Identifying Perceptually Indistinguishable Objects

John F. Santore

Department of Computer Science and Engineering

Center for Cognitive Science

University at Buffalo

Committee:

Stuart C. Shapiro (Major Professor)

William J. Rapaport

Erwin Segal

Carl Alphonce

Acknowledgments.

A Ph.D is never completed in a vacuum. I am indebted to many people for help over the years that I worked on this project. I will take a moment to thank some of the most involved here.

I would like to first thank God without whom nothing would be possible.

I also thank my advisor and my committee and outside reader. Stuart Shapiro was my major professor, who endured my productive and unproductive times with equal aplomb and professional poise while gently steering me towards what I needed to accomplish. William Rapaport was invaluable in finding useful references in the literature, even when there seemed to me to be nothing written on a topic. Erwin Segal helped with the analysis of the human subjects data and produced most of the statistics using spss. Carl Alphonce lent a keen eye in finding technical and grammatical errors in the dissertation. Benjimin Kuipers, my outside reader gave me several nice comments that rounded out rough edges in the dissertation.

I thank my parents who raised me to be the man I am today and were the first two to volunteer to be subjects in my experiment. My father helped me to plan out my timetable in applying for graduate school while working full time. He later gave up summers working with me so that I could work on my Ph.d and get to finish my degree. Throughout the long process, he never gave up even when I was ready to, and helped me to make ends meet, especially when going to conferences many times. My mother was quieter in her support, but was always supportive and was determined to soldier on while taking her turn as a participant in my experiment, despite suffering a bit of motion sickness while participating.

I would also like to thank my wife Imelda, who though she came later on in my degree, helped to transcribe 11 of the transcripts and copied all of the coded transcripts into tech report mentioned here.

The technical support staff here at UB are the most competent and professional that I have ever encountered. It takes working in another environment to really appreciate the quick turn around and complete service that they provide.

I lastly acknowledge all of my fellow graduate students who have helped me through the years particularly the following: Frances Johnson and Haythem Ismail, my officemates during much of my work on the PIO problem who help me with many of the quirks of SNePS that I stumbled over. Pallavi Tambay helped keep me sharp by engaging in some engaging intellectual arguments. Many other students have helped over the years to make graduate school life easier: my fellow research group members in SNeRG, the cognitive science student, and my masters degree classmates all deserve mention here.

Contents

List of Figures

List of Tables

Abstract

This dissertation reports on an investigation into how Perceptually Indistinguishable Objects (PIOs) can be identified. An experiment with 68 human participants was performed to investigate how and how well people identify PIOs. The experiment was designed as a protocol analysis experiment. Participants performed a video-game like task of counting or following, both of which entail identifying objects. The analysis of this experiment shows that the human participants had a marked preference for certain situations that they believed helped them identify the PIOs more readily. Participants would try, as much as possible, to keep themselves in these situations.

A cognitively plausible computational model of identifying PIOs is developed from the results of the human subjects experiment. The cues and strategies that participants in the experiment went out of their way to use are examined and treated separately. Some participant-preferred strategies always lead to the correct answer/identification when the participant's background beliefs are correct. These strategies are generally perceptually based and are called base cases. The other set of strategies that the participants tried to use are not quite as perceptually based and are called intermediate cases. These strategies, when correctly used, lead to the right answer a great deal of the time, but are slightly more prone to failure than the base cases. The knowledge needed for the general case of identifying PIOs is also discussed and an algorithm for the model is included.

Finally, a new simulated cognitive robot is described that includes an implementation of the computational model of identifying PIOs. The robot was tested in the same environment that the human participants used for the experiments and on the same tasks. The mistakes that the robot made fell into a subset of the mistakes

that the human participants made in the experiment.

Chapter 1

Introduction.

People see objects that look the same as other objects all the time. In the modern world of interchangeable parts and manufactured objects, we are continually confronted by objects that look so much alike that we cannot use our senses alone to distinguish them. Despite this inability to rely totally on our senses, we function quite well surrounded by these perceptually indistinguishable objects (PIOs). In fact we often don't even consciously notice that we are surrounded by and regularly use PIOs in our daily lives. We must therefore, have some sort of (common sense) reasoning mechanism to identify PIOs. This dissertation investigates the common sense reasoning mechanism that people use to identify PIOs and proposes a computational model of that reasoning mechanism. It should be noted that PIO is not a cognitive category that people think about, but rather a meta-concept used in this dissertation to describe a situation in which agents find themselves.

Let me first define the terms that I will use in this dissertation. Two objects are *perceptually indistinguishable* to an agent at a given time if the agent, using its sensors, cannot perceive any difference between the two objects at that time. That is, the agent has identical descriptions for the two objects after whatever sensing it has done. More precisely, let *percept(A, O, T)* is a function on agents, objects and times which outputs the agent's cognitive description or percept of the object which it encounters at time T. Let *recalled(var)* take a variable holding a percept and return the agent's stored representation of that percept from memory. Finally, let *compatible(mem-percept, var)* take the representation of a percept retrieved from an agent's memory, mem-percept, and a variable holding a percept, var; *compatible(mem-percept, var)* will return true if the agent finds no differences between the perceptual features in mem-percept and those in var. In the *compatible* function, missing perceptual features will not be counted as a difference. This allows a quickly perceived red object to be compatible with a square red object seen later. Given these function definitions we can now define perceptually indistinguishable PI as:

Let V_1= percept(A, O_1, T_1)

V_r=recalled(V_1)

V_2 = percept(A , O_2 ,T_2), where $T_2 \geq T_1$

then PI(O_1, O_2, T_2, A) iff compatible(V_r, V_2)

Note that according to this definition, objects are perceptually indistinguishable to a particular agent and moreover are perceptually indistinguishable to a particular agent at a particular time. It is possible for one agent to find two objects perceptually indistinguishable but for another to find the same two objects perceptually distinguishable. It is also possible for an agent to find two objects perceptually indistinguishable at some time but for that same agent to find the two objects perceptually distinguishable at another time. At time T_2 in the above definition, A perceives object O_2. Unless $T_1=T_2$, A is perceiving O_2, but is working with a memory of O_1, which is usually an incomplete description. This could come about because agent A didn't have enough time to get a more complete description from its sensors the first time, or perhaps didn't perceive a need to get a better description at time T_1. Agent A would of course be able to form a more complete description at a later time.

When I discuss *identifying* perceptually indistinguishable objects, I mean the following: When an agent finds an object that is perceptually indistinguishable from one encountered before, the agent *identifies* the object if it successfully decides whether the object is one the agent previously encountered, or decides correctly that it is a new object. If the object has been encountered before, and the agent has encountered more than one such object before, the agent must also know which one it is currently encountering. Thus I use the term 'identify' to cover both identifying an object as a new object and identifying an object as a previously seen object; the latter is sometimes called re-identifying (Pollock, 1974).

I need to distinguish objects which are perceptually indistinguishable from those that are *functionally indistinguishable.*[1] Two objects are functionally indistinguishable if they serve completely the same purpose and are thus interchangeable. There is often no need to distinguish an individual object from several other functionally indistinguishable objects. If you have a basket containing a pile of rags made from old T-shirts of various colors and you want to clean your car, any of the rags will do, you don't have to think of a particular rag, just use the rag till it is dirty and then use a clean rag. Likewise, if you are doing rough carpentry and have a bag full of 10-penny common nails, it doesn't matter which one you pull from the bag and hammer into the board at any given spot on the board; each of the nails is functionally indistinguishable from one another. Just as perceptual indistinguishability depends on the perceiving agent, functional indistinguishability

[1] Thanks to J.P. Koenig for the term 'functionally indistinguishable'.

depends on the agent and the situation and use of the object. Consider a cocktail party where the hostess has two dozen perceptually indistinguishable brand new glasses. When the party starts and the guests are first selecting a glass, the glasses are both perceptually and functionally indistinguishable; neither the hostess nor the guests care about which glass is which. However, after the guests have drunk out of their glasses, the glasses remain perceptually indistinguishable, but are no longer functionally indistinguishable, since most people will now care quite a lot about which glass is "theirs". In the preceding two examples, the rags were functionally indistinguishable but perceptually distinguishable, but the nails and glasses were both functionally and perceptually indistinguishable. The set of functionally indistinguishable objects intersects with the set of perceptually indistinguishable objects, but neither is a proper subset of the other. Having made the distinction between functionally and perceptually indistinguishable objects, I will not discuss functionally indistinguishable objects further in this dissertation.

