MoveIt, and receive back motion plans to visualize.

To receive the planned trajectory from MoveIt, the HoloLens directly subscribes to the */display_planned_path* ROS topic published by MoveIt. This topic contains a list of time-stamped joint angles that determine the trajectory visualization. To send poses to MoveIt, we send the poses to an intermediary node on the ROS network called the MoveIt Node (see Fig 6.2). This node receives the poses from the HoloLens and uses MoveIt's python API to create a planning service request to MoveIt.

6.3.3 Microsoft HoloLens and Unity

The Microsoft HoloLens is a standalone mixed reality headset which allows users to overlay digital imagery on top of the real world. This is accomplished with an inertial measurement unit, an array of four cameras, and an IR depth sensor, which combine to simultaneously map the environment and

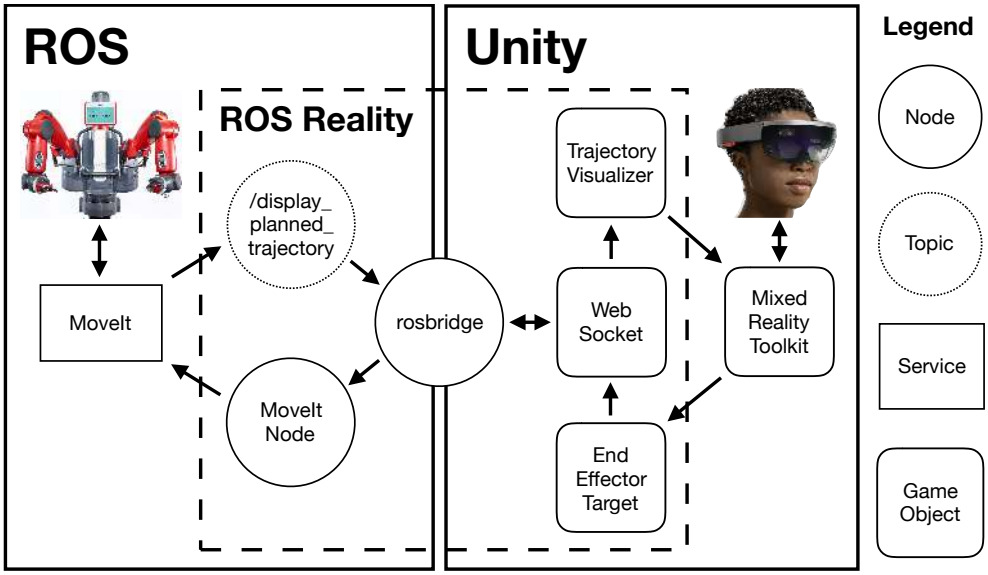


Figure 6.2: A schematic of our system. Human operators use the HoloLens to interact with a Unity scene using the MixedRealityToolkit, and specify robot end effector goal states using hand gestures. Goal poses are wirelessly communicated over rosbridge to a ROS network using ROS Reality. The MoveIt node receives the goal pose and sends it to MoveIt, which uses the current transform of the robot from /tf as the starting pose, and publishes a plan onto /display_planned_path topic. This motion plan is sent back over rosbridge to Unity, where the TrajectoryVisualizer renders the trajectory fed from the WebSocket client.

locate the headset inside of that map. The HoloLens supports the creation of mixed realities and gesture interfaces using the 3D game engine Unity [121] in conjunction with Microsoft's MixedReality Toolkit (MRTK) [122]. Unity applications are composed of scenes for human operators to interact within a virtual space. Operators perceive the scene through the MR headset and interact with it through hand gestures and voice commands.

ROS Reality contains functions for generating realistic Unity models of real ROS robots from URDFs. Additionally, ROS Reality allows the virtual robot model to mirror the live robot, and vice-versa. This provides natural situational and environmental awareness of the robot, plus robot control.

6.3.4 Interaction Walkthrough

The flow of an interaction using our system can be seen in Fig. 6.3 and is as follows:

0. *Once at startup:* Manually calibrate the MR-HMD coordinate system to the ROS coordinate system.

1. The user specifies a goal pose for each arm in the MR-HMD using gestures (see Fig. 6.3a and 6.3b).

2. Using speech, the user commands the MR-HMD to send the goal poses to MoveIt via ROS Reality, which computes a motion plan (Fig. 6.3c). Again via ROS Reality, this plan is sent back to the MR-HMD.

3. The human inspects the motion plan, visualized in the MR-HMD via Unity.

4. If the user approves of the trajectory, then the robot performs it (Fig. 6.3d). If not, then the robot repeats from step 2.

Step 0: To allow MR-HMD users to specify goal poses and visualize plans in the same workspace as the robot, the coordinate spaces of the virtual world in Unity and the real world robot must align. For this, we manually calibrate. When the MR-HMD app is launched, a life-size virtual version of the robot is displayed. The MR-HMD hand-tracking capabilities enable the user to "grab" the virtual robot and align its position and rotation such that the virtual robot is in the same place as the real robot. This defines a rigid transformation between the two coordinate spaces.

An automatic calibration system is also possible, e.g., using QR tags to calibrate (or constantly re-calibrate) the transformation. However, for this system we settled with the manual approach.

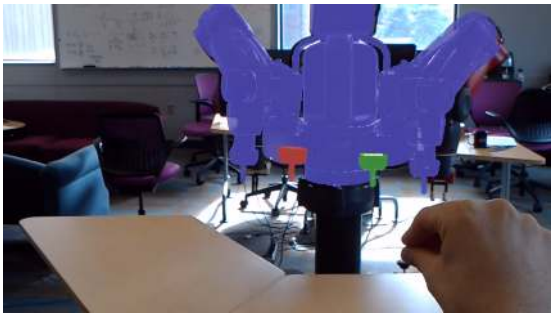
Step 1: After calibration, Unity and ROS have the same coordinate systems. To specify end effector poses for the robot, our interface uses virtual robot grippers (one for each arm) to represent the goal position and rotation. Using two hands, users move and rotate the virtual robot grippers by gesturing in free space.

Step 2: With the goal poses set, the users says 'plan' (or uses a button), triggering the MR-HMD to send this information to MoveIt through the intermediary MoveIt Node. MoveIt calculates a motion plan from the current robot pose to the user-specified goal pose. This plan is sent back to the HoloLens via rosbridge (Fig. 6.2).

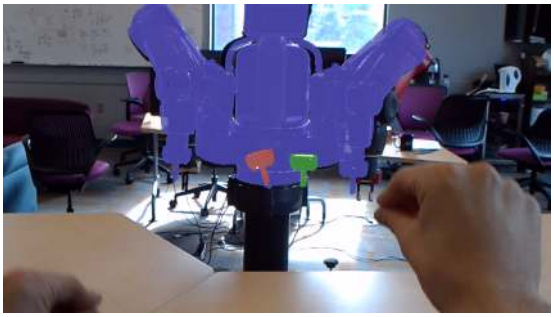
Step 3: In our Unity scene, we have a GameObject that acts as a WebSocket client (Fig. 6.2) and interfaces with rosbridge. As trajectories are streamed from MoveIt, the WebSocket client stores them so that they can be used by the TrajectoryVisualizer GameObject for visualization. The two possible visualizations are a looping animation or a sparse static trail. A full discussion of our visualization techniques can be found in Sec. 6.3.5.

Step 4: The user decides whether the proposed trajectory from MoveIt is acceptable or not. The user approves by saying 'move', and the robot performs the motion. If the user disapproves, then they can repeat step 2 again, and MoveIt will replan a new trajectory. Because we use a stochastic planner, it would be very unlikely to see the same trajectory twice. The user can also go back to step 1 and adjust the goal poses. This process enables both users and robots to communicate motion intent.[2]

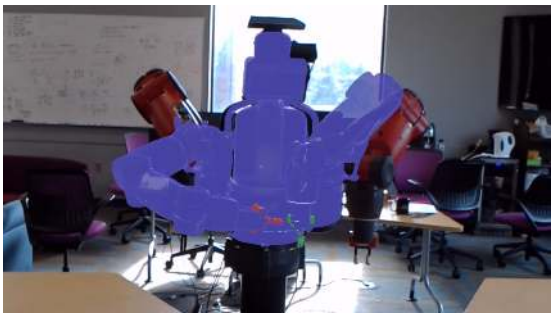
[2]Please see our supplemental video: `http://h2r.cs.brown.edu/videos/`



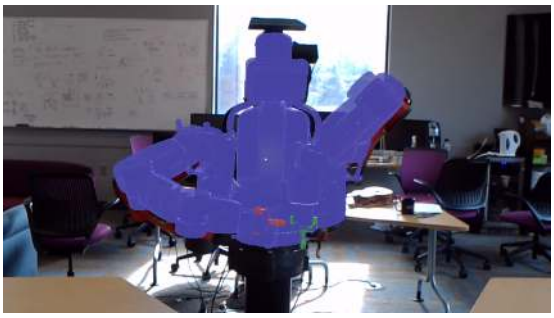
(a) The user specifies the goal position for the trajectory. The red and green models represent the right and left goal poses, respectively. The user is currently gesturing with one hand to translate the right arm (red model) towards the center.



(b) The user specifies the rotation of the red goal pose by using a two-handed gesture to rotate the model.



(c) With the goal poses specified, the user says "plan" to visualize the intended motion. (In this case, in the form of an animation).



(d) After inspecting and agreeing with the motion plan, the user says "move" to cause the real robot arms to execute the motion and move to the desired position.

Figure 6.3: An example interaction using our system. A human specifies goal poses of the robot end effector, and the robot visualizes the resulting trajectory generated by its motion planner. Following human approval, the robot executes the motion.

6.3.5 Visualization Design

Any visualization must consider the amount of information conveyed and the ease and efficiency of comprehension. Further, any design must consider hardware efficiency, too: The limited computing power of the HoloLens constrains the amount and quality of 3D models visualized, else the rendering and localization loops will slow down and create inaccuracy and virtual/real visual mismatch.

Visualization designs span a large gamut (see RViz [101] for examples). We could repeatedly play an animation of the planned motion in real time, which conveys all information but is slow to comprehend. We could show all poses of the motion at once as a continuous trail, which looks cluttered as it is somewhat redundant, and is computationally inefficient. At the other end, we could visualize only the planned end effector trail, which would be very efficient, but would provide incomplete information on intermediate arm joint locations which may collide with the world.

We drew inspiration from the visualization options provided in the RViz GUI to MoveIt. In that interface, users can toggle between an animation of an arm moving through the trajectory and a sparse stroboscopic trail made of multiple arms sampled along the trajectory. For either option, the virtual arm can either be the color of the real robot, or a different, user-specified color.

We implement all of the discussed visualization options in our package. Animation was initially our chosen technique, as it most limits the number of needed draw calls compared to the other options. Unfortunately, this comes at the expense of user comprehension. In our initial testing, we found users needed to watch the animation loop multiple times closely inspect the entire trajectory. For our study, we settled on the sparse trail option from RViz, with two major modifications. First, we had to reduce the polygon count of our virtual arms due to rendering bottlenecks on the HoloLens. Second, we used a light-to-dark color gradient on the trail to emphasize the direction of the motion plan (Fig. 6.4).

6.4 Experiment

We can now test whether MR HMDs can aid motion intent communication between humans and robots. We focused on robot-to-human communication, and so goal pose adjustment was not evaluated in this study as it pertains to human-to-robot communication. We asked novice participants to decide whether or not a robot arm motion plan would collide with blocks on a table using three interfaces: no visualization, an RViz-like 2D display visualization, and our MR visualization. Our evaluation used 32 participants (15 male, 17 female) with ages ranging from 20 to 55 ($M = 26, SD = 6.8$). We measured task completion time and true/false positive/negative rates as objective metrics, as well as the subjective assessments of system usability, likability, and workload via the System Usability Score (SUS) and NASA Task Load Index (TLX) questionnaires.

6.4.1 Task

In each interface, we presented each participant with the same set of 14 robot arm motions in a random order. These motions each moved from a start point to an end point over a table covered in blocks. We did not allow users to specify goal poses and used prerecorded trajectories of the robot's arm rather than use a motion planner in the loop to repeatably present the same motions to each of the participants. Unknown to the participant, exactly half of the motions collided with the blocks and half did not. Each participant was tasked with labeling the motions as either colliding or non-colliding as quickly and accurately as possible. The blocks were assembled such that it would be difficult to obtain a complete view of all blocks from just one perspective due to occlusion from other blocks. Participants could walk around to view the environment from different perspectives. Once a participant had decided how to classify a particular motion, they pressed a button on an Xbox controller to indicate their decision, allowing us to measure the time it took for them to decide.

6.4.2 Interfaces

We compared three interfaces (Fig. 6.4):

- **No visualization:** This simulated a participant super-vising a robot with an emergency stop button. Partici-pants watched the arm, and pressed an Xbox controller button to stop the arm if they thought it would collide.
- **Monitor:** Participants viewed and interacted with a 2D monitor on a desk. The visualization consisted of: 1) a 3D model of the robot, 2) a sparse trail of its future arm poses, and 3) a 3D point-cloud of the environment, captured by a Kinect v2 sensor mounted near the robot. In this interface, the robot arm did not move. Participants could move the virtual camera in the visualization to gain different perspectives using a keyboard-and-mouse-based control scheme (the control scheme was the same as in RViz [101]). The visualization remained on the screen for the entire trial. For consistency, participants again recorded their assessment using an Xbox controller.
- **Mixed Reality (MR):** Through a HoloLens, participants viewed the same visualization of the motion plan overlaid on top of the real world. In this case, there is no need to visualize the environment via a point cloud because the participant can see it directly. Users walked around the room to change their perspective of the robot and visualization. Like the other interfaces, participants decided upon whether the motion collided or not, and recorded their prediction using an Xbox controller. Like in the monitor interface, the robot arm did not move, and the visualizations remained for the entire trial.

Note that the no visualization interface differs from the monitor and mixed reality components because the arm moves. We move the arm because asking the participant to judge whether the arm will collide in the future with no clues whatsoever is pure guesswork. However, moving the arm makes it less comparable to the visualization components, especially in the case of measuring task time.

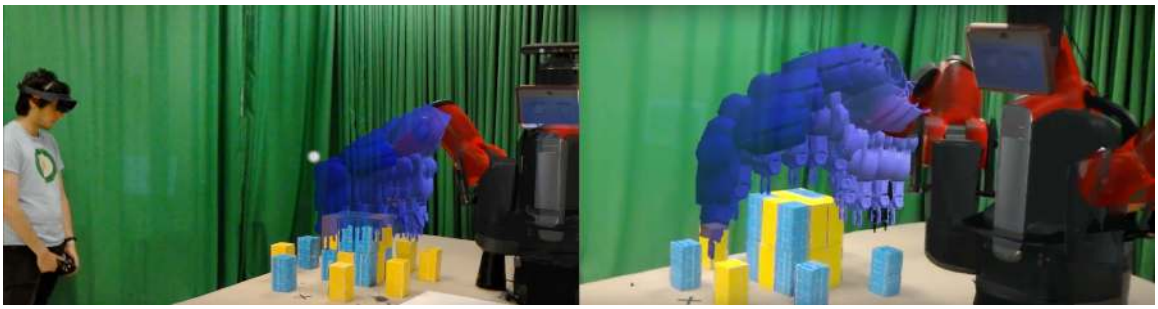
Given that our main interest was to evaluate the effectiveness of the mixed reality based visualization, we consider the no visualization interface to be a less direct comparison than the monitor.

6.4.3 Experimental Procedure

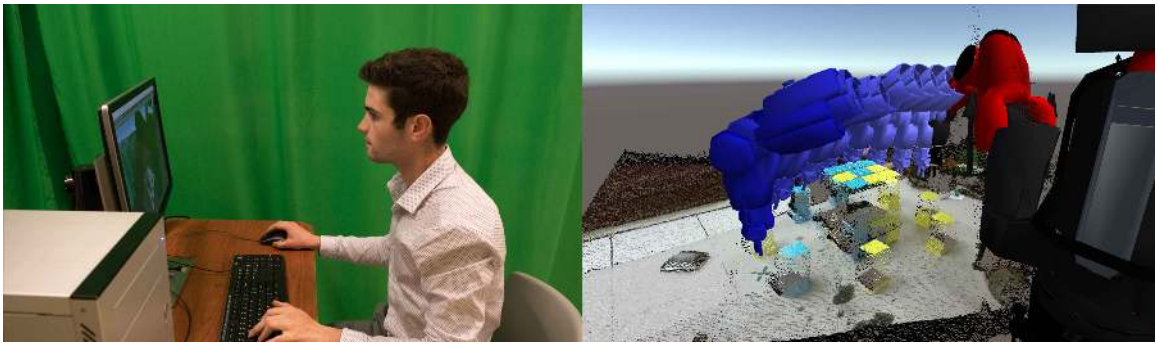
We began by reading a consent document to the participant. After consenting, participants completed our motion intent task using all three interfaces. The no visualization condition was always completed before the other two interfaces. Participants received instruction to hit the stop button if and only if they thought the arm was going to collide with a tower. We started the arm moving after a 3-2-1 countdown.

The monitor and MR interfaces then followed. We counterbalanced the order in which participants completed the monitor and MR conditions. Participants were randomly assigned to complete

(a) Mixed Reality visualization—view captured directly from MR Headset.



(b) 2D display visualization—an RViz-like interactive 3D scene.



(c) No visualization.

(d) Robot motion over time.

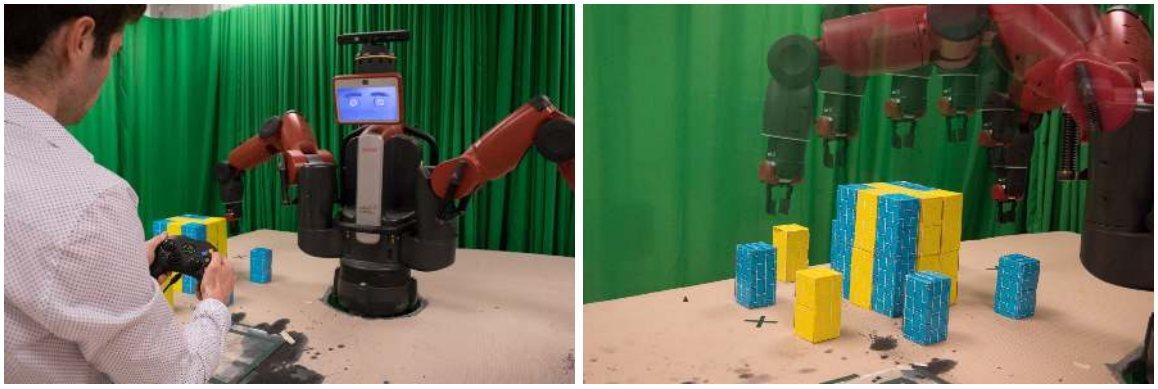


Figure 6.4: Participants must decide whether a robot motion plan collides with the light yellow and blue blocks on the table, across 14 trials and three interfaces. Our three interfaces are an MR visualization (a), a 2D display/mouse with an RViz-like visualization (b), and no visualization at all (c). In the first two cases, the experimental setup is shown on the right, and the participant view on the left. In (a), the HoloLens visualizes the robot arm motion plan as a sequence of blue virtual arm graphics overlaid onto the real world. In (b), the 2D display uses the same visualization, but the participant must use the system at a desk. In (c), the no visualization condition, the participant directly observes the robot arm move and pushes a 'stop' button on an Xbox controller if they think collision will occur. (d) shows what a robot motion over time would look like in the no visualization condition.

one of the two counterbalancing conditions. For the MR and monitor conditions, participants received instructions to label the robot's planned motion as quickly and accuracy as possible. Then, after a 3-2-1 countdown, we displayed the visualization. After completing the task for all 14 robot arm motions with each interface, the participant completed three questionnaires.

6.4.4 Measurements

We chose the choice of interface as the within-subjects independent variable. In all three interfaces, our objective dependent variables were the true and false positive rates of classifying a path as colliding, and the true and false negative rates of classifying a path as non-colliding. By using the mean adjusted accuracy (d') metric, we also accounted for participant strategy in labeling each motion as colliding or non-colliding (e.g., showing a tendency to always label a motion plan as colliding). This is discussed further in Sec. 6.4.6.

In the monitor and MR interface conditions, we also measured the average speed of labeling each motion plan by recording the time elapsed from first seeing the visualization of the planned path to labeling the path. This allowed us to measure the accuracy and precision with which each interface allowed participants to label the robot's intended motion.

Our subjective dependent variables were participant workload as measured by the NASA Task Load Index (NASA-TLX) questionnaire [79], system usability as measured by the System Usability Scale (SUS) questionnaire [80], and our own questionnaire measuring perceived predictability and preference for each interface. For a full description of each measure, see Section 5.5.5.

6.4.5 Hypotheses

We expected that participants would show the best performance in the Mixed Reality interface condition followed by the monitor interface (i.e., highest true positives/negatives, least false positives/negatives, lowest levels of mental workload, highest usability, predictability, and system preference scores). Additionally, we hypothesize that participants would have a faster labeling speed with the MR interface compared to the monitor interface.

- **H1:** MR will be the easiest interface for completing the motion labeling task, as demonstrated by participants achieving the best performance out of the three conditions, across (a) highest true positives/negatives, (b) lowest false positives/negatives, (c) lowest levels of workload, (d) highest usability scores, and (e) highest predictability and preference scores.

- **H2:** The monitor interface will be easier for completing this task than using no visualization at all. This will be demonstrated by participants achieving better performance than with no visualization, across (a) higher true positives/negatives, (b) lower false positives/negatives, (c) lower levels of workload, (d) higher usability scores, and (e) higher predictability and preference scores.

- **H3:** The MR interface will have faster labeling times than the monitor interface, as demonstrated by the average time it took for participants to label each motion as colliding or not colliding. Labeling times in the monitor and MR conditions are a function of evaluating the visualization of the planned robot motion, whereas in the no visualization condition, labeling times are generated by watching the robot enact the planned motion. As such, only the monitor and MR conditions are directly comparable.

6.4.6 Results

To test our hypotheses, several tests for significant differences between group mean scores (e.g., repeated measures ANOVA, paired samples t-test) on the measures of performance, workload, usability, and perceived predictability with planned comparisons between the three conditions were used. Specifically, ANOVA with planned comparisons were conducted to test for significant differences on the dependent measures of performance, workload, usability, and perceived predictability between the HoloLens condition, the monitor condition, and the no visualization condition. A t-test was used to test for significant differences in mean motion labeling times between the monitor condition and the HoloLens condition. Finally, response frequencies were computed to investigate participant interface preference.

Analysis Techniques

We used repeated measures analysis of variance (ANOVA) and signal detection theory (SDT) to determine if differences between measures in the three conditions were significant at the 95% confidence level. While ANOVA is likely to be familiar to the reader, SDT is less likely to be familiar, and so we will describe its use.

SDT describes accuracy in human perception and decision making tasks by taking into account preferences for responses [123, 124]. For instance, in our task, always responding that a motion plan will collide would yield high true positive scores ("hits"), and also high false positive scores ("false alarms"). In decision making tasks with innocuous false alarms, adopting this strategy would not affect overall performance. However, for tasks with high false alarm cost, a strategy that results in low false alarm rates while retaining high hit rates is better. For HRI tasks like ours, false alarms would slow the collaboration considerably and so we consider them high cost.

In SDT tasks, d' (also called sensitivity) is a common measure which considers decision making strategy. It is the standardized difference between the hit rate and the false alarm rate. To handle perfect scores (i.e., correctly labeling all the colliding and non-colliding paths), zero false alarm scores, and zero hit scores, we adopted the technique outlined by [125].

Accuracy

We counted the number of participant true positives, false positives, true negatives, and false negatives in each condition. From this, we report accuracy as the proportion of true positives plus

true negatives out of the total number of motion plans (Fig. 6.5a). MR was the most accurate (M= 0.76, SD= 0.19), followed by the monitor (M= 0.66, SD= 0.14), followed by the no visualization condition (M= 0.60, SD= 0.12). These differences were statistically significant (Wilks $\Lambda = 0.619$, $F(2, 30) = 9.244$, $p = .001$, $\eta^2 = 0.381$), and accuracy in the MR condition was significantly better than in the monitor condition ($p = .001$) and the no visualization condition ($p < .001$). Performance in the monitor condition was not significantly better than in the no visualization condition ($p = .065$).

We also report d' scores for each participant in each of the three conditions (Fig. 6.5b). There was a significant difference in d' performance scores between the conditions (Wilks $\Lambda = 0.523$ $F(2, 30) = 13.675$, $p < .001$, $\eta^2 = 0.477$). Further, the performance in the MR condition (M= 1.79, SD= 0.88) was significantly better than the monitor condition (M= 0.94, SD= 0.58) and the no visualization condition ($M = 0.79, SD = 0.72$), all with $p < .001$. The difference between performance in the monitor condition was not significantly better than performance in the no visualization condition (p= .38). A look at the mean accuracy and mean d' scores showed that performance in the MR, monitor, and no visualization conditions trended in the hypothesized direction although both performance indicators in the monitor condition were not significantly better the no visualization condition. Thus, hypotheses 1 (a) and (b) were supported, but hypotheses 2 (a) and 2 (b) were not supported.

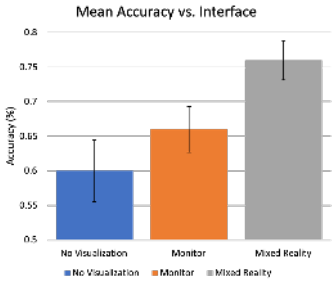
Finally, as a manipulation check, verified that participants who completed the no visualization condition followed by the monitor condition and then the MR condition (*Order 1:*, $M = 0.67$, $SD = 0.16$) did not have significantly different accuracy scores than participants who completed the no visualization condition followed by the MR condition then the monitor condition (*Order 2:*, $M = 0.68$, $SD = 0.17$), $t(94) = .220$, $p = .826$. The same was true for the d' scores (*Order 1:*, $M = 1.14$, $= 0.82$; *Order 2:* $M = 1.21$, $SD = 0.89$), $t(94) = 0.428$, $p = 0.669$.

Task Time

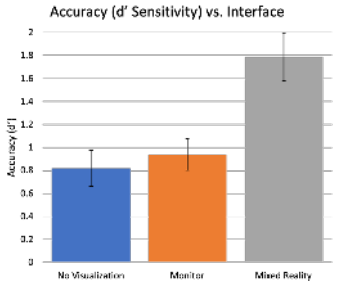
Hypothesis 3 stated that motion labeling times would be faster in the MR condition than in the monitor condition. A paired samples t-test showed significant differences in mean motion labeling times between the two conditions ($t(31) = 3.415, p < .001$). Mean labeling times trended in the hypothesized direction (Fig. 6.5c). Labeling times in the MR condition were significantly shorter ($M = 11.95, SD = 8.42$) than in the monitor condition ($M = 19.39, SD = 19.28$). Hypothesis 3 was supported.

Subjective Workload

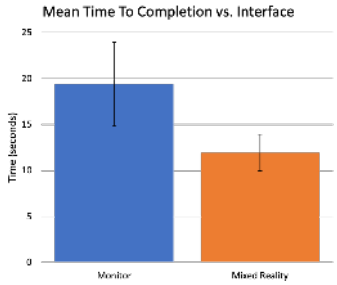
Hypotheses 1 (c) and 2 (c) stated that workload would increase from MR to monitor, and from monitor to no visualization. We used one-way repeated measures ANOVA to test for statistical significance in workload scores across the three interface conditions (Wilks $\Lambda = 0.802$, $F(2, 30) = 3.693$, $p = 0.037$, $\eta^2 = 0.198$).



(a) Mean accuracy across interfaces. The mixed reality interface is significantly more accurate than the other two interfaces.

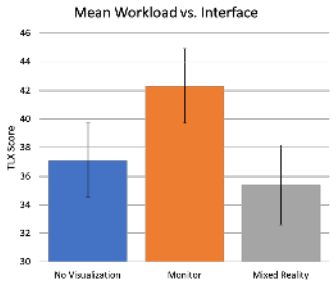


(b) Mean adjusted accuracy (d') across interfaces. The mixed reality interface has a significantly better d' compared to the other two baseline interfaces.

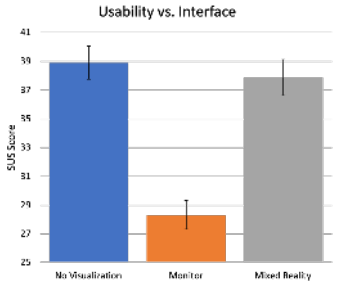


(c) Mean task times for comparable interfaces (see **H3** definition). The mixed reality interface is significantly faster than the monitor.