People often encounter objects that they find perceptually indistinguishable from objects with which they have previously come in contact. These objects range from the aforementioned ten-penny common nails found in hardware stores and garage workshops to new medium-red Saturn SL automobiles. When people encounter PIOs, they can often use knowledge of spatial and temporal location, as well as any background knowledge they have about the objects themselves, to identify the object they are currently viewing.

Sometimes the identification of a PIO is difficult, and the person trying to identify the PIOs is easily confused. For example, suppose a person is following a robot into a room a few seconds behind the robot, and that when the person enters the room there are 4 robots that are perceptually indistinguishable from the original robot. In this case, it seems intuitive that most people would have a hard time picking out (identifying) the original robot that the person was following. On the other hand, sometimes people have no trouble identifying PIOs, and it seems like a very easy "commonsense" task. For example, consider a person who has a drawer containing a pile of perceptually indistinguishable stamps. If the person takes out a stamp, puts it on an envelope, and mails that envelope, then, the next day, if she takes out another stamp, she would be able to instantly identify this stamp as being different from the one that she put on the envelope the day before. She would not spend any time wondering or worrying that the stamp might have fallen off the envelope or returned to the drawer in some way. One main focus of this dissertation is an investigation into exactly what knowledge and strategies people use to identify PIOs. The human focus of the dissertation also includes an investigation to determine situations in which people can easily identify PIOs and those situations in which people find it more difficult to identify PIOs.

When an artificial agent (in particular, a real or simulated robotic agent) interacts with the world, it will also often encounter objects that, to its perceptual system, are perceptually indistinguishable. Most robotic agents today are equipped with sensors that are not as acute as human perceptual organs.[2] In particular, most robotic agents use laser range finders, sonar, and cameras to obtain information about the world. The resolution of the data retrieved by these sensors is lower than that of human perceptual organs. Therefore, robotic agents will encounter even more PIOs than people do when the robots interact in the "real world", i.e., outside of a special, controlled environment.

An embodied artificial agent, such as a cognitive robot, can actively use its sensors to provide information needed by its reasoning system. I'll call this the first feature of an embodied agent (as opposed to a disembodied reasoning system). Such an agent can act in the service of reasoning, and reason about the actions that it needs to perform. An embodied agent can therefore use its reasoning and mobility to gain a more complete description of an object that it is trying to identify. A second feature of such an agent is that the agent can use its reasoning to decide when it does not need to use its sensors further. A third feature of an embodied agent is that it may investigate the world to gain additional information that it needs to identify an object when its sensors are not sufficient to identify that object (i.e. the object is a PIO). In this dissertation, I assume that the identification routines are given the best description that the agent has yet come up with and that the object must be identified based on that description. Therefore, the first feature of the embodied agent is assumed but not explicitly discussed again. In cases where an agent is trying to identify an object that is perceptually indistinguishable from some other object, if that agent can quickly identify the object, it has no need to continue to scan the object to try to find a perceptual difference between the two objects. Chapter 4 presents a computational model that describes several cases in which the identification of PIOs is very easy. In these

[2]Expensive, special-purpose robotic agents like the Spirit and Opportunity Mars rovers are excepted from this generalization.

cases, the agent should stop scanning the object with its sensors when trying to obtain a unique perceptual description. The computational model also makes use of the embodied agent's ability to search the world to verify the identity of an object that it is trying to identify. Chapter 3 details an experiment done with human participants designed to investigate how people identify PIOs. The human participant experiment explicated the kinds of mobile activity which are useful in determining the identity of perceptually indistinguishable objects in the world. Those activities are found, both implicitly and explicitly, in the computational model.

Such an embodied artificial agent will depend heavily on its vision system. Standard computer vision techniques, particularly for object recognition, are rather limited. See Jain et al. (1995) and Sanka et al. (1999) for two surveys of vision system techniques. Most object recognition techniques use a set of stored object representations to identify the objects that the system perceives. This places a fixed limit on the types of objects that a system can recognize. Since such systems store only the basic shape and edge information about a type of object (or color-blob information in the case of the CMUcam-style vision algorithms (Rowe et al., 2002)), very often only categories of objects can be identified rather than individual objects. Object recognition (the classification of an object into a known category) is often the goal of robotic vision systems. (Sanka et al., 1999, page 297) say "Object recognition is based on assigning classes to objects and the device that does those assignments is called the classifier." Computer vision for arbitrary object recognition is itself a hard (and quite possibly an AI-complete) problem. However, there are systems that can do object recognition quite well for a limited domain of objects. A successful object recognition system could identify an object as "a red car" but not as "a particular red car." The model described in this dissertation can be used to extend an agent that is already equipped with an effective object recognition system to also be an agent that can effectively identify individual objects.

When an agent is reasoning about the identity of an object, there are many cues that the agent might need to use to identify the object. I initially hypothesized several cues that I believed would be useful for the identification of PIOs. The experiment with human participants proved that many of those cues were used by people and suggested several other cues that I had not originally considered. Some of the resulting cues are relatively easy to formalize, such as whether two objects are perceived simultaneously. One cue that has been more difficult to formalize is a sense of time for the agent. It is particularly important when identifying moving objects that the agent have a sense of time. An object that is in place P_1 at time T_1 may not have time to move to place P_2 by time $T_{(1+\varepsilon)}$. In that case, the agent ought to be able to reason that there are two distinct objects, the one perceived at P_1 and one perceived at P_2. The model of identifying PIOs described in this dissertation will correctly make this distinction. The work described in this dissertation is built on an existing model of reasoning and acting in time (Ismail and Shapiro, 2000).

Other cues may not be as difficult to model in the beliefs of the agent, the challenge comes because there is no straightforward translation from the sensor data available to beliefs. One such example is the motivation of an object that the agent is trying to identify. People can use perceived *motivations* to help identify an object. I'll use the term "motivations" in this dissertation to refer to both the intentions and mental/emotional state of an anthropomorphized object, as well as the use/purpose of a non-anthropomorphized object. For example, a robot might be anthropomorphized as having considerable mental capacity, so one who is introduced as a tour guide, will be considered as though it "wanted" to be a tour guide. An oscillating fan on the other hand has a purpose of cooling the room, and an agent can expect its movement behavior to reflect that purpose. Both can be considered under the broad heading of "motivations" in this dissertation.

Once a person has a belief about what the motivation of the object is, the person can use that cue. The model presented in this dissertation is based on the beliefs of the agent and does not consider the additional problem of reasoning from sensor data to beliefs. Others have worked on modeling the beliefs, intentions, and plans of other agents; see for example (Chalupsky and Shapiro, 1996; Singh, 1998; Kaplan and Schubert, 2000). However, the general problem of identifying another agent's intentions from sensor data is still an open problem.

There is no way to guarantee the success of an agent trying to identify PIOs. Sowa notes that

> Without continuous observation, an object's identity can only be guaranteed by indirect means... In general, the recognition of identity depends on inference, and the inference may be mistaken (Sowa, 2000, pp 120–121).

Outside of the case of continuous observation (which turns out to be a very easy case of identifying PIOs, characterized as a “base case” in Chapter 4), people sometimes make mistakes when reasoning about the identities of PIOs. The frequency of those mistakes and the types of situations in which people tend to make more mistakes will be discussed in Chapter 3. Because the computational model of identification of PIOs is a cognitively plausible one, based on the methods that humans use to identify PIOs, an agent using this model to identify PIOs will be subject to the same mistakes that humans make.

An artificial agent implementing the model described in Chapter 4 should perform at least at the level of a human agent in most cases. For example, suppose a person were asked to follow a robot. If the person were a bit slow, and the robot passed behind an obstruction, the person would watch for a robot to come out from behind the obstruction at the correct time and assume that the robot seen coming out from behind the obstruction is the same one that was to be followed. The computational model described here will make the same assumption. This makes the model vulnerable to the same sorts of tricks that would fool a person, but the gain (in not having to quickly run behind the obstruction to check each time) is worth the possibility of being fooled occasionally. This model also benefits from the usual cognitive modeling benefit of being able to predict human behavior and performance on identifying PIOs in novel situations.

As a second example of how the model will behave as a human would in identifying PIOs, suppose we first have a person who encounters a robotic agent. At this point, the person realizes that she wants to go in the other direction, so she makes a 180-degree turn. Immediately upon turning, she encounters another robotic agent, perceptually indistinguishable from the one that she just saw. A human agent in this situation would know intuitively that the just-encountered agent could not possibly be the previously encountered agent, because the original agent did not have the time to get from the first location to the second. The cognitive model described in this dissertation reaches the same conclusion.