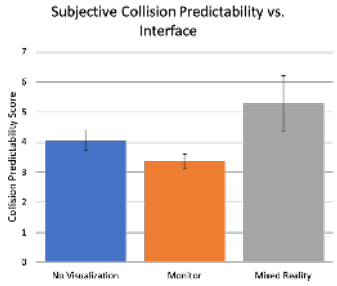
Figure 6.5: Objective measure user study results. Error bars represent standard error.



(a) Mean NASA-TLX scores across all interfaces. Participants reported the lowest levels of subjective workload in the MR condition, significantly lower than in the monitor condition.



(b) Mean SUS scores across all interfaces. The monitor interface had significantly lower usability scores than the other interfaces. All interfaces were significantly different from one another.



(c) Mean subjective collision predictability scores across all interfaces. The mixed reality interface score was significantly higher than the other two interfaces. The monitor and no visualization conditions were not significantly different.

Figure 6.6: Subjective questionnaire user study results. Error bars represent standard error.

The MR condition was associated with the lowest workload scores ($M = 35.39, SD = 15.73$), followed by the no visualization condition ($M = 37.11, SD = 14.78$), and then the monitor condition ($M = 42.32, SD = 14.71$; Fig. 6.6a). Post hoc comparisons showed that mean scores in the MR condition were significantly lower than in the monitor condition ($p = .040$). There was no significant difference in workload scores between the MR condition and the no visualization condition. The difference between workload scores in the monitor condition and the no visualization condition were not significantly different. Hypotheses 1 (c), which stated that MR would have the lowest workload scores, was partially supported. Hypothesis 2 (c) was not supported as the workload scores in the monitor condition were higher than in the no visualization condition.

Subjective Usability

Hypotheses 1 (d) and 2 (d) stated that MR would have the highest usability scores, followed by monitor, followed by no visualization. A one-way repeated measures ANOVA, showed that there was a significant difference in mean usability scores across the three conditions (Wilks $\Lambda = 0.151$, $F(2, 30) = 84.342$, $p < 0.001$, $\eta^2 = 0.849$). However, the no visualization condition was associated with the highest SUS scores ($M = 38.91, SD = 6.52$), followed by the MR condition ($M = 37.88, SD = 7.10$), and the monitor condition ($M = 28.31, SD = 5.62$; Fig. 6.6b). Mean SUS scores in the MR condition were significantly higher than the monitor condition ($p < 0.001$), and mean SUS scores in the no visualization condition were significantly higher than the monitor condition ($p < 0.001$). The difference between the MR condition and the no visualization condition was not significant. Hypotheses 1 (d) and 2 (d) were not supported.

Subjective Collision Predictability

Hypotheses 1 (e) and 2 (e) stated that the ordering of highest collision predictability scores would be MR, then monitor, then no visualization. We used one-way repeated measures ANOVA to test for significant differences in participants' assessments of whether or not they felt the interfaces could help them predict collisions. There were significant differences between the interfaces on this measure (Wilks $\Lambda = 0.246$ $F(2, 30) = 45.891$, $p < 0.001$, $\eta^2 = 0.754$). Participants showed the highest agreement that MR helped them to predict collisions ($M = 5.28, SD = 1.11$), followed by the no visualization condition ($M = 4.06, SD = 1.95$), and then the monitor condition ($M = 3.38, SD = 1.31$; Fig. 6.6c). The difference between mean scores in the MR condition were significantly higher than in the monitor condition and the no visualization condition (both $p's < .05$), supporting Hypothesis 1 (e). Means scores in the monitor condition were lower than the no visualization condition but not significantly so ($p = .15$). Hypothesis 2 (e) was not supported.

Subjective Enjoyment

We compared the frequencies with which participants selected each interface as the one they enjoyed the most, the one they preferred for completing the task, and the one they felt made understanding the robot's motion the easiest. All participants selected MR as the interface they enjoyed the most ($N = 32$). For the interface participants felt made understanding the robot's motion the easiest, almost all of the participants selected MR ($N = 29, 90.6\%$), while only three participants (9.4%) selected the monitor. Finally, when asked about preference for completing that task, almost all participants selected MR ($N = 30, 93.8\%$). Only two participants (6.3%) selected the monitor interface as their preferred interface for completing the task. No participants selected the no interface condition.

6.5 Discussion

Overall, our results demonstrate the potential benefit of MR to communicate robot motion intent to humans. Participants in the MR condition significantly outperformed the monitor condition, showing a 15% increase in collision prediction accuracy and a 38% decrease in time taken. Mixed reality also allowed participants to outperform the control condition of no visualization. Almost universally, participants selected MR as the most enjoyable interface, the easiest for completing the task, and the one they preferred for assessing the robot motion plans. Taken together, these findings strongly support our hypotheses that MR would be associated with the best objective performance measures.

As MR-HMDs are a novel technology, it would be unsurprising for there to be a corresponding novelty effect in our subjective enjoyment measures. However, considering the objective benefits of the MR interface, we feel it is unlikely to be the only cause of the reported subjective enjoyment.

An examination of participant free responses regarding why they preferred MR over monitor offers some insight into these findings. Many participants reported that using the monitor and mouse to virtually move around the robot was cumbersome, unintuitive, difficult to manipulate, distracting, and confusing. Participants reported that MR was not perfect: the motion plan overlay was not always correctly aligned on top of the robot due to the manual calibration and due to inaccuracy in HoloLens tracking (we noticed a drift of several centimeters over a long period of use), the set up took a long time, and physically moving around the robot was sometimes difficult. Even so, 34% of participants reported that they liked that they could freely move around the robot to see the planned motion, and that this made determining whether or not collisions would occur faster, easier, and more intuitive than when using the monitor and mouse.

The subjective questionnaire responses offered mixed but promising support for the MR condition. Although participants working with the MR condition reported lower workload than in the no visualization condition, it was not significantly lower, which offered only partial support for hypothesis H1 (c). The mean workload scores did trend in the hypothesized direction as the MR condition had the lowest workload scores overall, and the results suggests that participants did not find the MR interface more taxing than using no interface at all. Although participants rated the no visualization condition as slightly more usable than the MR condition (counter to hypothesis H1 (d)), the no visualization condition was not rated significantly more usable. The similarity of SUS and NASA-TLX scores between the no visualization and the mixed reality condition was somewhat surprising, as the interfaces are extremely different. It's possible that the increased cognitive load of interpreting the mixed reality visualizations was offset by the increase in ease of task resulting from those visualizations.

Perhaps surprisingly, the monitor condition did not significantly outperform the no visualization condition for both objective and subjective measures. Participant accuracy (and accuracy accounting for decision making strategy) was not significantly better, and when working with the computer monitor, participants reported higher workload and lower assessments of usability than when working with the no visualization condition. Put another way, looking at a robot with an emergency stop

button in your hand is about as simple an interface as you could build. Finally, participants also reported the least agreement that the monitor interface could help them to accurately predict robot collisions. Thus, no part of hypothesis 2 was supported.

As consideration for future work, our system only considers one method of human-robot motion intention communication, and alternative methods may prove effective. Mixed reality can also be used to communicate other things beside motion intent, such as shared goals, needed objects, or other aspects of robot state.

In addition, further user studies could be conducted to evaluate the effectiveness of using the MR-HMD for communicating motion intent in scenarios with varying cost of mistakes. In our study, we instructed users to label the trajectories as quickly and accurately as possible, but did not directly penalize the users for mislabeling the trajectories. Real world situations that have actual costs associated with making mistakes may more heavily rely on having an interface that has higher usability for conveying information.

Although this study evaluated the effectiveness of using MR-HMDs for communicating robot motion intent, we did not evaluate the use of our system for enabling users to adjust trajectories, measuring the effectiveness of human-to-robot communication. Future work will address different methodologies of allowing end-users to interact directly with the planned trajectories, such as the end-effector goal pose specification described in Section 6.3.4, or perhaps a broader system that allows for fine-grain and high level adjustment.

6.6 Conclusion

If robots and humans are to form fluid cooperative work partnerships, we will need efficient communication and control of robot motion. We describe a system to allow mixed reality visualizations of robotic motion intent, an interface to control robot motion using mixed reality, and a user study investigating the hypothesis that mixed reality would be a natural interface for robot motion intent communication. We found that both participant performance and participant perceptions were improved with an MR visualization over the more traditional monitor interface for visualization and over no visualization at all. Our results provide evidence that mixed reality is one way to bridge the robot-human motion communication gap.

In the next chapter, we will combine the results of previous chapters into one cohesive model that combines the high bandwidth communication channels of mixed reality with the multimodal observation capabilities of POMDPs.

Chapter 7

Mixed Reality as a Bidirectional Communication Interface for Human-Robot Interaction

This chapter presents the Physio-Virtual Deixis (PVD) POMDP, a decision theoretic model for bidirectional communication between a human and robot via mixed reality. It is the second multimodal social feedback POMDP this thesis presents (see Section 2.4).

7.1 Introduction

Communicating human knowledge and intent to robots is essential for successful human-robot interaction (HRI). For example, when a surgeon says "hand me the scalpel," it is crucial that the assistive robot hand over the correct instrument. In order to efficiently collaborate, humans intuitively communicate through modalities such as language, gesture, and eye gaze. Failures in communication, and thus collaboration, occur when there is mismatch between two agents' mental states.

Having a robot infer a human's mental state is difficult because the observations the robot uses are noisy, especially in ambiguous situations. Tracking the human's gesture with an RGB-D (Red, Green, Blue plus Depth) sensor, such as the Microsoft Kinect, is subject to errors from hardware imperfections and environmental conditions. Speech-to-text software is imperfect and sometimes inaccurate. Using human eye gaze is noisy because it requires high-precision tracking of rapid movement.

Question-asking allows a robot to acquire information that targets its uncertainty, facilitating recovery from failure states. However, all question-asking modalities have tradeoffs, making choosing which to use an important and context-dependent decision. For example, for robots with "real" eyes (like the Nexi [126]) or pan/tilt screens (like the Baxter [127]), looking requires fewer joints to move less distance compared to pointing, decreasing the speed of the referential action. However, eye

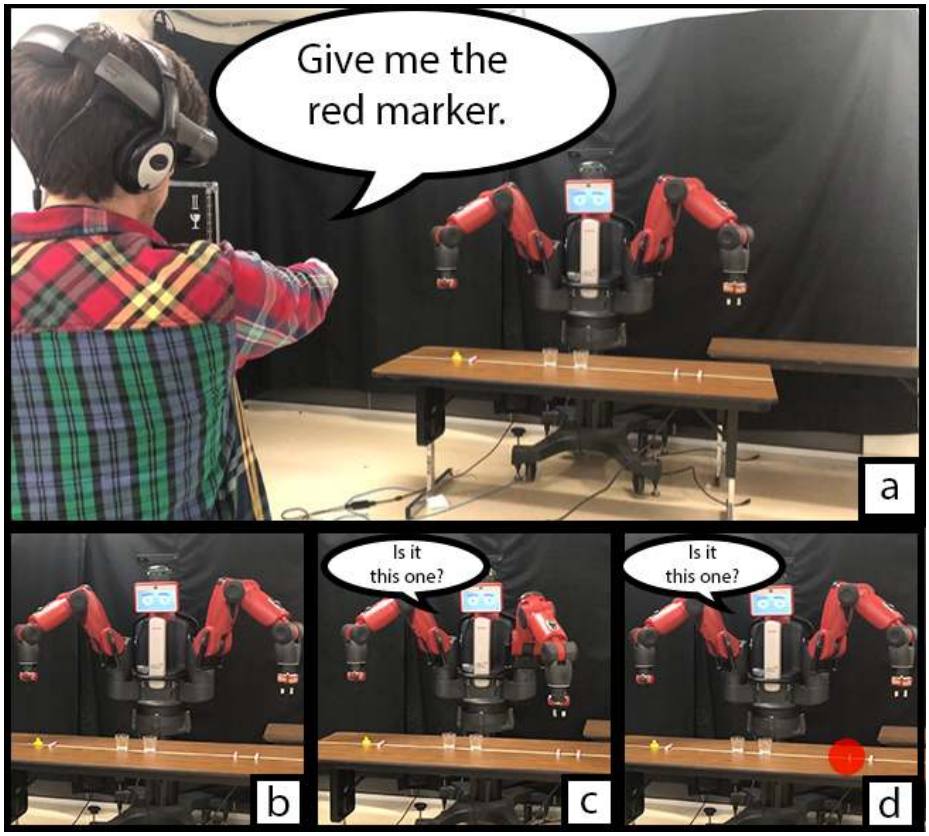


Figure 7.1: An example interaction. In (a), the participant first uses speech, pointing, and eye gaze to ask for the red marker. Then the participant experiences one of three conditions: In (b), the no feedback control condition, the robot waits for more information before choosing. In (c), the physical feedback condition, the robot asks about the red marker via pointing. In (d), mixed reality feedback condition, the robot asks about the red marker via highlighting with a 3D sphere in mixed reality.

gaze is inherently more difficult to interpret. On the other hand, Mixed Reality Head-Mounted Displays (MR-HMD), which have been shown to reduce mental workload in HRI [61], can indicate items quickly, is very accurate given proper calibration, and is independent of the physical robot. However, visualizations may distract the user's attention more than a typical pointing or looking action. Furthermore, MR technology is still new, and users may prefer to instead interact with a robot that performs physical actions. We aim to close the gap of research on how MR compares to physical actions for reducing robot uncertainty.

This work investigates how physical and visualization-based question-asking compare for reducing robot uncertainty under varying levels of ambiguity (Fig. 7.1). To do this, we first model our problem as a POMDP, termed the Physio-Virtual Deixis POMDP (PVD-POMDP), that observes a human's speech, gestures, and eye gaze, and decides when to ask questions (to increase accuracy) and when to decide to choose the item (to decrease interaction time). Then, we conduct a between-subjects user study, where 83 participants interact with a robot in an item-fetching task. Participants experience

one of three different conditions of our PVD-POMDP: a no feedback control condition, a physical feedback condition, or a mixed reality feedback condition. Our results show that our mixed reality model significantly outperforms the physical and no feedback models in both speed and accuracy, while also achieving the highest usability, task load, and trust scores.

7.2 Related Work

Previous research has investigated different communication modalities between robots and users, identifying the costs and benefits of each. A large amount of work has investigated physical robot actions used to reference objects to communicate with a human user, with two effective modes being robot eye gaze and robot pointing. Other research has opted instead to utilize a visualization-based approach, with visualizations displayed through 2D monitors, augmented reality, and mixed reality.

Eyes tend to move very quickly, and are used to both collect and communicate information. This makes eye-tracking a natural way to ground the references of other agents [128–135]. However, it is often difficult to perceive where an agent is looking, especially compared to pointing. Pointing is another natural deictic gesture that requires more effort but is easier to interpret. Admoni et al. [133] show that gaze and gesture are good at distinguishing between locationally unambiguous (far apart) items, while speech is good at distinguishing between visually unambiguous (different looking) items. However, related works [128–135] do not compare using eye gaze and pointing gestures to visualizations for reducing robot uncertainty.

Language has also been shown to be an effective means of symbol grounding, as in Chai et al. [136]. Their system enables users to use natural language to describe objects in the shared environment in order to ground them. The authors use a NAO robot with pointing and language to ask questions to clarify the human's references. Having the robot act in order to share its uncertainty to the human was shown to be important for establishing common ground. As in their work, we investigate pointing and language for disambiguation. However, we also investigate eye gaze, visualization, and question asking for mediating human-robot interaction.

Shridhar and Hsu [137] develop their own system, INGRESS, to interpret unconstrained natural language commands for unconstrained object class references. The authors also integrate question-asking for disambiguating symbol grounding by using a two-stage neural network to estimate what relevant visual features are in the scene, and then decide what object is being referenced. Their system outperforms state-of-the-art baselines, though they recognize that integration of nonverbal commands would help with requiring less complicated verbal references. Our approach, in contrast, uses a relatively simple language model, but also incorporates human gesture and eye gaze. Our model also allows the agent to ask questions via gesture, eye gaze, and visualizations for disambiguation.

Sibirtseva et al. [138] perform a comparison of different visualization techniques for robot question-asking in an item-fetching domain. The authors use a semi-wizarded system to compare a 2D monitor interface, an augmented reality interface (fixed overhead projector), and a mixed reality interface

for highlighting tabletop items. The authors found the mixed reality interface most engaging, but augmented reality most accurate and most preferred. They posit that technical limitations were to blame for the poor performance of MR. Our approach, in contrast, directly compares MR visualization to physical behaviors such as pointing and eye gaze.

Williams et al. [139] propose a framework to study virtual, augmented, and mixed reality deictic gestures for HRI. The authors devise three classes for their framework: (a) *egocentric*, where robots use their physical bodies to act, such as pointing; (b) *allocentric*, where the mixed reality visualizations are generated from the perspective of the display user; and (c) *perspective-free*, which present visualizations over the environment independent of the user's perspective, such as a projector. This work lays a foundation for classifying mixed-reality deictic gestures, where our physical behaviors fall under egocentric and our virtual question-asking approach falls under allocentric. Williams et al. [140] builds on their previous work [139] by conducting a user study to evaluate how effective allocentric approaches paired with natural language descriptions are for communicating robot references, and found that the allocentric approach was more accurate than using ambiguous complex noun phrases for referencing the object. Our work extends theirs, comparing physical behaviors to virtual behaviors for item disambiguation in an interactive, question-asking setting, using both objective measures (speed, accuracy), and subjective measures (usability, trust, and workload).

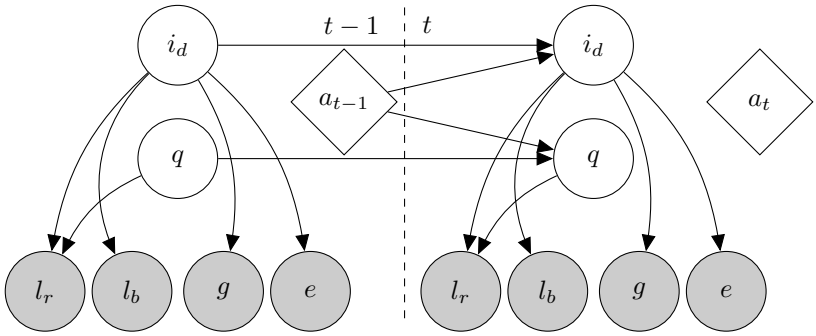


Figure 7.2: A graphical model of the PVD-POMDP. Hidden variables are white, observed variables are gray.

7.3 Technical Approach

We take a decision-theoretic approach to the item fetching problem by modeling our domain as a POMDP. This allows our robot to intelligently balance the informativeness and speed of its actions and gracefully handle its uncertainty. The Physio-Virtual Deixis Partially Observable Markov Decision Model, or PVD-POMDP, has as observations the speech, pointing gestures, and eye gaze of the user. Depending on the condition, the model enables our robot to look at, point to, and/or virtually highlight an item to ask if it is the desired item. The general intuition of our actions is that robot pointing is slower because the robot arm must move, but can be interpreted easily. Robot looking

is faster because only the face and screen move, but is more difficult to interpret than pointing, especially when items are close together. MR visualizations are just as interpretable as pointing gestures because MR isolates items via highlighting, yet is faster to perform than robot looking because it requires no robot motion.

7.3.1 Model Definition

The PVD-POMDP[1] (Physio-Virtual Deixis POMDP) is given by components $\langle I, S, A, T, R, \Omega, O, \gamma \rangle$

- I is the list of all items on the table. Each item $i \in I$ has a known location (x, y, z) and set of associated words $i.\texttt{vocab}$.
- $S : i_d \in I$ is the human's desired item, which is hidden. q is the agent's last question, which is known. q is initialized to `null`. The state is (i_d, q).
- A: We divide the actions into two types: non-question-asking and question-asking. The non-question-asking actions are *wait* and *pick*(i) for $i \in I$. A *pick* action ends the interaction. The question-asking actions are *point*(i), *look*(i), and *highlight*(i) for $i \in I$. *look* is cheaper but less accurate than *point*, while *highlight* is cheaper than *look* and as accurate as *point*.
- $T(s, a, s')$: i_d remains constant throughout an interaction. q is initialized to `null` and updated to a whenever a question-asking action a is taken.
- $R(s, a)$: The agent receives large positive and negative rewards for picking the right and wrong item respectively, and small negative rewards for all other actions. In decreasing magnitude of reward, the non-pick actions are *point*, *look*, *highlight*, *wait*. We calibrate these rewards roughly accordingly to how long each of the non-pick actions take: physical actions like *point* and *look* require physical robot behavior, thus take more time. *highlight* only needs to visualize on the MR-HMD, thus costs less. *wait* takes very little time.
- Ω: Each observation is composed of language, gaze, and gesture. Language is subdivided into base and response utterances. The response utterance can be positive, negative, or `null`.
- $O(o, s, a)$: The observation function can be factored into base utterance, response utterance, gaze, and gesture components. It is explained in detail in the Observation Model section below.
- γ: The discount factor is $\gamma = 0.99$.

7.3.2 Observation Model

Each observation o is a quadruple of base utterance l_b, response utterance l_r, gesture g, and eye gaze e. The components are assumed conditionally independent of each other given the state s (see Fig. 7.2):

$$\Pr(o \mid s) = \Pr(l_b \mid s) \Pr(l_r \mid s) \Pr(g \mid s) \Pr(e \mid s). \tag{7.1}$$

[1] See supplemental video for system demonstration: `https://youtu.be/5FmfntezYQE`

Following Goodman and Stuhlmüller [141], we assume each base utterance l_b has a literal interpretation probability $\Pr_{lex}(i_d \mid l_b)$ and that the speaker chooses their utterance by soft-max optimizing the probability that the listener infers the correct desired item from their base utterance. Each base utterance is interpreted as a vector l_b whose i^{th} component $l_b(i)$ is the number of words in the utterance that refer to the i^{th} object. Let U be the set of base utterance vectors and $|l_b| = \sum_{i \in I} l_b(i)$. Then we set:

$$\Pr_{lex}(i_d \mid l_b) = \begin{cases} \dfrac{(1-\alpha)l_b(i) + \alpha}{(1-\alpha)|l_b| + \alpha|I|} & |l_b| > 0 \\ \dfrac{1}{I} & |l_b| = 0, \end{cases} \tag{7.2}$$

where $\alpha = 0.02$ is a noise parameter. Let $p_l = 0.1$ be the probability a base utterance is made and $\theta = 15$ the soft-max parameter. Then:

$$\Pr(l_b \mid i_d) = \begin{cases} p_l \dfrac{e^{\theta \Pr_{lex}(i_d|l_b)}}{\sum_{l_b \in U} e^{\theta \Pr_{lex}(i_d|l_b)}} & |l_b| > 0 \\ 1 - p_l & |l_b| = 0. \end{cases} \tag{7.3}$$

When planning, we assume each base utterance will have at most three words to lower computation time.

The equation for $\Pr(l_r \mid s, a)$ has three components. The probability of receiving a response is $p_r = 0.6$. The probability that the human interprets the agent's question as asking about i if the agent is asking about j is $\Pr_*(i \mid j)$, which is defined in Equation 7.4 for *point* and *highlight*, and in Equation 7.6 for *look*. The probability that the human responds correctly based on their interpretation is $p_{rc} = 0.999$.

The human is assumed to always understand a *point* or *highlight* action, so the interpretation probabilities for pointing and highlighting are:

$$\Pr_p(i \mid j) = \Pr_h(i \mid j) = \begin{cases} 1 & i = j \\ 0 & i \neq j. \end{cases} \tag{7.4}$$

The interpretation probability for the *look* action uses a modified version of the model from Admoni et al. [133]. While humans have trouble identifying the exact angle of a *look*, they are very good at determining the general direction because of the robot's head motion, so we assume the human never mistakes a leftward *look* for a rightward *look* and vice versa.

Let $\text{ang}(i, j)$ be the angle between item i and item j relative to the robot's face, d_i the distance from the agent's face to item i, and $w_0 = 6$, $w_1 = 6$ noise parameters. Let $M(i, j)$ represent whether items i and j are on the same side of the robot:

$$M(i,j) = \begin{cases} 1 & i \text{ and } j \text{ are on the same side of the robot} \\ 0 & \text{otherwise.} \end{cases} \tag{7.5}$$

Then the probability $\Pr_l(i \mid j)$ that a human thinks the robot is looking at i when they are in fact looking at j is:

$$\Pr_l(i \mid j) \propto \frac{1}{d_i(1 + w_0|\text{ang}(i,j)|)^{w_1}} M(i,j). \tag{7.6}$$

Suppose the robot asked about item i using *point* or *highlight*. Then probability of receiving a response l_r is:

$$\Pr(l_r \mid s) = \begin{cases} p_r p_{rc} & l_r = \text{yes} \\ p_r(1 - p_{rc}) & l_r = \text{no} \\ 1 - p_r & l_r = \texttt{null}. \end{cases} \tag{7.7}$$

Let $\Pr_l(i)$ denote $\Pr_l(i \mid i)$. If the robot asked about item i using *look*, then the probability of receiving a response l_r is:

$$\Pr(l_r \mid s) = \begin{cases} p_r(\Pr_l(i)p_{rc} + (1 - \Pr_l(i))(1 - p_{rc})) & l_r = \text{yes} \\ p_r(\Pr_l(i)(1 - p_{rc}) + (1 - \Pr_l(i))p_{rc}) & l_r = \text{no}. \end{cases} \tag{7.8}$$

Human eye gaze e is modeled as a vector from the user's head to the point they are looking at. Gesture g is modeled as a vector from the user's the hand to the point they are pointing at. Angles are measured relative to the vector ending at the desired item. The probabilities of receiving a gaze or gesture are $p_e = 0.8$ and $p_g = 0.3$ respectively. When present, gaze and gesture are assumed to come from Gaussian distributions with mean 0 error and with and standard deviations $\sigma_e = 0.02$ and $\sigma_g = 0.06$ radians respectively:

$$\Pr(g \mid i_d) = \begin{cases} p_g \mathcal{N}(\theta_{i_d}; 0, \sigma_g^2) & g \neq \texttt{null} \\ 1 - p_g & g = \texttt{null} \end{cases} \tag{7.9}$$

$$\Pr(e \mid i_d) = \begin{cases} p_e \mathcal{N}(\theta_{i_d}; 0, \sigma_e^2) & e \neq \texttt{null} \\ 1 - p_e & e = \texttt{null}. \end{cases} \tag{7.10}$$

A human's gaze is attracted to referenced items, so the robot ignores gaze observations for 1 second after asking a question.

Due to the differing noise models combined with a decision-theoretic approach, the robot considers pointing to be more costly than looking, and thus will only point at an item when the increased accuracy is worth the cost. Roughly speaking, the robot will look at an item if it is far enough away from other items that looking is unambiguous and will point at an item when it is in close proximity to other items.

7.3.3 Implementation Details

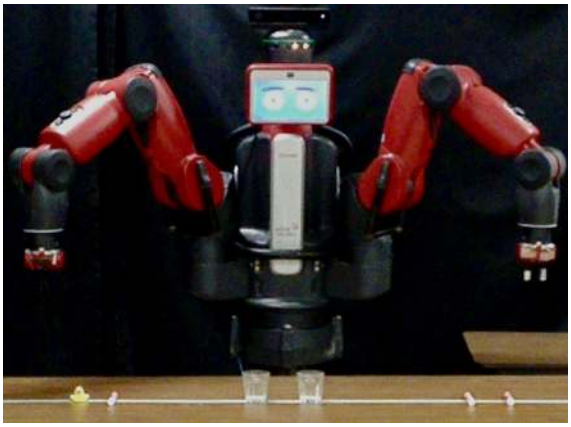
To observe the human's speech, we use Google's Cloud Speech [142] to transcribe the user's speech. For gesture tracking, we use the Microsoft Kinect v2 in conjunction with OpenNI's skeleton tracker software [143], and calculate pointing vectors from the user's head to hand. Lastly, we use the Magic Leap One [144], a MR-HMD, to track eye gaze.

We used Perseus, an offline POMDP planner from Spaan and Vlassis [145], as our planning algorithm. It took 6, 5191, and 724 seconds to train the control, physical, and mixed reality paradigms, respectively. Since human gesture and gaze are analogous, we planned using only gaze, but utilized both gaze and gesture during interaction.