These are some of the typical reasoning heuristics that allow an agent equipped with the model described in this dissertation to perform at a human level. The agent is allowed to make the same mistakes that a human makes but should also make correct inferences when a human would easily make them in the same situation.

The rest of the dissertation is organized as follows. In Chapter 2 I review the relevant literature on PIO-related work done in philosophy, psychology, and computer science. In Chapter 3 I report on human performance at the task of identifying PIOs. I describe the results of an experiment with 68 human participants designed to elucidate the strategies people use to identifying PIOs. The experiment was also designed to show how well humans can identify PIOs. In Chapter 4 I lay out a cognitively plausible computational model of PIO identification that is based on the results of the human participant experiment. In Chapter 5 a simulated agent is described. The agent is used as a testbed for the computational model. Finally in Chapter 6 I summarize the conclusions and significance of the work described in this dissertation.

Chapter 2

Related Work.

2.1 Introduction

The identification of perceptually indistinguishable objects (PIOs) is a topic that belongs firmly in the realm of cognitive science. A limited amount of previous work has been done on the identification of PIOs. This work spans at least philosophy, psychology and computer science/artificial intelligence, three traditional cognitive science disciplines. Many previous researchers touched on the identification of PIOs only tangentially in the pursuit of their own projects. Some of those projects are discussed in this chapter. I will discuss several projects that indicate a need to identify PIOs or indicate an expectation that identifying PIOs is easily performed by people even though the project itself might not be involved with the identification of PIOs. These projects indicate that people can and often do identify PIOs, so a robotic agent will need to identify PIOs as well if is to be fully functional in the world.

2.2 Indistinguishable Objects in Philosophy

Leibniz is credited with formulating the "Principle of the Identity of Indiscernibles" (Forrest, 2002; Leibniz et al., 1969). He postulates that objects that are indistinguishable, in that they share all of their intrinsic and extrinsic properties, are in fact, the same object. Intrinsic properties can be thought of as those properties that are part of the object itself. An object's composition, texture and shape are all examples of intrinsic properties. Extrinsic properties are relational properties involving the object in its relation to other entities. The property of being three meters from a flag pole would be an example of an extrinsic property. Many philosophers have argued for or against Leibniz's Principle since he originally published his ideas. The most notable objection comes from Black (1952) with his two-sphere universe. All of these philosophical arguments are based on an abstract, absolute notion of indistinguishability. I am not concerned with objects that are intrinsically identical; but rather with objects that have no differences in those intrinsic properties which are perceivable by our agent. For this reason, Leibniz's principle is of nothing more than academic interest to this work.

In the late nineteenth century, Frege (1892a) recognized that the problem of object re-identification requires more than simple perceptual identity. Frege states "[t]he discovery that it is not a different and novel sun which rises every morning, but that it is the very same, certainly was one of the most consequential ones in astronomy (p 85)." Even though we all assume that the sun is the same today as it was yesterday, it is not immediately apparent why we do this. We cannot really identify the sun by its perceptual properties because we are visual creatures and we cannot look directly at the sun without damaging our eyes. Therefore there must be something more going on when we identify the sun. Frege postulated two aspects for expression about objects, a sense or intension, of the properties of an object, and a reference, which is the actual object itself. Frege recognized that an object cannot be identified based solely on its perceptual properties.

Fodor (1994) takes a stance counter to that of Frege. Though he acknowledges that Frege is right in that there are objects that cannot be identified purely by their perceptual properties, he claims that these cases are so rare as to be 'evolutionarily irrelevant.' Fodor claims that PIOs, though they exist, are so rare that we can still identify objects solely on the basis of their unique perceptual properties. Fodor's claim is that while many objects may appear to be PIOs to us (such as perhaps blades of grass in the field), to an expert the objects are quite perceptually distinguishable. However, since most people are not experts at all aspects of their environment, most people will still encounter objects that are perceptually indistinguishable to *themselves.* While Fodor's claim about the lack of PIOs as far as experts are concerned may have been correct before the 19th century, with the advent of modern manufacturing techniques and mass production of identical, interchangeable units, there are clearly many PIOs around us all the time that even experts cannot identify based purely on perceptual properties. Not only are there many PIOs around us every day in our modern industrial society[1], but we are also often able to identify most of these objects (see chapter 3 for details of human performance of the identification of PIOs). Fodor's claim that we identify objects solely on the basis of unique perceptual properties seem then to be not quite correct.

The philosopher Pollock (1974) has discussed issues involved in the reidentification of objects. He holds that "...the problem of reidentification can be regarded ultimately as the problem of reidentification of perceived objects." (Pollock, 1974, page 138) As this statement implies, Pollock is interested in the general issue

[1] I will grant that certain peoples living in non-industrialized rural areas are not likely to encounter as many PIOs on a daily basis as those living in industrialized societies, but they will still encounter many objects that are perceptually indistinguishable because of the lack of time and ability to scan the objects carefully.

of reidentification of objects, both when an object looks the same as it looked before and when its appearance has changed. Pollock's focus differs from the focus of this work in an important way. Pollock's work assumes the bias that the object to be identified is a previously identified object that should be reidentified as that same object. The focus of this work is to identify an object that looks just like one that the agent has seen before. It might be new or it might be the same as a previously encountered object. There is no bias toward one or the other. Pollock's interest in reidentification tasks focuses on cases where sufficient perceptual similarity to a previously perceived item is enough to reidentify it. In this case perceptual indistinguishability of the object itself is not sufficient to reidentify the object. For example, a car that has been scratched since the agent last saw it is not perceptually indistinguishable from the previous perception of the car. Pollock is interested in this type of reidentification task.

Pollock notes that there are two cases of perception-based reidentification of objects. The first case is when the agent has the object under continuous observation. The agent operates on an object, then rests or performs some task with the object in sight at all times, and later needs to operate on the same object again. This means that the agent needs to reidentify the original object. Pollock accepts that "continuity of appearance is a logical reason for reidentification."(Pollock, 1974, page 144)

The second case of perception-based reidentification is when the agent has not had the object under continuous observation, a situation Pollock calls "discrete observation." Pollock points out that we human agents have discovered inductively from objects under continual observation that some traits of objects, like specific gravity, tend to be stable, such that if an object has the attribute it tends to keep it, while others, such as wetness, tend to be unstable. From these observations, we inductively reidentify an object if the newly perceived object matches enough of the traits of the previously identified object.

Pollock proposes that reidentification is, in essence, a perceptual task. When Pollock discusses perceptual acts, he discusses them as single acts of perceiving from a single angle. This includes whatever local spatial relations to nearby objects can be determined. However, in the case of discrete observation, it is a contingent fact that agents are more often wrong in their reidentification tasks. For this reason, agents often use non-perceptual clues to augment their support for reidentifying an object.

My work diverges from Pollock's on the issue of identification based on single acts of perceiving. My agent will be able to use its vision system as much as it wants to determine whether the object is perceptually indistinguishable from one it has seen before.

Pollock also notes that although non-perceptual attributes can't be required to directly reidentify an object, those non-perceptual attributes may be used as defeaters for a reidentification. By non-perceptual attributes, Pollock means any attribute that requires manipulation of the object to identify. He uses the example of an oriental box with personal letters as their secret contents. The letters are not available to perception without manipulating the box. However, if an agent reidentifies an oriental box as the one he/she saw before, the presence or absence of the letters can defeat the reidentification if the state of the letters in the box currently perceived is not the same as the state of the original box.

Pollock is viewing this task at a lower level than I do. Since he is using only a "single act of perceiving," he does not allow reasoning to play a positive role in the task reidentification. I am concerned with an identification task at a higher level, one that uses multiple acts of perception and the use of reasoning to come to a conclusion about the identity of an object.

Pollock does not explicitly discuss the idea of reidentifying perceptually indistinguishable objects, except in the context of the currently fantastic idea of teleportation. He discusses the following example, in which indistinguishable objects are to be reidentified, as a case that might lead to failure to reidentify an object that is under continuous observation. Suppose there are two identical boxes on the table in front of an observer, which he keeps under continuous observation. He can assume that the one belonging to him, which starts on the left, is the same one that is on the left 20 minutes later, providing that there has been no observed movement of the boxes. However, Pollock admits that, since teleportation is not a logical absurdity (and may someday be achieved), the observer cannot be perfectly certain that the left box is his. As discussed above, an examination of the contents of the box can defeat the assumption that the left box belongs to the observer.

Though he is interested in a slightly different problem, one with a bias toward identifying an object as one previously seen, Pollock's treatment of reidentification in general provides a philosophical treatment of the intuitive ideas which act as general guidelines and restrictions for some cases that a computational theory of identifying perceptually indistinguishable objects must handle.