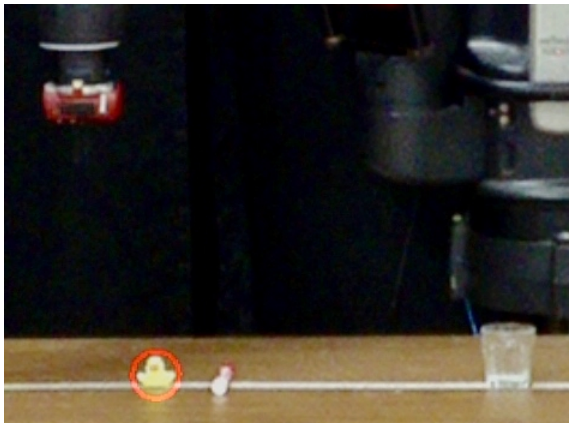
7.3.4 Visualization Design

For the user to understand which item the robot is asking about, the visualization presented to the user must isolate the referenced item from all the others. However, there are various designs that can be used to isolate the item. As part of our design process, we iterated over several visualizations via a series of small pilot studies (Fig. 7.3). In these studies, we would display our current set of visualizations to the user and ask for open-ended feedback on each.

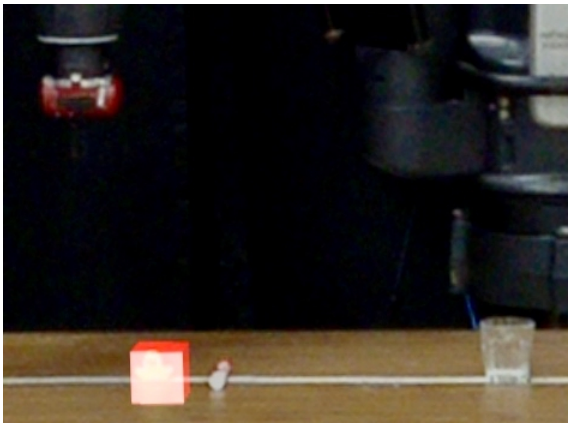
Our first design choice was to place an annulus (flattened ring) around the referenced item (Fig. 7.3b). This is similar to related works that have compared various interfaces for visualizations [138].



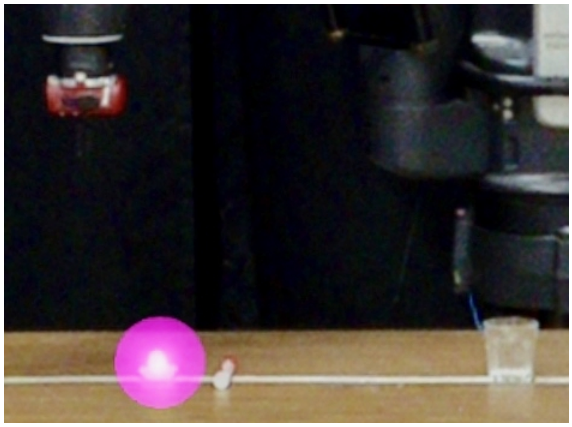
(a) All six objects on the table.



(b) Flattened ring visualization.



(c) Three dimensional cube visualization.



(d) 3D sphere visualization.

Figure 7.3: Image of items in (a), with images of potential visualizations (b, c, d). Our final design choice was the 3D sphere (d).

This visualization method is attractive because it is clear which item is being referenced (the item inside the annulus) and the item remains visible through the hole in the annulus. However, if the user moves around the scene to inspect items, the annulus may no longer be facing the user correctly, and thus no longer completely encircling the item. This can be solved by using a "billboarding effect," which causes the annulus's orientation to be tied to the user's head pose, so that as the user moves around, the annulus turns with them. However, in our pilot studies, users found the billboarding effect distracting or unsettling. Users also reported that a 2D shape felt out of place in our 3D world. Thus, in order to avoid applying a billboarding effect to our visualizations, we sought out a 3D visualization.

Our second design choice was to visualize a 3D cube over the referenced item (Fig. 7.3c). This design choice resolved our previous billboarding issue, since the 3D cube was able to be viewed from arbitrary angles, unlike a 2D design. However, users reported that as they moved around to look at items, the changing edges of the visualization would distract their eye gaze. Thus, we decided that we wanted a 3D design that was invariant under rotations.

Our third design choice was to use a 3D sphere visualized over the referenced item (Fig. 7.3d). A sphere is the only fully rotationally invariant 3D shape, so it can be viewed equally well from all angles. We found that during our pilot studies, users were less distracted when they moved, and generally looked directly at the item. We found 3D spheres to be the most highly regarded design in our pilot studies, and chose it as our final visualization method.

7.4 Evaluation

To evaluate our hypothesis, we designed an evaluation task where the robot disambiguated what item the human referred to as quickly and accurately as possible from an array of potential objects on a table in front of the robot. We defined interaction time as the time elapsed from when the robot first hears the human request an item to when the robot decided which item to pick, or after 30 seconds if it had not yet picked. Accuracy was calculated as the percentage of trials the robot correctly picked the human's desired item. The aim of our evaluation was to investigate how communicating questions via physical robot behaviors, like looking and pointing, compare to communicating those questions via mixed reality visualizations.

We devised a user study to compare three conditions of communication modalities. In the *no feedback control condition*, the robot did not ask any questions and only decided to pick an item when it was sufficiently confident based on observations from the human. In the *physical feedback condition*, the robot was able to ask questions by moving, either using gesture or looking to reference items. In the *mixed reality (MR) feedback condition*, the robot was able to ask questions by visualizing a sphere over the referenced item in the user's mixed reality headset. We posited two hypotheses (H1 and H2) about the objective measures, and two hypotheses (H3 and H4) about the subjective measures:

- **H1:** The feedback conditions (physical and MR) will outperform the no feedback condition

(control), as demonstrated by: (a) greater trial accuracy and (b) lower trial time.

- **H2:** The MR feedback condition will outperform the physical feedback condition, as demonstrated by: (a) greater trial accuracy and (b) lower trial time.
- **H3:** Users in the feedback conditions (physical and MR) will have a better user experience than users in the no feedback condition (control), as demonstrated by: (a) greater usability scores, (b) greater trust scores, and (c) decreased workload scores.
- **H4:** Users in the MR feedback condition will have a better user experience than users in the physical feedback condition, as demonstrated by: (a) greater usability scores, (b) greater trust scores, and (c) decreased workload scores.

7.4.1 Physical Configuration

The physical configuration of our experiment can be seen in Fig. 7.3a. For the interaction, the human stood 2 meters away from a table with six items on it, and the robot stood on the other side of the table. Our item set consisted of three red expo markers, two glass cups, and one yellow rubber duck. The expo markers and glasses were identical except for their different spatial positions. The items were placed on the table in three groups of two, with the rubber duck and a marker on the far left, the two glasses in the middle, and the last two markers on the far right. The distances between the objects, from left to right, were 10cm, 40cm, 15cm, 45cm, and 10cm.

We chose the items and their locations to represent visually and spatially ambiguous scenarios. Specifically, the leftmost group is least ambiguous, as the duck is a unique item, and the marker is very far from its identical copies. The middle group is more ambiguous, as the two glasses are identical, and are somewhat close together. The rightmost group is most ambiguous, as the two markers are identical, and very close together.

The Microsoft Kinect v2 sensor was placed on top of the robot and calibrated to accurately track the pose of the human relative to the robot. The user wore the Magic Leap One HMD and headphones with a microphone in order to track the user's eye gaze and speech, respectively. The user heard the robot's question-asking through the headphones.

7.4.2 Experimental Procedure

Participants were randomly assigned to one of the three between-subjects conditions (no feedback control condition, physical feedback condition, MR feedback condition). After reading the IRB approved consent procedure, we calibrated the Magic Leap One for each user's eye gaze by using the supplied visual calibration program. We then went through the instructions for the study, and informed users there would be 18 trials with the robot. For each trial, the user was told an item number associated with an object and instructed to use speech, gesture, and eye gaze to reference the item to the robot in a clear and natural manner. If the user was in a condition with feedback, the user was told what feedback to expect from the robot (i.e., either physical or MR visualization-based

question-asking). The experimenter then counted down from three to start the trial, at which point the user could reference the item; each trial ended when the robot selected an item or 30 seconds had passed. Every user was asked to reference each of the six items three times, totaling 18 trials. The order of items was randomly shuffled for each user. In each of the trials, we recorded the interaction time and whether the correct item was selected or not. After all 18 trials were completed, the user completed a series of subjective questionnaires.

7.4.3 Objective Measures

The performance of the robot in the task was evaluated using two objective measures, accuracy and time.

Accuracy

Accuracy was calculated as the number of correct items selected by the robot divided by the total number of trials (18 trials). We treated a trial timeout as an incorrect pick when calculating accuracy.

Time

Each trial began when the robot heard the user speak and ended when the robot picked an item; if the robot did not pick an item, the trial timed out after a 30 second period. The time measure was calculated as the average time of the interaction across all 18 trials.

7.4.4 Subjective Measures

Participants completed a series of three questionnaires to evaluate the success of the interaction on the basis of the task load of the interaction, the perceived usability of the system, and the trust in the robot on the task. The task load and usability were measured using the NASA Task Load Index and the System Usability Scale, previously described in Section 5.5.5.

Multi-Dimensional-Measure of Trust

The Multi-Dimensional-Measure of Trust (MDMT) was developed by Ullman and Malle [146] to assess human trust in robots across tasks and domains. There are two superordinate dimensions of the MDMT: moral trust and capacity trust. For this study, we were interested in user evaluations of capacity trust in the robot. We used two of the four subscales from the MDMT: reliable (reliable, predictable, someone you can count on, consistent) and capable (capable, skilled, competent, meticulous). The MDMT consists of rating scales for each item from 0, "Not at all," to 7, "Very," with an option for "Does Not Fit." We calculated capacity scores for participants by averaging across ratings on these eight items.

Condition	Accuracy	Time	SUS	MDMT	TLX
CTRL	.72±.20	7.59±4.05s	65.74±20.34	4.45±1.33	29.88±16.74
PHYS	.82±.17	8.10±2.86s	73.27±15.66	4.90±0.99	24.13±12.84
MR	**.93±.07**	**5.07±1.25s**	**76.76±12.71**	**5.40±0.85**	**19.01±11.37**

Table 7.1: The means and standard deviations of all five of our metrics for all three conditions (CTRL = Control, P = Physical, MR = Mixed Reality). Bolded numbers are the best for that metric.

7.5 Results

Participants were recruited from population of Brown University, with participants required to be at least 18 years old and able to see without glasses (contacts were acceptable). We first conducted a pilot study with 10 participants to test the system. We then conducted the main study with a total of 83 participants. Three participants were excluded from data analysis (two for failure to follow study instructions, and one due to system technical error). Analysis was performed on the data from 80 participants: 27 in the no feedback control condition, 26 in the physical feedback condition, 27 in MR feedback condition. Please see Fig. 7.4 and Table 7.1 for data.

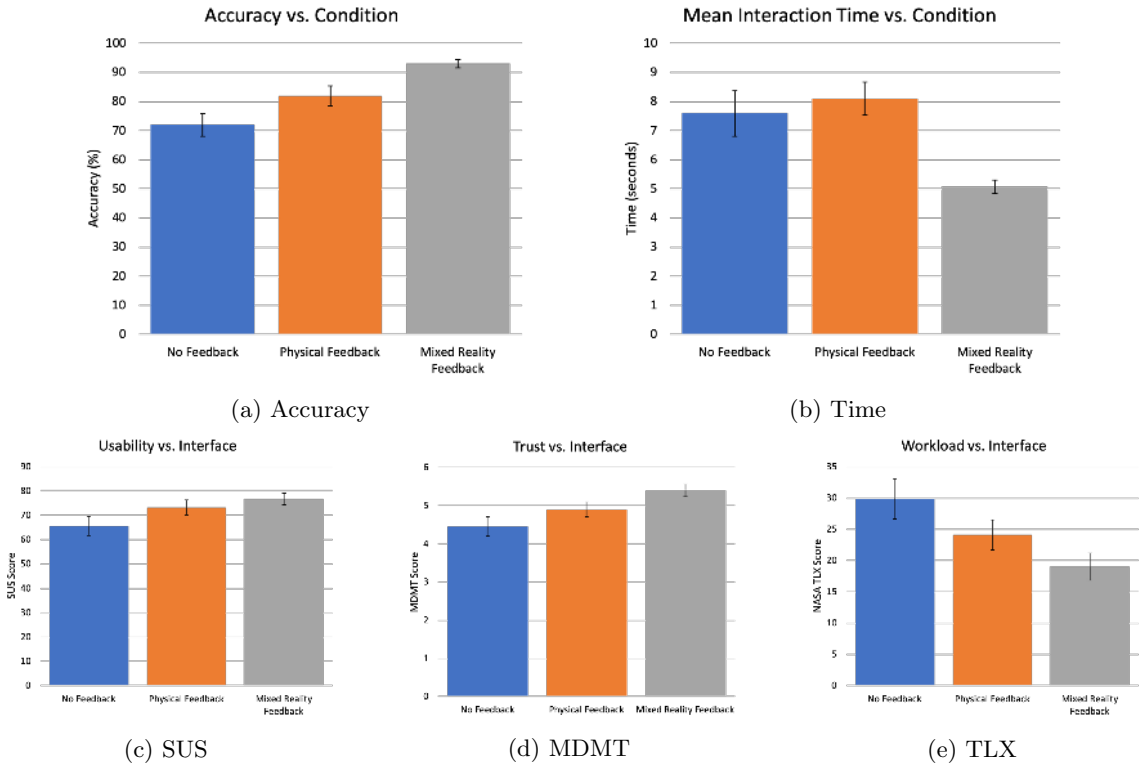
7.5.1 Objective Measures

The two objective dependent measures (accuracy, time) were correlated ($p < .001$) with each other, r = -.68. This correlation suggests that as accuracy increased, time for the task decreased. The correlation between the dependent variables also indicates that a multivariate analysis of the data is warranted to account for the relationship between the dependent variables.

A MANOVA was conducted using a pair of a priori orthogonal Helmert contrasts in order to test hypothesis H1 (that the feedback conditions would outperform the no feedback condition) and hypothesis H2 (that the MR feedback condition would outperform the physical feedback condition). An examination of the multivariate relationships of the data reveals strong support for hypothesis H1: the feedback conditions outperformed the no feedback condition. There was also strong support for hypothesis H2: the MR feedback condition outperformed the physical feedback condition.

The first Helmert contrast was significant and supports hypothesis H1, $F(2, 76) = 10.14$, $p < .001$, *multivariate* $\eta^2 = .21$. The univariate F-tests revealed that, compared to the no feedback condition, the feedback conditions were (a) higher on accuracy, $F(1, 77) = 17.69$, $p < .001$, $\eta^2 = .19$; and (b) not statistically significant different for time, $F(1, 77) = 2.21$, $p = .14$, $\eta^2 = .03$. These results indicate that the effect of increased performance in the feedback conditions is driven by higher accuracy.

The second Helmert contrast was significant and supports hypothesis H2, $F(2, 76) = 6.86$, $p < .01$, *multivariate* $\eta^2 = .15$. The univariate F-tests reveal that the MR feedback condition outperformed the physical feedback condition with (a) significantly higher accuracy, $F(1, 77) = 5.80$, $p = .02$, $\eta^2 = .07$; and (b) significantly lower time, $F(1, 77) = 13.91$, $p < .001$, $\eta^2 = .15$. These results indicate that the MR feedback condition was superior to the physical feedback condition.



(a) Accuracy (b) Time

(c) SUS (d) MDMT (e) TLX

Figure 7.4: Our objective measures (a) Accuracy and (b) Time, and subjective measures (c) SUS, (d) MDMT, and (e) TLX, shown for all three between-subjects conditions. Error bars represent standard error.

7.5.2 Subjective Measures

The three subjective dependent measures (SUS, MDMT, TLX) were all correlated ($ps < .001$) with each other: r = .65 for SUS and MDMT; r = -.60 for SUS and TLX; and r = -.54 for MDMT and TLX. These correlations suggest that usability and trust increase in tandem, and that workload decreases as both usability and trust increase. The correlations between the dependent variables also indicate that a multivariate analysis of the data is warranted to account for the relationships among the dependent variables.

A MANOVA was conducted using a pair of a priori orthogonal Helmert contrasts in order to test hypothesis H3 (that the feedback conditions would facilitate better user experiences than the no feedback condition) and hypothesis H4 (that the MR feedback condition would facilitate better user experiences than the physical feedback condition). An examination of the multivariate relationships of the data reveals strong support for hypothesis H3: Feedback from the robot in both the physical and MR conditions facilitated better overall user experiences than no feedback. There was also a trend in the data consistent with hypothesis H4: MR feedback facilitated better user experiences than physical feedback. The means and standard deviations of all three subjective metrics for each condition are shown in Figure 7.4.

The first Helmert contrast was significant and supports hypothesis H3, $F(3, 75) = 3.13$, $p = .03$, *multivariate* $\eta^2 = .11$. The univariate F-tests revealed that the feedback conditions were rated as (a) significantly better on usability, $F(1, 77) = 5.66$, $p = .02$, $\eta^2 = .07$; (b) significantly higher on trust, $F(1, 77) = 7.56$, $p < .01$, $\eta^2 = .09$; and (c) significantly lower on workload, $F(1, 77) = 6.50$, $p = .01$, $\eta^2 = .08$. These results offer strong support for hypothesis H3.

The second Helmert contrast was not significant, $F(3, 75) = 1.19$, $p = .32$, *multivariate* $\eta^2 = .05$. However, the means of the measures are consistent with hypothesis H4, with ratings in the MR feedback condition greater on usability and trust than in the physical feedback condition, as well as lower on workload. None of the univariate F-tests were statistically significant, but workload and trust had noteworthy effect sizes of 2-4% explained variance: $F(1, 77) = 0.59$, $p = .45$, $\eta^2 = .01$ for usability; $F(1, 77) = 2.86$, $p = .10$, $\eta^2 = .04$ for trust; and $F(1, 77) = 1.81$, $p = .18$, $\eta^2 = .02$ for workload. Given the interesting trend but insufficient statistical confidence, future work will aim to elucidate whether there is in fact a qualitative difference between the two feedback conditions along subjective measures.

We gain some additional insight from the MANOVA by examining the semi-partial coefficients (discriminant function weights) for the three user experience measures. The semi-partial coefficients are like weights in a multiple regression and indicate which of the three measures most strongly discriminates between the conditions. When contrasting feedback to no feedback, MDMT (trust) makes the strongest contribution (.57), TLX (workload) also makes a notable contribution (-.46), but SUS (usability) makes little unique contribution (.15) above and beyond TLX and MDMT. Taken together, while the three measures show high correlations and share some ability to discriminate between the feedback and no feedback conditions, the MDMT is able to stand by itself as a parsimonious tool to capture user attitudes towards a robot. This is perhaps because it is a user-friendly measure, derived from natural language people use in the domain of trust [146].

7.6 Discussion

The results from the objective and subjective measures in our user study paint a single, coherent story about the conditions we tested. In general, the feedback conditions (physical, MR) outperformed the no feedback condition, and the MR feedback condition (control) outperformed the physical feedback condition. The user experience of each condition roughly paralleled the performance of the system. Ultimately, we conclude that models that integrate feedback perform better and are preferred by users, and that MR is a promising modality for this communication.

In terms of objective measures, the feedback conditions (physical, MR) were more accurate than the no feedback condition (control), as was the MR condition compared to the physical condition. While the MR condition averaged less time than the physical condition, the time difference between the feedback condition and the no feedback condition was not statistically significant; this appears to stem from the reduced speed of the physical condition, which required extra time for the robot to move its end effector to offer feedback. The results thus fully support hypothesis H2 (MR feedback

condition compared to physical feedback condition on objective measures), with nuanced support for hypothesis H1 (feedback condition compared to no feedback condition on objective measures). Remarkably, the MR condition was simultaneously the most accurate and the fastest, contrary to the typical speed-accuracy tradeoff. These results show particular promise for the MR feedback model, which appears to exhibit the best performance in terms of both accuracy and speed.

The subjective measures on user experience offer a similar story. Participants gave better user experience ratings across all three subjective measures (usability via SUS, trust via MDMT, workload via TLX) in the feedback conditions (physical, MR) as compared to the no feedback condition (control). Although there was no statistically significant difference between the user experience ratings in the MR feedback condition and the ratings in the physical feedback condition, the means across all three subjective measures improve from no feedback to physical feedback, and again from physical feedback to MR feedback. As a result, we believe that the benefits of MR are worth exploring further in future work. The results thus fully support hypothesis H3 (better user experience in feedback conditions compared to no feedback condition on subjective measures), with trending support for hypothesis H4 (better user experience in MR feedback condition compared to physical feedback condition on subjective measures).

Finally, in running the study we anecdotally found eye gaze to be less interpretable than we expected. Specifically, although we knew that the eye gaze angle would be difficult to infer without any error, our initial pilot study indicated that looking right vs. left was easily distinguishable. However, during our user study, participants reported that they had issues discerning which direction the robot was looking.

7.7 Conclusion

This work presents a robot interaction model that is able to interpret multimodal human communication and use a mixed reality interface to perform question-asking in an item disambiguation task. We approach our problem from a decision-theoretic standpoint, and ultimately offer our new model called the Physio-Virtual Deixis (PVD) POMDP. Lastly, we report the results of our user study, which compared two feedback conditions (physical, MR) to a no feedback condition, as well as compared the physical and MR feedback conditions to each other. We found statistically significant support along both objective and subjective measures in favor of conditions that offer feedback (physical, MR) over no feedback (control), as well as statistically significant support from objective measures (and trending support from subjective measures, though not significant) in favor of a MR feedback condition over a physical feedback condition.

Chapter 8

Conclusions

This thesis, through a description of my research, has described my efforts to improve human-robot collaboration by combining decision theoretic interaction managers with multimodal mixed reality interfaces.

8.1 Summary of Results

In Chapter 3, we found that by combining multimodal observations with a Bayesian filtering state estimator, our model could infer the correct object an item referencing task with 90% accuracy. Our approach incorporated learned contextual dependencies, and ran in real time. This chapter demonstrates steps toward continuous language understanding and more effective human-robot interaction.

In Chapter 4, I incorporated robot action into the model, and showed how social feedback improves human robot communication, and how POMDPs are effective methods of generating this feedback. Our method achieved greater accuracy and a faster interaction time compared to state-of-the-art baselines: it was 2.17 seconds faster (25% faster) and 2.1% more accurate than the next best model. The FETCH-POMDP's ability to intelligently balance between clarifying uncertainty with speed allows for realistic interactions between a social robot and a human. This ability allows for realistic interactions with human users, which affords natural collaborations over tasks between humans and robots.

In Chapter 5, I depart the world of proximate human-robot interaction, and investigate how consumer-grade virtual reality hardware can enable untrained users to accurately and intuitively control a remote robot over the internet. Our model of teleoperation, which we call virtual gantry, significantly improved novice user teleoperation ability, with an improvement of 66% compared in a cup stacking task compared to keyboard and mouse interface. In the process, I created ROS Reality, the first open source bridge between ROS enabled robot and XR hardware, which formed the technical and system basis for the remainder of my work.

In Chapter 6, I return to proximate human-robot interaction, and focus on the problem of safety during HRI. We found that our mixed reality interface for communicating robot motion intent as 16%

more accurate and 62% faster compared to a keyboard and monitor based system. I describe how mixed reality enabled robots to intuitively communication their motion intent to users, preventing accidental collisions.

In Chapter 7, I combined the knowledge gained from my previous experience with decision theoretic interaction managers with mixed reality HRI to create the Physio-Virtual Deixis POMDP, a model and system that is able to interpret multimodal human communication and use a mixed reality interface to perform question-asking in an item disambiguation task. We found mixed reality feedback was 10% more accurate than the physical condition with a speedup of 160%, and improved all subjective metrics.

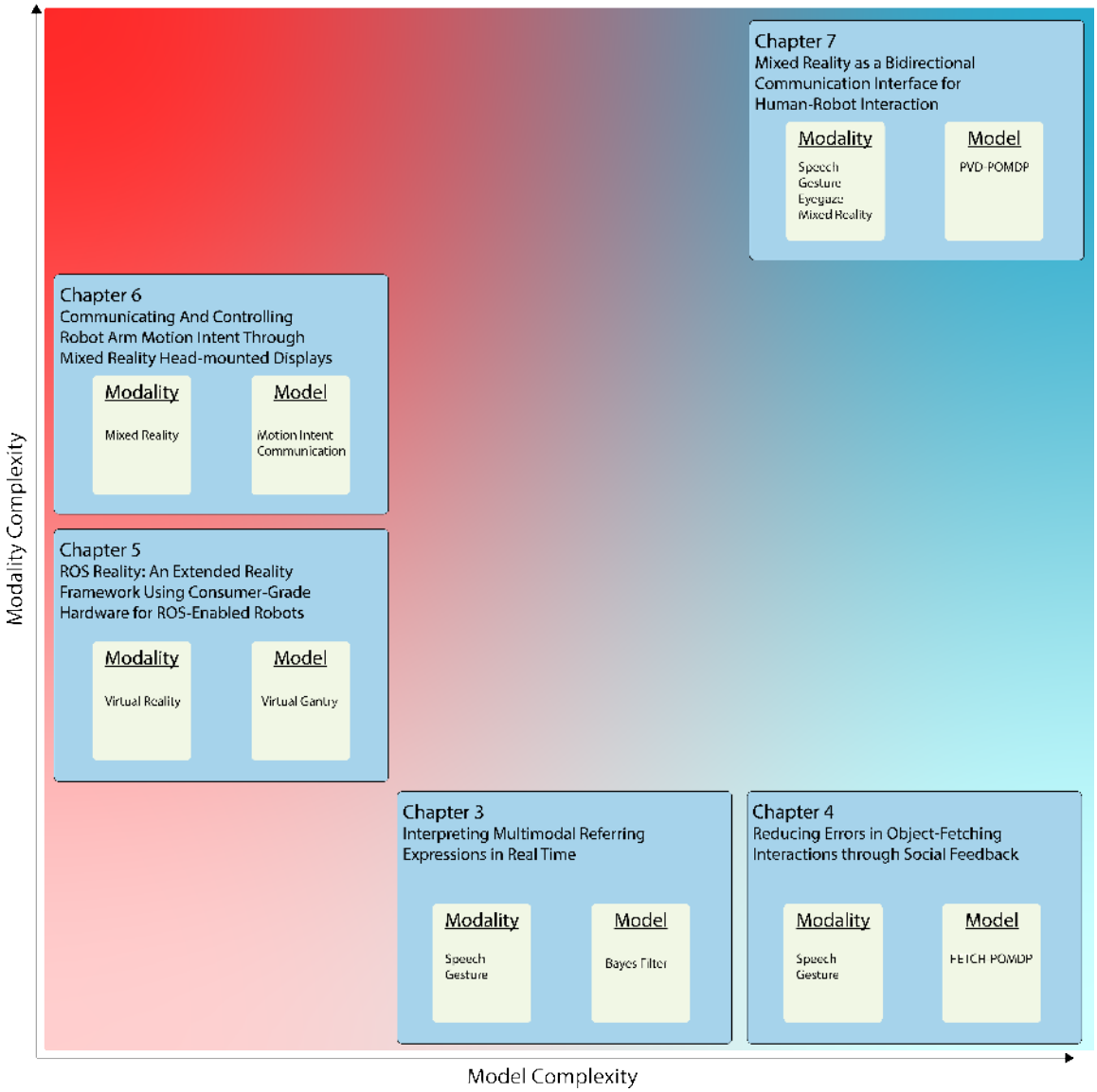


Figure 8.1: The results of this work in the problem space of human-robot collaboration.

8.2 Future Work

The field of mixed reality based human-robot interaction is still young. Throughout this work, I have focused primarily on the problems of object grounding and motion intent communication. There are yet more models and modalities that will need to be considered for human-robot interaction to rival its human-human counterpart. Imagine works that lie above and to the right of my works in Figure 8.1. I envision systems which communicate long-term tasks to robots, including novel items and locations. As mixed reality technology improves, it can be more effectively integrated into future systems, communicating not just motion or state, but also properties and affordances of the environment and its objects.

Bibliography

[1] Deepika Phutela. The Importance of Non-Verbal Communication. *IUP Journal of Soft Skills*, 9(4):43–49, 12 2015.