The philosopher Strawson (1964) has also written about issues concerning the reidentification of objects. Strawson is interested more in identifying and reidentifying the referents being discussed by a speaker, something that is not the focus of this work. He does discuss some aspects that apply to the general issue of reidentification as well.

Strawson defines two types of identity that correspond with two meanings of "the same x". He calls the first "numerical identity." Numerical identity between two perceptions means that the extensions of those perceptions are actually the same object in the world. When I talk about an object being the same as one that the agent has seen before, I mean it in the sense of Strawson's numerical identity. Strawson calls the second type of identity "qualitative identity." This is very similar to the relationship that I call "perceptually indistinguishable." He uses this definition to equate objects because the objects have identical perceptual properties and function. Strawson's bias toward reidentifying objects differs from this work in the same way that Pollock's work does.

Philosophers interested in the philosophy of art have directly discussed perceptually indistinguishable objects. Good forgeries are perceptually indistinguishable from the original piece of art. Philosophers have debated whether forgeries which are perceptually indistinguishable from the original have the same aesthetic value, or any aesthetic value. Jenkins (1995) and Gough (1996) have argued that a forgery, even one perceptually indistinguishable from the original, does not have the same aesthetic qualities as the original. This contrasts with the claims of earlier philosophers in the field such as Lessing (1965), who claimed that a perceptually indistinguishable forgery had all the aesthetic qualities of the original and that it is only people's snobbery that makes then think a forgery has a lower aesthetic value than the original. These debates from the philosophy of art illustrate that there are real differences between objects that are perceptually indistinguishable. However, they do not give us any insight into how to identify perceptually indistinguishable objects themselves.

2.3 Indistinguishable Objects in Psychology.

Much of the previous psychological work that is relevant to the identification of PIOs has been carried out in the subfield of developmental psychology. Such work includes discussions of how young infants and children display the kinds of knowledge that I believe are necessary to identify PIOs.

One of the earliest researchers to look at the development of knowledge and reasoning in children was Piaget, an early 20th century psychologist who closely observed children, especially his own (Piaget, 1952, 1954, 1948). Piaget devised a theory of stages of infant cognitive development. Each stage of development was marked by the infant gaining the ability to make additional inferences about objects in its world. Piaget notes that at the earliest stage of development, infants seem to act as if an object no longer exists when it is out of sight. At the next stage, infants will retrieve an object hidden by a covering if the infant is touching it, otherwise the infant again behaves as though it no longer exists. At each stage the infant develops a better apparent understanding of objects until at "Stage 6" (about 18 to 24 months old) infants appear to have a full understanding of object permanence. Object permanence is clearly necessary for identifying PIOs. Without a concept of object permanence, an agent has no hope of identifying objects unless the agent can keep the object continuously in sight. Everyday experience tells us that this is not always possible.

More recent research, summarized by Johnson (1998), calls some of the stage divisions proposed by Piaget into question. However, the more recent literature confirms that children by the age of two years have most of the very basic background knowledge needed to identify objects. They have an understanding of object permanence (an object doesn't cease to exist when out of perceptual range unless destroyed) and object identity (an object is still the same object if it moves or is briefly out of the child's perception). These are two notions that any agent that attempts to identify objects, including PIOs, must have.

An understanding of object motion is needed to identify moving PIOs. Spelke and her colleagues (von Hofsten et al., 2000; Spelke et al., 1992) have found that children as young as 6 months can learn the basics of object motion and some physical laws, like the continuity and solidity. They have shown however that understanding of some other physical laws that govern the movement of objects, such as gravity and inertia, are not understood until later. Some of the knowledge needed to identify moving objects is available to people at a very young age. However, other cues are not available until later. The participants in the experiment described in chapter 3 were all adults so they should all have both the knowledge available to infants and the common sense knowledge that Spelke and her colleges have found is not available to infants.

The objects that a PIO-identifying agent encounters must be reasoned about differently depending on certain basic properties of their potential mobility. Mandler (1991) has noted that several basic categories of image objects appear to be available to prelinguistic infants. Mandler notes that infants seem to distinguish both animate motion and self motion. Animate motion is defined as the default (and more organic) motion that children respond to while self motion is defined as a more mechanical sort of motion. For my purposes, simply categorizing all auto-mobile objects, both biological and mechanical, in the same class should be sufficient. But certainly objects that move on their own must be treated differently than objects that do not. Mandler also notes that infants seem to have categories for agent-caused motion and mechanically-caused motion. A PIO-identifying agent would be well served by having a concept of those sorts of objects that can be moved readily by other objects or agents and those which cannot be. An implementation of this very basic human cognitive process will help in deciding if it is reasonable that an object we are seeing now might be the same one we saw before.

While many developmental psychologists have discussed the emergence of object identification strategies in infants and small children, I am not aware of any that have directly looked at the identification of PIOs in small children. Indeed, in at least one case, researchers have simply assumed that children could identify PIOs. Gopnik and Sobel (2000) assume that 3-year old children can reliably identify PIOs in their experiment. Gopnik and Sobel were interested in the naming of objects by young children. In their experiment, they had a "blicket detector", a machine that would light up and make a noise when certain objects were placed on it. The experiment involved placing an object against the detector, at which the detector would activate. The children were told that this object was a blicket. Next the remaining objects were placed next to the detector one at a time. One of the other objects caused a reaction in the detector and the other two did not. When the objects were handled, care was taken to make sure that the children could see the objects at all times as the object was removed from its place, placed on the detector, and then returned to its previous resting place.

After the last object had been returned to its original position, the children were asked which of the three remaining objects was a blicket. The experiment included several conditions, the most relevant one being the condition in which all four objects were perceptually indistinguishable from one another. In the case where the four objects were all PIOs, most of the children still identified the second blicket as the second object that caused a reaction in the detector: 74% of children chose the detected object as the second blicket, significantly above chance of 33%. Gopnik and Sobel have provided evidence that children as young as three can identify PIOs at least some of the time. It is important to remember, though, that the objects were viewed by the children throughout the experiment, did not move on their own, and were returned, after being placed on the detector, to the exact place where they were previously positioned. These conditions turn out to be some of the easiest conditions under which to identify PIOs, conditions that will be called base and intermediate cases in chapter 4.

Pylyshyn is another psychologist who has worked with PIOs while interested in something completely orthogonal to PIO identification (Pylyshyn, 1989; Pylyshyn and Storm, 1988). Pylyshyn was interested in showing that people have a mechanism for tracking multiple moving objects at the same time, in parallel rather than serially "time sharing" attention. His experimental setup required people to keep track of several perceptually indistinguishable dots that moved around a screen at the same time that several other perceptually indistinguishable distractor dots were moving around the screen. All dots, both focus and distractor, were perceptually indistinguishable from one another. The participants in Pylyshyn's experiment could successfully track a small number (up to 5 or 6) of these dots simultaneously with a high degree of success. Pylyshyn proposed a parallel tracking mechanism called a FINST[2] which would account for people's ability to track multiple targets simultaneously. However, the FINST mechanism is only useful while an object is under continuous observation. As will be seen in chapter 4, this also proves to be one of the conditions under which people can identify PIOs very easily. One could get an artificial agent to identify PIOs under the same conditions by providing a computational implementation of FINSTs.

People do not always identify objects based on their physical properties even when those physical prop-

[2] FINST originally stood for "FINger of INSTantiation" but now stands on its own as an independent term according to (Schmidt et al., 1998).

erties are sufficient to individuate those objects. Simons and Levin (1998) designed an experimental scenario in which an experimenter asked for directions from unsuspecting pedestrians on a university campus. In the middle of the conversation between the pedestrian and the experimenter, two accomplices carried a large door between the two. As the door passed, the original experimenter would walk away with the door-carrying party, while a second experimenter (previously traveling with the door-carrying party) would remain behind and continue the conversation. More than half of the participants failed to react or notice the change in conversation partners despite the fact that the two experimenters differed in clothing, appearance, and voice. A second experiment in which the two experimenters assured themselves of appearing as "out-group" to the participants provided even more convincing results. Only 33% of participants noticed the change when the experimenters deliberately dressed as out-group members. This despite the fact that again the experimenters varied in appearance, clothing, and voice. There are, therefore, cases in which people only cognize (or conceive of) some part of an object's possible description. If the person found reason to look more closely, the person could gain a more complete description of the object, but often a more complete description is not necessary. So people generally build a description that is only as complete as the person believes necessary for the task at hand. This tendency of people to not build complete descriptions can lead people to have to identify objects as PIOs even if they are not truly perceptually indistinguishable to that agent. If the person didn't build a complete enough description while he had the object in sight, the person may be forced to identify the object as a PIO when he next sees an object with that description. For an example of this see §3.5.5.