[2] Sachin Chitta, Ioan Sucan, and Steve Cousins. MoveIt! *IEEE Robotics & Automation Magazine*, 19(1):18–19, 2012.

[3] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2008.

[4] R. Bellman. A Markovian Decision Process. *Indiana University Mathematics Journal*, 6: 679–684, 1957.

[5] L.P. Kaelbling, M.L. Littman, and A.R. Cassandra. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence*, 101(1–2):99–134, 1998.

[6] Guy Shani, Joelle Pineau, and Robert Kaplow. A Survey of Point-Based POMDP Solvers. *Autonomous Agents and Multi-Agent Systems*, 27(1):1–51, 2013.

[7] Paul Milgram and Fumio Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information and Systems*, 77(12):1321–1329, 1994.

[8] Microsoft Corporation. What is Mixed Reality?, 2018. URL `https://docs.microsoft.com/en-us/windows/mixed-reality/mixed-reality`.

[9] Inc. Magic Leap. What is Spatial Computing?, 2019. URL `https://creator.magicleap.com/learn/guides/design-spatial-computing`.

[10] Maximilian Speicher, Brian D Hall, and Michael Nebeling. What is Mixed Reality? In *Proceedings of the ACM CHI Conference on Human Factors in Computing Systems*. ACM, 2019.

[11] David Whitney, Miles Eldon, John Oberlin, and Stefanie Tellex. Interpreting Multimodal Referring Expressions in Real Time. In *IEEE International Conference on Robotics and Automation*, pages 3331–3338. IEEE, 2016.

[12] Jean MacMillan, Elliot E Entin, and Daniel Serfaty. Communication Overhead: The Hidden Cost of Team Cognition. *Team Cognition: Process and Performance at the Inter and Intra-individual Level*, 2004.

[13] Herbert H Clark and Meredyth A Krych. Speaking While Monitoring Addressees for Understanding. *Journal of Memory and Language*, 50(1):62–81, 2004.

[14] Matthew MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions. In *Proceedings of the 21st National Conference on Artificial Intelligence*, volume 2, pages 1475–1482. AAAI Press, 2006.

[15] Juraj Dzifcak, Matthias Scheutz, Chitta Baral, and Paul Schermerhorn. What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution. In *IEEE International Conference on Robotics and Automation*, pages 3768–3773, 2009.

[16] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *5th ACM/IEEE International Conference on Human-Robot Interaction*, pages 259–266. IEEE, 2010.

[17] Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, and Dieter Fox. Learning to parse natural language commands to a robot control system. *Experimental Robotics: The 13th International Symposium on Experimental Robotics*, pages 403–415, 2013.

[18] Casey Kennington and David Schlangen. Simple learning and compositional application of perceptually grounded word meanings for incremental reference resolution. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 292–301. Association for Computational Linguistics, 2015.

[19] Kotaro Funakoshi, Mikio Nakano, Takenobu Tokunaga, and Ryu Iida. A unified probabilistic approach to referring expressions. In *Proceedings of the SIGDIAL 2012 Conference, The 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 237–246, 2012.

[20] P. Matikainen, P. Pillai, L. Mummert, R. Sukthankar, and M. Hebert. Prop-free pointing detection in dynamic cluttered environments. In *2011 IEEE International Conference on Automatic Face Gesture Recognition and Workshops*, pages 374–381, March 2011.

[21] Stefan Waldherr, Roseli Romero, and Sebastian Thrun. A gesture based interface for human-robot interaction. *Autonomous Robots*, 9(2):151–173, 2000.

[22] Matthew Marge, Aaron Powers, Jonathan Brookshire, Trevor Jay, Odest C Jenkins, and Christopher Geyer. Comparing heads-up, hands-free operation of ground robots to teleoperation. *Robotics: Science and Systems VII*, 2011.

[23] Sy Bor Wang, A. Quattoni, L. Morency, D. Demirdjian, and T. Darrell. Hidden conditional random fields for gesture recognition. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1521–1527, 2006.

[24] L. Morency, A. Quattoni, and T. Darrell. Latent-dynamic discriminative models for continuous gesture recognition. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[25] Boris Schauerte, Jan Richarz, Gernot Fink, et al. Saliency-based identification and recognition of pointed-at objects. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4638–4643. IEEE, 2010.

[26] Mary Ellen Foster, Andre Gaschler, Manuel Giuliani, Amy Isard, Maria Pateraki, and Ronald Petrick. Two people walk into a bar: Dynamic multi-party social interaction with a robot agent. In *Proceedings of the 14th ACM International Conference on Multimodal Interaction*, pages 3–10. ACM, 2012.

[27] Dan Bohus, Chit W. Saw, and Eric Horvitz. Directions robot: In-the-wild experiences and lessons learned. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, pages 637–644, 2014.

[28] Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. Learning from unscripted deictic gesture and language for human-robot interactions. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 2556–2563, 2014.

[29] OpenNI Tracker. http://wiki.ros.org/openni_tracker, 2014.

[30] Fei Song and W Bruce Croft. A general language model for information retrieval. In *Proceedings of the Eighth International Conference on Information and Knowledge Management*, pages 316–321. ACM, 1999.

[31] U-V Marti and Horst Bunke. Using a statistical language model to improve the performance of an hmm-based cursive handwriting recognition system. *International Journal of Pattern Recognition and Artificial Intelligence*, 15(01):65–90, 2001.

[32] Christopher D Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*, volume 999. MIT Press, 1999.

[33] Mark Bittman. *How to Cook Everything.* John Wiley and Sons, Inc., 2008.

[34] David Whitney, Eric Rosen, James MacGlashan, Lawson LS Wong, and Stefanie Tellex. Reducing Errors in Object-Fetching Interactions Through Social Feedback. In *IEEE International Conference on Robotics and Automation*, pages 1006–1013. IEEE, 2017.

[35] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011.

[36] P.J. Gmytrasiewicz and P. Doshi. A Framework for Sequential Planning in Multi-Agent Settings. *Journal on Artificial Intelligence Research*, 24:49–79, 2005.

[37] J.D. Williams and S. Young. Partially Observable Markov Decision Processes for Spoken Dialog Systems. *Computer Speech & Language*, 21(2):393–422, 2007.

[38] Terrence Fong, Charles Thorpe, and Charles Baur. Robot, Asker of Questions. *Robotics and Autonomous Systems*, 42(3):235–243, 2003.

[39] Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas N. Roy. Grounding Verbs of Motion in Natural Language Commands to Robots. In *Proceedings of the International Symposium on Experimental Robotics*, December 2010.

[40] Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashis Gopal Banerjee, Seth Teller, and Nicholas Roy. Approaching the Symbol Grounding Problem with Probabilistic Graphical Models. *AI Magazine*, 32(4):64–76, 2011.

[41] Maya Cakmak and Andrea L Thomaz. Designing robot learners that ask good questions. In *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pages 17–24. ACM, 2012.

[42] Adam Vogel, Christopher Potts, and Dan Jurafsky. Implicatures and nested beliefs in approximate Decentralized-POMDPs. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

[43] J. Hoey, P. Poupart, A. von Bertoldi, T. Craig, C. Boutilier, and A. Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519, 2010.

[44] B. Thomson and S. Young. Bayesian update of dialogue state: A POMDP framework for spoken dialogue systems. *Computer Speech & Language*, 24(4):562–588, 2010.

[45] J.D. Williams and S. Young. Scaling POMDPs for spoken dialog management. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(7):2116–2129, 2007.

[46] Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, pages 33–40. ACM, 2014.

[47] E. Wu, Y. Han, D. Whitney, J. Oberlin, J. MacGlashan, and S. Tellex. Robotic Social Feedback for Object Specification. In *AAAI Fall Symposium on AI for Human-Robot Interaction*, 2015.

[48] F. Doshi and N. Roy. Spoken language interaction with model uncertainty: an adaptive human–robot interaction system. *Connection Science*, 20(4):299–318, 2008.

[49] S.C.W. Ong, S.W. Png, D. Hsu, and W.S. Lee. Planning under uncertainty for robotic tasks with mixed observability. *International Journal of Robotics Research*, 29(8):1053–1068, 2010.

[50] S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. The hidden information state model: A practical framework for POMDP-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174, 2010.

[51] M.L. Littman. *Algorithms for sequential decision making*. PhD thesis, Brown University, 1996.

[52] Blai Bonet. Deterministic pomdps revisited. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 59–66, 2009.

[53] CMU Sphinx. http://cmusphinx.sourceforge.net/, 2016.

[54] M. Kearns, Y. Mansour, and A.Y. Ng. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. *Machine Learning*, 49(2–3):193–208, 2002.

[55] Sidney Siegal. *Nonparametric statistics for the behavioral sciences*. McGraw-hill, 1956.

[56] Samir Yitzhak Gadre, Eric Rosen, Gary Chien, Elizabeth Phillips, Stefanie Tellex, and George Konidaris. End-User Robot Programming Using Mixed Reality. In *IEEE International Conference on Robotics and Automation*, 2019.

[57] Baichuan Huang, Deniz Bayazit, Daniel Ullman, Nakul Gopalan, and Stefanie Tellex. Flight, Camera, Action! Using Natural Language and Mixed Reality to Control a Drone. In *IEEE International Conference on Robotics and Automation*, 2019.

[58] David Whitney, Eric Rosen, Elizabeth Phillips, George Konidaris, and Stefanie Tellex. Comparing Robot Grasping Teleoperation across Desktop and Virtual Reality with ROS Reality. In *International Symposium on Robotics Research*, 2017.

[59] David Whitney, Eric Rosen, Daniel Ullman, Elizabeth Phillips, and Stefanie Tellex. ROS Reality: A Virtual Reality Framework Using Consumer-Grade Hardware for ROS-Enabled Robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–9. IEEE, 2018.

[60] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source Robot Operating System. In *IEEE International Conference on Robotics and Automation Workshop on Open Source Software*, volume 3, page 5. IEEE, 2009.

[61] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. Communicating Robot Arm Motion Intent Through Mixed Reality Head-mounted Displays. In *International Symposium on Robotics Research*, 2017.

[62] Eric Rosen, David Whitney, Elizabeth Phillips, Daniel Ullman, and Stefanie Tellex. Testing Robot Teleoperation using a Virtual Reality Interface with ROS Reality. *Human-Robot Interacton, 2018 Workshop on Virtual, Augmented and Mixed Reality*, 2018.

[63] Christopher M Dellin, Kyle Strabala, G Clark Haynes, David Stager, and Siddhartha S Srinivasa. Guided manipulation planning at the DARPA robotics challenge trials. In *Experimental Robotics*, pages 149–163. Springer, 2016.

[64] Carlos Beltrán-González, Antonios Gasteratos, Angelos Amanatiadis, Dimitrios Chrysostomou, Roberto Guzman, András Tóth, Loránd Szollosi, András Juhász, and Péter Galambos. Methods and techniques for intelligent navigation and manipulation for bomb disposal and rescue operations. In *IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 1–6. IEEE, 2007.

[65] Ken Goldberg, Michael Mascha, Steve Gentner, Nick Rothenberg, Carl Sutter, and Jeff Wiegley. Desktop teleoperation via the world wide web. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 654–659. IEEE, 1995.

[66] Jean Vertut. *Teleoperation and Robotics: Applications and Technology*, volume 3. Springer Science & Business Media, 2013.

[67] Gunter Niemeyer and Jean-Jacques E Slotine. Toward bilateral internet teleoperation. *Beyond Webcams: An Introduction to Online Robots*, page 193, 2002.

[68] John C Byrn, Stefanie Schluender, Celia M Divino, John Conrad, Brooke Gurland, Edward Shlasko, and Amir Szold. Three-dimensional imaging improves surgical performance for both novice and experienced operators using the da Vinci Robot System. *The American Journal of Surgery*, 193(4):519–522, 2007.

[69] Martin Mallwitz, Niels Will, Johannes Teiwes, and Elsa Andrea Kirchner. The CAPIO active upper body exoskeleton and its application for teleoperation. In *Proceedings of the 13th Symposium on Advanced Space Technologies in Robotics and Automation*, 2015.

[70] Tianhao Zhang, Zoe McCarthy, Owen Jowl, Dennis Lee, Xi Chen, Ken Goldberg, and Pieter Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. In *IEEE International Conference on Robotics and Automation*, pages 1–8. IEEE, 2018.

[71] Jeffrey I Lipton, Aidan J Fay, and Daniela Rus. Baxter's Homunculus: Virtual Reality Spaces for Teleoperation in Manufacturing. *IEEE Robotics and Automation Letters*, 3(1):179–186, 2018.

[72] Alexander Kasper, Zhixing Xue, and Rüdiger Dillmann. The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics. *The International Journal of Robotics Research*, 31(8):927–934, 2012.

[73] Corey Goldfeder, Matei Ciocarlie, Hao Dang, and Peter K Allen. The Columbia Grasp Database. In *IEEE International Conference on Robotics and Automation*, pages 1710–1716. IEEE, 2009.

[74] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M. Dollar. The YCB object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics*, pages 510–517. IEEE, 2015.

[75] Christopher Crick, Graylin Jay, Sarah Osentoski, Benjamin Pitzer, and Odest Chadwicke Jenkins. Rosbridge: ROS for non-ROS users. In *Robotics Research*, pages 493–504. Springer, 2017.

[76] Thiemo Wiedemeyer. IAI Kinect2. `https://github.com/code-iai/iai_kinect2`, 2014 – 2015. Accessed June 12, 2015.

[77] Russell Toris, Julius Kammerl, David V Lu, Jihoon Lee, Odest Chadwicke Jenkins, Sarah Osentoski, Mitchell Wills, and Sonia Chernova. Robot Web Tools: Efficient Messaging for Cloud Robotics. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4530–4537. IEEE, 2015.

[78] Nathan Koenig and Andrew Howard. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 2149–2154. IEEE, 2004.

[79] NASA Human Performance Research Group and others. Task Load Index (NASA-TLX) v1.0 Computerised Version. *NASA Ames Research Centre*, 1987.

[80] John Brooke et al. SUS-A quick and dirty usability scale. *Usability Evaluation in Industry*, 189(194):4–7, 1996.

[81] Aaron Bangor, Philip T Kortum, and James T Miller. An empirical evaluation of the system usability scale. *International Journal of Human–Computer Interaction*, 24(6):574–594, 2008.

[82] William F Moroney, David W Biers, F Thomas Eggemeier, and Jennifer A Mitchell. A comparison of two scoring procedures with the NASA task load index in a simulated flight task. In *Proceedings of the IEEE 1992 National Aerospace and Electronics Conference*, pages 734–740. IEEE, 1992.

[83] Eric Rosen, David Whitney, Elizabeth Phillips, Gary Chien, James Tompkin, George Konidaris, and Stefanie Tellex. Communicating And Controlling Robot Arm Motion Intent Through Mixed Reality Head-mounted Displays. *International Journal of Robotics Research*, 2019.

[84] Terrence Fong, Illah Nourbakhsh, and Kerstin Dautenhahn. A Survey of Socially Interactive Robots. *Robotics and Autonomous Systems*, 42(3):143–166, 2003.

[85] Yuxin Han. The Social Behavior Guide for Confused Autonomous Machines. Master's thesis, Rhode Island School of Design, 2016.

[86] Brian Scassellati and Bradley Hayes. Human-Robot Collaboration. *AI Matters*, 1(2):22–23, 2014.

[87] Emanuele Ruffaldi, Filippo Brizzi, Franco Tecchia, and Sandro Bacinelli. Third Point of View Augmented Reality for Robot Intentions Visualization. In *International Conference on Augmented Reality, Virtual Reality and Computer Graphics*, pages 471–478. Springer, 2016.

[88] Toru Nakata, Tomomasa Sato, Taketoshi Mori, and Hiroshi Mizoguchi. Expression of Emotion and Intention by Robot Body Movement. In *Proceedings of the 5th International Conference on Autonomous Systems*, 1998.

[89] Bilge Mutlu, Fumitaka Yamaoka, Takayuki Kanda, Hiroshi Ishiguro, and Norihiro Hagita. Nonverbal Leakage in Robots: Communication of Intentions Through Seemingly Unintentional Behavior. In *ACM/IEEE International Conference on Human Robot interaction*, pages 69–76. ACM, 2009.

[90] Elizabeth Cha, Yunkyung Kim, Terrence Fong, Maja J Mataric, et al. A Survey of Nonverbal Signaling Methods for Non-Humanoid Robots. *Foundations and Trends in Robotics*, 6(4): 211–323, 2018.

[91] Leila Takayama, Doug Dooley, and Wendy Ju. Expressing Thought: Improving Robot Readability with Animation Principles. In *International Conference on Human-Robot Interaction*, pages 69–76. ACM, 2011.

[92] Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and Predictability of Robot Motion. In *8th ACM/IEEE International Conference on Human-Robot Interaction*, pages 301–308. IEEE, 2013.

[93] Daniel Szafir, Bilge Mutlu, and Terrence Fong. Communication of Intent in Assistive Free Flyers. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 358–365. ACM, 2014.

[94] Stefanos Nikolaidis, Minae Kwon, Jodi Forlizzi, and Siddhartha Srinivasa. Planning with verbal communication for human-robot collaboration. *ACM Transactions on Human-Robot Interaction*, 7(3):22, 2018.

[95] Daniel Szafir, Bilge Mutlu, and Terry Fong. Communicating Directionality in Flying Robots. In *ACM/IEEE International Conference on Human-Robot Interaction*, pages 19–26. ACM, 2015.

[96] Ravi Teja Chadalavada, Achim Lilienthal, Henrik Andreasson, and Robert Krug. Empirical Evaluation of Human Trust in an Expressive Mobile Robot. In *RSS Workshop on Social Trust in Autonomous Robots*, 2016.

[97] Kristin E Schaefer, Edward R Straub, Jessie YC Chen, Joe Putney, and AW Evans. Communicating Intent to Develop Shared Situation Awareness and Engender Trust in Human-Agent Teams. *Cognitive Systems Research*, 2017.

[98] Moondeep C Shrestha, Ayano Kobayashi, Tomoya Onishi, Hayato Yanagawa, Yuta Yokoyama, Erika Uno, Alexander Schmitz, Mitsuhiro Kamezaki, and Shigeki Sugano. Exploring the Use of Light and Display Indicators for Communicating Directional Intent. In *Advanced Intelligent Mechatronics*, pages 1651–1656. IEEE, 2016.

[99] Moondeep Chandra Shrestha, Ayano Kobayashi, Tomoya Onishi, Erika Uno, Hayato Yanagawa, Yuta Yokoyama, Mitsuhiro Kamezaki, Alexander Schmitz, and Shigeki Sugano. Intent Communication in Navigation Through the Use of Light and Screen Indicators. In *ACM/IEEE International Conference on Human Robot Interaction*, pages 523–524. IEEE Press, 2016.

[100] Adam Eric Leeper, Kaijen Hsiao, Matei Ciocarlie, Leila Takayama, and David Gossow. Strategies for human-in-the-loop robotic grasping. In *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, pages 1–8. ACM, 2012.

[101] Hyeong Ryeol Kam, Sung-Ho Lee, Taejung Park, and Chang-Hun Kim. RViz: A Toolkit for Real Domain Data Visualization. *Telecommunication Systems*, 60(2):337–345, 2015.

[102] Paul Milgram, Shumin Zhai, David Drascic, and Julius Grodski. Applications of Augmented Reality for Human-Robot Communication. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1467–1472. IEEE, 1993.

[103] Jennifer L Burke, Robin R Murphy, Michael D Coovert, and Dawn L Riddle. Moonlight in Miami: Field Study of Human-Robot Interaction in the Context of an Urban Search and Rescue Disaster Response Training Exercise. *Human-Computer Interaction*, 19(1-2):85–116, 2004.

[104] JL Burke and RR Murphy. Situation Awareness and Task Performance in Robot-Assisted Technical Search: Bujold goes to Bridgeport. Technical report, tech. report CRASAR-TR2004-23, Department of Computer Science and Engineering, University of South Florida, 2004.

[105] Ravi Teja Chadalavada, Henrik Andreasson, Robert Krug, and Achim J Lilienthal. That's On My Mind! Robot to Human Intention Communication Through On-Board Projection on Shared Floor Space. In *European Conference on Mobile Robots*, pages 1–6. IEEE, 2015.

[106] Jong-gil Ahn and Gerard J Kim. Remote Collaboration Using a Tele-Presence Mobile Projector Robot Tele-Operated by a Smartphone. In *IEEE/SICE International Symposium on System Integration*, pages 236–241. IEEE, 2016.

[107] Rasmus S Andersen, Ole Madsen, Thomas B Moeslund, and Heni Ben Amor. Projecting Robot Intentions into Human Environments. In *Robot and Human Interactive Communication*, pages 294–301. IEEE, 2016.

[108] Jun Rekimoto. Transvision: A Hand-Held Augmented Reality System for Collaborative Design. In *Virtual Systems and Multimedia*, volume 96, pages 18–20, 1996.

[109] Hirokazu Kato and Mark Billinghurst. Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. In *IEEE and ACM International Workshop on Augmented Reality*, pages 85–94. IEEE, 1999.

[110] Toshikazu Ohshima, Kiyohide Satoh, Hiroyuki Yamamoto, and Hideyuki Tamura. AR^2 Hockey: a Case Study of Collaborative Augmented Reality. In *Proceeding of the IEEE Virtual Reality Annual International Symposium*, pages 268–275. IEEE, 1998.

[111] Henry Chen, Austin S Lee, Mark Swift, and John C Tang. 3D Collaboration Method over HoloLens and Skype End Points. In *Proceedings of the 3rd International Workshop on Immersive Media Experiences*, pages 27–30. ACM, 2015.

[112] Michael Walker, Hooman Hedayati, Jennifer Lee, and Daniel Szafir. Communicating robot motion intent with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 316–324. ACM, 2018.

[113] Mel Slater and Maria V Sanchez-Vives. Enhancing Our Lives with Immersive Virtual Reality. *Frontiers in Robotics and AI*, 3:74, 2016.

[114] Randy Pausch, M Anne Shackelford, and Dennis Proffitt. A User Study Comparing Head-Mounted and Stationary Displays. In *Proceedings of the IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, pages 41–45. IEEE, 1993.

[115] Colin Ware and Glenn Franck. Viewing a Graph in a Virtual Reality Display is Three Times as Good as a 2D Diagram. In *Proceeding of the IEEE Symposium on Visual Languages*, pages 182–183. IEEE, 1994.

[116] Mel Slater, Vasilis Linakis, Martin Usoh, and Rob Kooper. Immersion, Presence, and Performance in Virtual Environments: An Experiment with Tri-Dimensional Chess. In *ACM Virtual Reality Software and Technology*, volume 163, page 72. ACM Press New York, NY, 1996.

[117] Roy A Ruddle, Stephen J Payne, and Dylan M Jones. Navigating Large-scale Virtual Environments: What Differences Occur Between Helmet-Mounted and Desk-Top Displays? *Presence: Teleoperators & Virtual Environments*, 8(2):157–168, 1999.

[118] David J Kasik, James J Troy, Stephen R Amorosi, Marie O Murray, and Shankar N Swamy. Evaluating Graphics Displays for Complex 3D Models. *IEEE Computer Graphics and Applications*, 22(3):56–64, 2002.

[119] Cagatay Demiralp, Cullen D Jackson, David B Karelitz, Song Zhang, and David H Laidlaw. Cave and Fishtank Virtual-Reality Displays: A Qualitative and Quantitative Comparison. *IEEE Transactions on Visualization and Computer Graphics*, 12(3):323–330, 2006.

[120] Beatriz Sousa Santos, Paulo Dias, Angela Pimentel, Jan-Willem Baggerman, Carlos Ferreira, Samuel Silva, and Joaquim Madeira. Head-mounted Display Versus Desktop for 3D Navigation in Virtual Reality: a User Study. *Multimedia Tools and Applications*, 41(1):161, 2009.

[121] Unity Technologies. Unity® Software, 2018. URL `https://unity.com`.

[122] Microsoft. Mixed Reality Toolkit, 2017. URL `https://github.com/Microsoft/MixedRealityToolkit-Unity`.

[123] Neil A Macmillan. Signal Detection Theory. *Stevens' Handbook of Experimental Psychology*, 2002.

[124] Wilson P Tanner Jr and John A Swets. A Decision-Making Theory of Visual Detection. *Psychological Review*, 61(6):401, 1954.

[125] Harold Stanislaw and Natasha Todorov. Calculation of Signal Detection Theory Measures. *Behavior Research Methods, Instruments, & Computers*, 31(1):137–149, 1999.

[126] Toney Allman. *The Nexi Robot*. Norwood House Press, 2009.

[127] Cliff Fitzgerald. Developing Baxter. In *IEEE International Conference on Technologies for Practical Robot Applications*, pages 1–6. IEEE, 2013.

[128] Zahar Prasov and Joyce Y Chai. What's in a gaze?: The role of eye-gaze in reference resolution in multimodal conversational interfaces. In *Proceedings of the 13th International Conference on Intelligent User Interfaces*, pages 20–29. ACM, 2008.

[129] Stephanie Gross, Brigitte Krenn, and Matthias Scheutz. The reliability of non-verbal cues for situated reference resolution and their interplay with language: Implications for human robot interaction. In *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pages 189–196. ACM, 2017.

[130] Pernilla Qvarfordt. Gaze-informed multimodal interaction. In *The Handbook of Multimodal-Multisensor Interfaces*, pages 365–402, 2017.

[131] Dilek Hakkani-Tür, Malcolm Slaney, Asli Celikyilmaz, and Larry Heck. Eye gaze for spoken language understanding in multi-modal conversational interactions. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 263–266. ACM, 2014.

[132] Shaolin Qu and Joyce Y Chai. An exploration of eye gaze in spoken language processing for multimodal conversational interfaces. In *Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 284–291, 2007.

[133] Henny Admoni, Thomas Weng, and Brian Scassellati. Modeling communicative behaviors for object references in human-robot interaction. In *IEEE International Conference on Robotics and Automation*, pages 3352–3359. IEEE, 2016.

[134] Adrian Bangerter. Using pointing and describing to achieve joint focus of attention in dialogue. *Psychological Science*, 15(6):415–419, 2004.

[135] Gregor Mehlmann, Markus Häring, Kathrin Janowski, Tobias Baur, Patrick Gebhard, and Elisabeth André. Exploring a model of gaze for grounding in multimodal HRI. In *Proceedings of the 16th International Conference on Multimodal Interaction*, pages 247–254. ACM, 2014.

[136] Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction*, pages 33–40. ACM, 2014.

[137] Mohit Shridhar and David Hsu. Interactive visual grounding of referring expressions for human-robot interaction. In *Proceedings of Robotics: Science and Systems*, 2018.

[138] Elena Sibirtseva, Dimosthenis Kontogiorgos, Olov Nykvist, Hakan Karaoguz, Iolanda Leite, Joakim Gustafson, and Danica Kragic. A comparison of visualisation methods for disambiguating verbal requests in human-robot interaction. *27th IEEE International Symposium on Robot and Human Interactive Communication*, pages 43–50, 2018.

[139] Tom Williams, Nhan Tran, Josh Rands, and Neil T Dantam. Augmented, mixed, and virtual reality enabling of robot deixis. In *International Conference on Virtual, Augmented and Mixed Reality*, pages 257–275. Springer, 2018.

[140] Tom Williams, Matthew Bussing, Sebastian Cabroll, Elizabeth Boyle, and Nhan Tran. Mixed reality deictic gesture for multi-modal robot communication. In *ACM/IEEE International Conference on Human-Robot Interaction*, 03 2019.

[141] N.D. Goodman and A. Stuhlmüller. Knowledge and implicature: Modeling language understanding as social cognition. *Topics in Cognitive Science*, 5, 2013.

[142] Google Cloud Speech-to-Text. `https://cloud.google.com/speech-to-text/`, 2018.

[143] Kinect2 Tracker Package. `https://github.com/mcgi5sr2/kinect2_tracker`, 2018.

[144] Magic Leap. Magic Leap Bootcamp in a Box. `https://creator.magicleap.com/learn/guides/bootcamp-in-a-box`, 2018.

[145] Matthijs TJ Spaan and Nikos Vlassis. Perseus: Randomized point-based value iteration for pomdps. *Journal of Artificial Intelligence Research*, 24:195–220, 2005.

[146] Daniel Ullman and Bertram F Malle. What Does it Mean to Trust a Robot?: Steps Toward a Multidimensional Measure of Trust. In *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, pages 263–264. ACM, 2018.