Simons and Levin's work illustrates another hazard in the task of identifying PIOs. The participants in this experiment who didn't notice the switch of experimenters simply identified the person asking directions (the two were functionally indistinguishable to the participants). So when the switch was made, the participants identified the new experimenter as the same object, using the same mental entity (using what will be called "the continuously perceived object" intermediate case in Chapter 4.) Unless the person (or computational agent) notices a problem with its identification, there is no need to ever consider it erroneous. The human participants in the experiments, described in Chapter 3 of this dissertation, showed similar behavior. A participant might follow a robot thinking that it is the only robot. When the participant sees another, the participant is sometimes shocked; such participants (in the experiment) often vocalized that they are no longer certain of identity of the robot that they were following.

2.4 Indistinguishable Objects in AI.

Maida (1993) has done some research on agents making mistakes about the existence of objects in the world. Maida developed an agent system in which a sort of superagent looked into the beliefs of an agent who was observing the world. The superagent would then notice the perceptual mistakes of the observed agent, particularly when the observed agent mistook the same object to be two different objects or mistook two objects in the world to be the same object. Maida's intention was to detect the misconceptions that the observing agent made about the identity of objects in the world. Maida developed a notion of an e-class of mental objects for his agent. An e-class is a set of all ground-term mental entities that the agent believes are equivalent by means of the agent's equality operator. Maida proposed two possible mistakes that an agent could make based on this notion of an e-class. The first he called an error of compression. This is defined as the agent using one e-class to represent more than one object in the world. Thus, the agent mistakenly believes that two real world objects are the same object. The other mistake that Maida identifies is one he calls an error of dispersion. He defines an error of dispersion as one in which the agent uses more than one e-class to represent some real world object. This sort of error may be glossed as the agent mistakenly believing that one real world object is actually two different objects.

I have gone farther than the work done by Maida by actually having the agent try to reason about whether two objects that are perceptually identical are indeed the same object, or if they are simply two objects that are too similar in appearance to the agent's sensors to be distinguished. Maida was interested in finding the possible mistakes that an agent might make. I am interested in a reasoning agent that is subject to the types of mistakes that Maida discovered. Unlike Maida's experimental setup, where one agent knew everything there was to know about the world and about the agent it was observing, I will not be able to guarantee success. My agent will be subject to both the compression and dispersion errors that Maida discussed.

Symbol Anchoring (Coradeschi and Saffioti, 2003; Coradeschi and Saffiotti, 2001a,b), defined as "the process of creating and maintaining the correspondence between symbols and sensor data that refer to the same physical objects," is a computational problem that requires a model of PIO identification in any working solution. Coradeschi and Saffioti propose three functionalities that a solution to the symbol anchoring problem requires in their framework. The three are *Find*, *Track*, and *Reaquire.* The ability to identify PIOs is implicitly necessary in both the *Track* and *Reaquire* functionalities. The *Track* functionality assumes that the object being tracked is continuously observed by the agent. This continuous observation will certainly make the task of identifying the object and anchoring its description to the appropriate mental symbol easier. The *Reaquire* functionality in particular, however, needs a complete theory of identifying PIOs. *Reaquire* is the case "in which an object is reobserved after some time" (Coradeschi and Saffioti, 2003, p 91). In order to know that the same object has been reobserved after some time (rather than a new object that looks just like the old one), the *Reaquire* functionality requires a mechanism for identifying PIOs. Coradeschi and Saffioti do not describe such a mechanism, and have not considered the problem until recently.[3]

Chachich et al. (1997) have developed a vision system for traffic management that reidentifies cars as they travel from one station to another. This system is built around a series of vision systems mounted on traffic lights. The visual recognition is done based on the shape and size of the vehicle, as well as a complex "color fingerprint." Once a vehicle is identified by one of the vision stations, its characteristics are passed on to the main system which tries to reidentify the vehicle at the next station. In addition to the purely perceptual cues that this system has to work with, it also uses several other cues to help it identify the vehicle. The system uses arrival time likelihood, and, after tracking a vehicle through several stations, it begins using some aspects of driver behavior to help identify the vehicle it is tracking. The aspects of driver behavior that the system uses include how often a car changes lanes, how closely the car follows the one in front of it, and relative speed of the car compared to the rest of the traffic stream. Using these additional non-perceptual cues, the system is able to distinguish the vehicle it was tracking from a new, perceptually indistinguishable vehicle that has entered its traffic stream. The use of these sorts of behavioral cues might be a very useful thing to

[3] Personal communication at the symbol anchoring workshop of the AAAI-04 conference.

have my system use when it tries to identify the motivations of mobile agents, which is one of the useful cues that a PIO identification agent needs to consider. The algorithm described in chapter 4 checks to see if the agent has beliefs about the motivations of the other agent it is trying to identify. However, the algorithm does not form those beliefs itself. Other strategies like those proposed by Chachich et al. can be used to form those beliefs.

This system does many of the things that my system will be able to do; however, it differs in two critical ways. First its sensors are in a fixed position. Even though this system reidentifies and distinguishes perceptually indistinguishable objects, the system is passive and completely at the mercy of the objects that it is perceiving. Our system uses its mobility in service of this identification task (at least in a limited way at the moment). The Chachich et al. system is not a system capable of reasoning. Its reliance on domain specific heuristics results in the system being very domain specific; its domain is limited to a single place and a single class of objects that it can identify: moving vehicles. My solution is a more general reasoning solution to the problem of perceptually indistinguishable object identification.

Doherty and his colleagues (Doherty et al., 2000; Doherty, 2004; Heintz and Doherty, 2004) have worked on the Wallenberg laboratory for Information Technology and Autonomous Systems (WITAS) project, an autonomous mobile agent with the ability to track an individual vehicle as it navigates urban and rural environments. The WITAS project is focused on an embodied agent in the form of an autonomous helicopter approximately: 8.8 feet long and 2.3 feet tall with a rotor length of 9.8 feet. As such, the helicopter can carry on board itself all the computers needed to control the agent. The system therefore exists as an independent mobile agent. This mobile tracking agent is a step beyond the work by Chachich et al. in that the Chachich project was based on a fixed set of distributed cameras operated by a single agent. Each camera was at a known location and so there was no question of the agent mistaking where it saw something. The WITAS agent is able, once given a vehicle signature and an initial location, to follow and track that vehicle indefinitely as it drives around an urban road system. The WITAS agent can also track and follow that car along a more rural road that includes a tunnel through a mountain. There is no indication that there is another perceptually indistinguishable car to act as a distractor in either tracking scenario. However, the ability to track an object and predict its emergence from behind an occluder is essential to any system that can identify PIOs. In order to track an object, one must be able to recognize the object as the same object at regular intervals. The WITAS agent tracks continuously perceived objects using tracker objects and activity demons to supervise the tracker objects (Sandewall, 2002). This tracker object/activity demon system can track several objects simultaneously and is therefore essentially a computational implementation of Pylyshyn's FINST proposal discussed in section 2.3. Therefore, though the WITAS project does not explicitly tackle the problem of identifying PIOs, the WITAS agent has the ability to identify PIOs in at least the cases where an object can be continuously viewed, and where the object is out of view for only a short time. These are cases where people find the identification of PIOs relatively easy, as will be seen in chapters 3 and 4. Like the Chachich et al. project described above, the WITAS project is currently limited to identifying only very limited classes of objects, vehicles, and buildings.

Perceptual aliasing (Kuipers and Beeson, 2002; Kuipers and Byun, 1991) is a potential error in identifying places which arises from arises "from sensorily indistinguishable places" (Kuipers and Byun, 1991). That is, perceptual aliasing occurs when two places cannot be distinguished using an agent's sensors. Kuipers and his colleagues use their term "sensorily indistinguishable" in the same way that I use "perceptually indistinguishable." The concept of perceptual aliasing was developed by Kuipers and his colleagues for their work in cognitive robotics. It is closely related to, and indeed is a special case of, identifying PIOs. If a robot's sensors return the same sensor image at two different times, the robot must decide if the two images refer to the same place or to a different place. This identification of perceptually indistinguishable places is very similar to the idea of identifying perceptually indistinguishable *objects*. One could think of places as being objects that cannot move and that together make up the world that an agent operates in. Kuipers and his colleagues have methods to refine sensor data to reduce perceptual aliasing, but readily admit that in many environments it cannot be completely eliminated. Their robot uses "historical context", considering the likelihood of encountering a particular place that looks like the one the robot now perceives given the recently encountered places, to overcome perceptual aliasing and identify perceptually indistinguishable places. Essentially, the robot uses (relative) location information, gleaned from knowledge of its own motion and the places it has seen recently, to identify perceptually indistinguishable places. Kuipers and his colleagues are able to rely solely on location information because places, locations in the environment, are static and do not move.[4] I hypothesized that location information will be important in identifying perceptually indistinguishable objects as well. However, often objects in the world move or can be moved, so I need to use more than just location information (though location information is useful, too) to identify perceptually indistinguishable objects rather than perceptually indistinguishable places. It is also possible that perceptually aliased places will cause added difficulties in identifying perceptually indistinguishable objects found in those places. See chapter 3 for evidence that humans exhibit this difficulty.

In this dissertation, I propose a cognitively motivated computational theory of how agents, particularly artificial embodied agents (such as robots) can use reasoning to identify PIOs the same way humans do. The use of human performance as a model and a metric when designing a robotic agent is not new. Most recently, Trafton et al. (2004) have used the performance of humans at a hide-and-seek task to design a robot that can do that same task in the same manner (and at the same level of performance). The scope of the human trials discussed in this dissertation to formulate this theory is much larger than that used by Trafton et al. (they used a single subject; I used 68.) However, the intention is similar: A theory based on human cognitive methods will allow me to develop an agent capable of doing a task, in this case identifying PIOs, in a manner similar to the way humans do. Basing my algorithm on human performance gains me the twin advantages of, first, giving me a basis for a computational solution to a very difficult commonsense problem and, second, making the artificial agent easier to work and interact with, because it does the PIO identification task in a way that humans would expect it to.

Identifying a PIO is a sub-problem of object *identification* rather than object *recognition*. Object recognition is an important part of computer vision and is defined in a computational vision textbook (Jain et al., 1995) as the process of finding and “labeling objects [in the real world] based on known object models.” That is, object recognition in computer vision is the process of deciding what *category* an object belongs to. By object identification I mean deciding which individual object is is being observed, rather than deciding

[4] Elevators and other unusual features are ignored in this dissertation since all of the environments described are single floor environments.

what category of objects it belongs to. When an agent perceives an object, it first needs to use its (probably computer-vision based) object recognition system to decide what category of thing it is, then it uses its object identification routine to choose and associate a mental concept to the object. An object identification system uses non-perceptual properties and background knowledge to identify the object as being the same one that the agent perceived at some previous time or to identify it as something new that the agent has never thought about or encountered before.

Other embodied robotic agents, for example (Santos, 2003), require a model of identifying PIOs in order to be fully functional agents in the world. Santos's project involves an embodied robotic agent that interacts with a large room-sized world. Because computer vision is such a difficult problem, he abstracted the world objects into large columns with various textures. Santos's hardware robot operated in a large room filled with pillars approximately 6 feet tall and covered in various textures such that no two identical textures were next to each other. This allowed the agent's vision system to process each of the columns that it perceived as an individual object and sidestep many of the hard issues related to computer vision. However, the easier computer vision comes at the price of turning potentially perceptually distinguishable objects into PIOs. His agent needs a model of identifying PIOs in order to completely understand the world that it operates in.

Chapter 3

Human Identification of Perceptually Indistinguishable Objects.

3.1 Introduction.

In this chapter I describe an experiment designed to investigate how people identify perceptually indistinguishable objects (PIOs). The purpose of the experiment is two-fold. First, the experiment provides real data about what strategies people use to identify PIOs in a variety of common situations. Second, the experiment yields data on how good people are at identifying PIOs in those same situations. The experiment described in this chapter meets both of these needs.

When designing the experiment, I initially hypothesized that several informational cues would be important for humans to identify PIOs. Many of these cues were expected to interact with each other. I hypothesized that an object's location would be important in identifying PIOs. When an agent perceives something that is perceptually indistinguishable from something else it has perceived, and the two perceptions occur close to one another in space, then the two perceptions are more likely to be of the same object. I also hypothesized that an object's mobility would be important. If an object cannot move on its own, one can treat location with more importance than if the object can move on its own. Moreover, how mobile, that is how fast, the object is was expected to play a role. In relation to mobility, time was expected to play an important role in identification. The more time since the last time one saw an object of a particular type, the less sure one can be about the object's identity when seeing one again. And finally, I hypothesized that how common a type of object appears, would be important. I hypothesized that objects that have a unique appearance, or at least those that very rarely share appearances, like people, can be more readily identified than objects that have a more common appearance. I made one additional hypothesis about the world itself. I hypothesized that objects in the world would be harder to identify when the rooms in the world that the objects were in were themselves perceptually indistinguishable so that people would likely fall victim to perceptual aliasing as described by Kuipers and Beeson (2002).

I designed a human study consisting of several PIO identification tasks to determine what reasoning strategies people use to identify PIOs. The study was conducted using the protocol analysis method (Newell and Simon, 1972; Ericsson and Simon, 1984) to help elicit the strategies that people use, and which tasks people find difficult and which are easier.

The tasks in the study are grouped into two types: counting tasks and following tasks. The experiments were conducted using a virtual three dimensional (3D) "world" rendered on a computer screen similar to those used in many first-person video games. The counting tasks were done in a world with four interconnecting rooms, and the following tasks were done in a world built to look like a wing of an academic building. In the counting tasks, the participants were asked to count the total number of some type of object in the virtual suite of rooms rendered on a computer screen. In the following tasks, the participant was to follow a tour guide despite the presence of distractors in the virtual suite of rooms.

Both counting and following tasks require the identification of objects. Certainly, in order to follow a single object (a robot, a person, a car or any other object) one has to identify the object when one begins to follow it. And, as one follows the object, one must again identify the object, if not continuously, then at least at regular intervals whenever there is a choice point (a time in the performance of the task when the participant must make a decision, in this case about the identity of the object the participant is following). If there is ever the possibility that one of two objects is the followed object, a participant may follow both for as long as they both go in the same direction; however, when the followed objects move in different directions and create a choice point in the task the follower must certainly identify, at this point, the one that she was originally following. In a counting task as well, objects must be identified in order to be counted. Each object that is counted must be individuated and identified as different from all of the other objects, or the counter will undercount. Likewise, each object that is seen again must be identified as being the same object that was seen before or the counter will over count. Because counting and following tasks require object identification and allow a reliable method of detecting errors in identifying objects, these two types of tasks were chosen for the experiment discussed in this chapter.

In the remainder of the chapter, I will briefly discuss the protocol-analysis method, describe the methodology common to all of the tasks in the study, discuss the individual tasks, and finally I will then examine the qualitative data obtained in these protocol-based experiments.

3.2 Using the Protocol-Analysis Method of Analyzing Data.

The experiment discussed in this dissertation was designed and analyzed using the *protocol analysis* technique described by Newell and Simon (1972), and particularly the refined approach to the technique described by Ericsson and Simon (1984). In particular, I use the assumptions that “The subject’s verbalizations correspond to some part of the information in the subject’s short term memory, usually information recently acquired.”[1] and “The information in a subject’s short term memory (and, as per [the previous assumption], that information reported verbally) consists mostly of information required... and active goals and subgoals and strategies driving the subject’s activity.” (Ericsson and Simon, 1984, pages 263-264)

In a protocol analysis experiment, participants are asked to speak aloud their activities and motivations during performance of the task. These “verbal protocols” are recorded and later analyzed using the above assumptions that everything the participants verbalized related to the information in their short term memory, and the information in short term memory is related to the “goals and subgoals and strategies driving the subject’s activity.”

Operating under these assumptions I assume that most of a participant’s verbalizations are directly related to their current thinking in solving the problem in the experiment.

I analyzed the protocols collected in these experiments from transcripts of the participant’s vocalization, which were recorded on audio tapes during the experiments. The original audio tapes were transcribed as faithfully as possible to include as many of the false starts, pauses, and vocal inflections as possible. These transcripts were then coded according to the above assumptions.

Transcripts were broken into protocol segments, each of which represented a single statement by the participant. The statements were made as small as possible, often clauses rather than sentences, so that only a single action, plan, or sensory report (referred to here as operators) is expressed in each statement. I did aggregate segments that continually describe the application of a single operator by the participant. This follows the (Ericsson and Simon, 1984) methodology in which they suggest that “during the segmentation, units should be defined that are large enough that all information for making an encoding decision is contained

[1] Ericsson and Simon’s use of the word ‘subject’ is synonymous with the use of the word ‘participant’ in this dissertation.

in a single segment" (page 290).

A small set of common operators was developed that would code, with arguments, nearly all of the segments in the protocols. Thirty operators (the operators represent actions and are often in functional form) were sufficient to code the vast majority of segments. Another seven operators were used to code rarely occurring statements that were clearly relevant to the task. These other operators however accounted for only one half of one percent of the total codes. The operators were used compositionally (the results of one operator were used as the input to another) in order to code the full breadth of participant utterances. See §A.1 in Appendix A for the operator codes used and their intended meanings. In addition to the operators, object codes were used to code the objects that were the parameters of some of the operators. §A.2 of that appendix shows the operators used. Notice that while most of the object codes are constant terms the "object name" code is simply a stand-in for any common object name.

In the coding of the transcripts, a '**' means that the code represents what the participant said, but that code and hence participant's utterance represents an erroneous belief on the part of the participant.

Most of the participants' speech is directly related to the task in some way. However, some is not, such as the following whimsical statement produced at one point by participant 45: "Am I a robot? Think I'm a robot. O-oh! I've been living a lie. I'm a robot!" Such speech is marked as a comment in the coding of the transcripts.

The tasks that the participants performed differed in at least one important aspect from those described by Newell and Simon (1972) and most of those referred to by Ericsson and Simon (1984). Newell and Simon (1972)'s tasks included cyptarithmetic, logic, and chess problems. All of these problems provide a very structured framework for solving the problem. All of the tasks have a series of discrete states. There are very rigid rules for how to get from one state to another. In a cryptarithmetic task, for example, Newell and Simon's participants were limited to operations involving one or more of the letters in the problem. Since the problems had only at most 10 unique characters in them, the participant is limited to making operations on those 10 characters. My experimental tasks, in contrast, were all continuous. Participants could move in small increments and be as diligent or as lax as as they chose in their performance of the task. The counting tasks, in particular, are much less structured and less discrete than any of Newell and Simon's tasks; the participants were allowed to use considerably different actions at any time or situation. They could travel to any of the rooms, attempt to interact with the other objects, and often explored the capabilities of their virtual avatar (the virtual avatar is the participant's "game self") as they were working on the task. In the following task, participants who had lost the tour guide robot, often wandered throughout the suite looking for it, often entering rooms that were skipped by the original robot that the participant was told to follow. Both sets of tasks lacked discrete states and well-specified operators to move from one state to another. However, even though the behaviors of some participants were apparently strikingly different from others when viewed on the screen, I still found that the vast majority of participant protocol sections could be coded using the small set of operators discussed above.

3.3 Experimental Methodology.

3.3.1 Participants.

There were 68 participants in the experiment. All participants were adult volunteers. They were recruited through posters on the University at Buffalo campus, postings made to newsgroups local to the University at Buffalo, and short overview talks given to graduate students in the Department of Computer Science and Engineering at the University at Buffalo. Participants in the experiment included a broad range of graduate and undergraduate students as well as people not affiliated with the University. Participants all spoke excellent English, though English was not the first language of some of the participants.

3.3.2 Instruments.

The virtual suite of rooms, the world that participants interacted with, was rendered and presented on a custom-built personal computer with a 19" monitor. The suite was presented using the Crystal Space 3D graphics engine (Tyberghein et al., 2002) in full-screen mode. The scenes are rendered at a rate of 20–35 frames per second, providing a feeling of realistic motion. See figure 3.1 for a screen capture showing

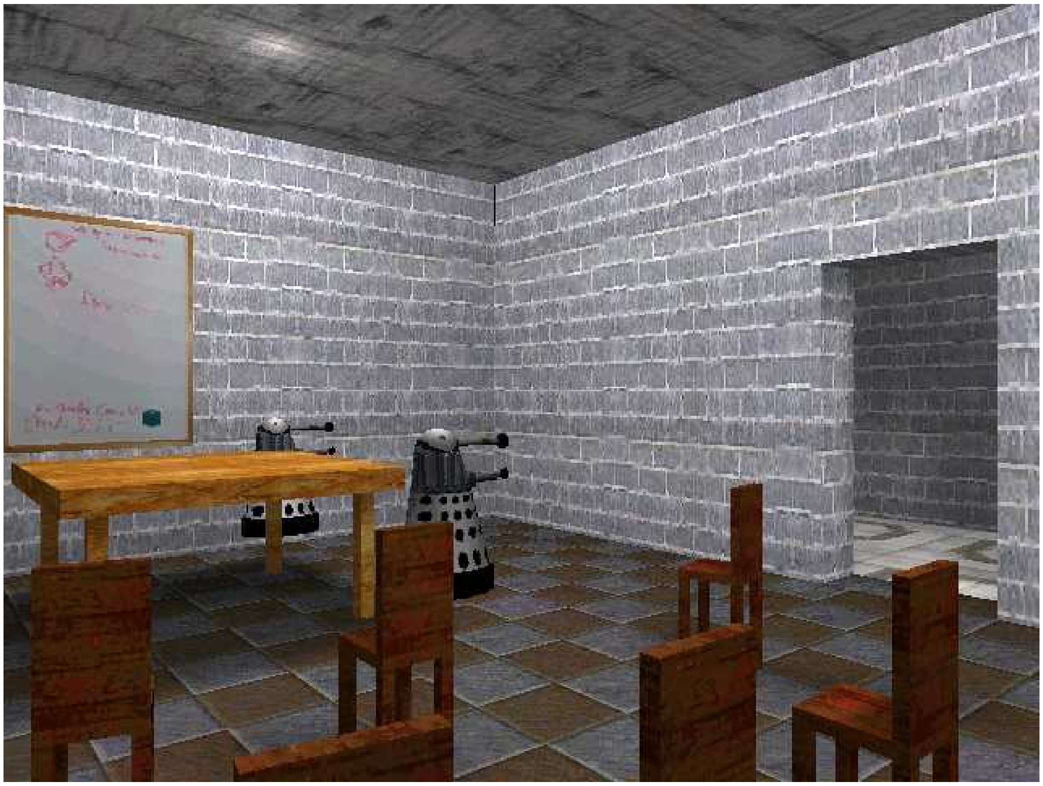

Figure 3.1: A screen capture of a sample scene that a participant might see in the experimental setup.

a scene from one of the experiments. The protocols were recorded using a Labtec Axis-301 headphone-mounted microphone plugged into a Panasonic RQ-L30 tape recorder.

3.3.3 Experimental procedure.

All of the tasks in the experiment involved participants navigating through a suite of virtual rooms. The participants were given a description of the task and their instructions on a single sheet of paper. (The text of the instructions can be found below with the relevant task.) The experimenter set up the task on the computer while the participant read the instructions. The experiments were conducted in a closed room containing the participant and the experimenter. Participants were asked to describe aloud what they were doing and why they wanted to do it while they were performing the experiment.

Participants were each given one counting task and one following task which required the participant to follow a focus object. The counting tasks and the following tasks used different suites of rooms, so that each participant began each task without knowledge of the layout of the suite. The tasks were assigned in random order so that some participants did the following task first and and some participants did the counting task first.

While the participant was engaged in the task, the experimenter sat beside and slightly behind the participant, not saying anything, unless the participant fell silent for more than approximately 3–5 seconds, whereupon the experimenter would prompt the participant to "remember to explain what you are doing and why you are doing it."

When the task was complete, the experimenter asked participants for a retrospective. The first question in this retrospective was always "What strategies did you use to do this task?" Depending on the answer to the first question, other questions would often follow. Follow-up questions are particularly likely if there appeared to be an inconsistency between the participant's actions during the task and their answer to the strategy question.

3.3.4 The practice task.

The participants were first given a practice task to familiarize themselves with the virtual environment and the keyboard navigation mechanism. The instructions for the practice task were:

> This task is intended to give you practice using the experimental setup.
>
> In this task you will explore a small suite of rooms, allowing you to become used to controlling the simulation. In one of the rooms there are several perceptually *distinguishable* objects. Your task is to inform the experimenter how many objects there are in that room. (The light fixtures on the wall need not be counted.) You may ask the experimenter what the objects are if you do not recognize something.
>
> For this practice task you will not be recorded, however, please report what you are doing and why until you have reported the number of objects in the room.

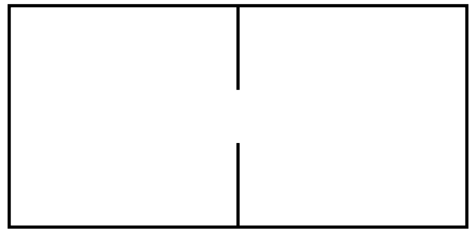

Figure 3.2: The floorplan of the two room suite used in the practice task.

> After reporting your answer, if you are not yet comfortable with using the keyboard navigation, you may continue to practice for a few minutes.

The suite of rooms was a simple, two-room suite laid out as in figure 3.2. One of the rooms was empty, and the other had five objects in it: a table, a glass, a bottle, a chair, and a robot. None of the objects moved around in the room; the robot "wiggled" in place but did not move around the room. Participants would encounter some of these objects in the real tasks. (Participants would encounter only some of the objects again in the recorded trials; which objects a particular participant would encounter depended on task assignment.) Participants started out in the empty room facing the blank wall opposite the door to the room with the objects. They had to go through the doorway between the two rooms to see the objects. Maneuvering through doorways was something that non-game-players found quite difficult in pre-experiment testing of the scenarios with members of the SNePS Research Group (A research group in the University at Buffalo's Computer Science and Engineering Department).

3.3.5 Task1: Counting Immobile Glasses.

Participants selected for task 1 searched a square suite of four interconnected virtual rooms (see figures 3.3 and 3.4) in order to count the number of drinking glasses in the suite. All of the glasses were perceptually indistinguishable from one another. See Figure 3.5 for a sample scene from this experiment.

Figure 3.3: A floorplan of the four room suite in which all of the rooms are perceptually distinguishable.

Participants.

There were 33 participants assigned to this task, split as evenly as possible between two task variations. There were 16 in the first variant and 17 in the second.

Procedure.

Participants were randomly assigned one of two variations of the floorplan. In one, each room had different textures on the walls and ceilings, making every room perceptually distinguishable. Figure 3.3 shows this layout. In the second case, the diametrically opposing rooms had the same textures on the walls, ceilings, and floors, making them perceptually indistinguishable, increasing the likelihood that the participant would perceptually alias (Kuipers and Byun, 1991; Kuipers and Beeson, 2002) two or more of the rooms. This layout is shown diagrammatically in figure 3.4. In the simulation, the glasses were immobile in a room on some piece of furniture, either a table or a chair. A few of the participants did explicitly check to see if they could move the glasses and found that they could not.

Participants in this task were given the following instructions:

> When the program starts you will find yourself in a suite of rooms.
>
> In this task you must count the total number of (drinking) glasses in the suite of rooms —

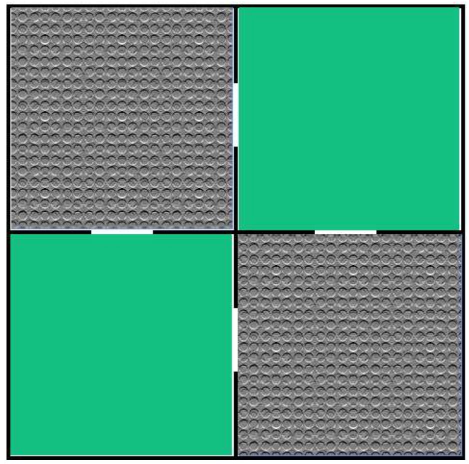

Figure 3.4: A floorplan of the four room suite in which the opposing rooms are perceptually indistinguishable.

not in each room, but in all the rooms combined. The glasses are square-ish shaped glasses. When you believe that you know how many glasses there are, stop and report that number to the experimenter. When you report your answer, this task will end.

Remember that as you perform the task, you are to explain what you are doing and why you are doing it.

The experimenter timed the participants from the time they saw the virtual suite of rooms till the time they stopped and reported the number of glasses using a Sportline digital stopwatch.

3.3.6 Task 2: Counting Mobile Robots.

Participants in task 2 counted the number of randomly moving robots in a square suite of interconnected virtual rooms. The suite was laid out with different textures for the walls and ceilings of each room as in figure 3.3.

Participants.

Thirty-six participants were assigned to this task, split evenly into two groups of 18 for two variations of the task.

Figure 3.5: A bottle and several perceptually indistinguishable glasses on a table in the immobile glass counting task.

Figure 3.6: The silver-gray version of the robot.

Procedure.

There were two versions of this task. In the first version, all of the robots were copies of the silver-gray robot shown in figure 3.6. In the second version there were two groups of robots, one group was copies of the silver-gray robot and the second group was made up of copies of the orange and gold robot seen in figure 3.7. The robots moved randomly, turning 180 degrees if they bumped into anything and occasionally, at random intervals, turned 90 degrees to the left or right. This random movement led the robots to wander within the rooms and often between the rooms as well.

Participants in this task were given the following instructions:

> When the program starts you will find yourself in a suite of rooms.
>
> In this task you must count the total number of robots in the suite of rooms —not in each of the rooms but the total for all of the rooms. As you will see, the robots can move. When you believe you know how many robots there are, stop and report that to the experimenter. If, after trying to accomplish this task for sometime, you believe that you cannot get an accurate count, stop and report this. At this point this task will end. If you were unsure of how many robots were in the suite of rooms, the experimenter may still ask for your best "guess".
>
> Remember that as you perform the task, you are to explain what you are doing and why you

Figure 3.7: The orange and gold version of the robot.

are doing it.

The experimenter timed the participants from the time they saw the virtual suite of rooms till the time they stopped and reported the number of robots using a Sportline digital stopwatch.

3.3.7 Task 3: Following a robot with randomly moving distractors.

Participants in task 3 followed a silver-gray robotic tour guide through a suite of rooms (see figure 3.1 and 3.8). There were several randomly moving distractor robots in the suite. The distractors were all perceptually indistinguishable from the tour guide that the participants were to follow.

Participants.

Twenty-seven adult volunteers participated in this experiment.

Procedure.

The suite was laid out with the floorplan shown in figure 3.8. The line in the diagram represents the approximate path that the tour guide took. There were five moving distractor robots that moved randomly as in

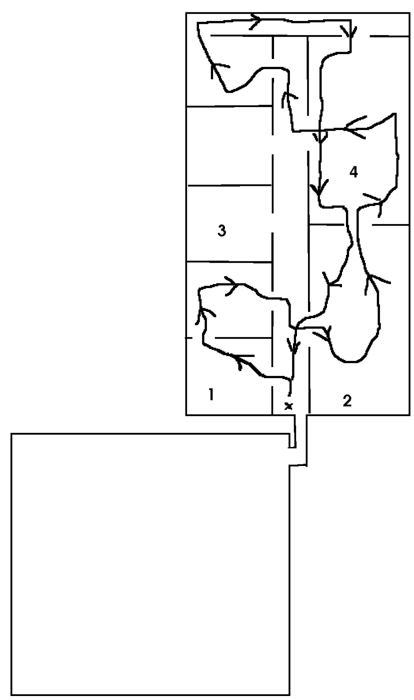

Figure 3.8: The floor plan for the larger suite of rooms used in Experiment 3. The line represents the approximate path of the robot tour guide, starting and ending at the point marked “x”. Though irrelevant to the task, the large room was in the virtual space. It was accidentally visited by two participants.

Experiment 2. They would turn 180 degrees if they hit something, and would turn 90 degrees to the left or right at occasional random intervals. When the experiment began, there was one distractor robot in the room marked 1 in figure 3.8, three distractor robots in the room labeled 2, and one distractor in the room labeled 3. As in the previous experiment, the distractors moved around the room and, if their random path took them through a doorway, entered another room.

Participants were given the following instructions for this task:

> When the program starts you will find yourself in a suite of rooms.
>
> For this task, a robot will appear in front of you when the experimenter presses the key to begin the task. You are to follow *that* robot through the suite of rooms until the experimenter tells you to stop.
>
> Remember that as you perform the task, you are to explain what you are doing and why you are doing it.

Participants in the experiment began at the 'x' marked on the floorplan in figure 3.8. The robot that they were to follow appeared just in front of them and immediately began its movement through the suite of rooms. Participants were able to move their first person view at about twice the speed of the robot they were following. They could also move more slowly if they wished.

3.3.8 Task 4: Following a robot with distractors that have purposeful movement.

We developed task 4 when we realized that some participants were using the randomness of the distractors' movement in experiment 3 to help them identify their focus robot. In this task both the tour guide and the distractor robots follow predetermined paths. Each path is unique.

Participants.

There were twenty-seven adult participants in this task. These participants did not participate in any other following task.

Procedure.

The participants in the task were once again asked to follow a tour guide. They were given the following instructions:

> When the program starts you will find yourself in a suite of rooms.
>
> For this task, a robot tour guide will appear in front of you when the experimenter presses the key to begin the task. You are to follow *that* robot through the suite of rooms until the experimenter tells you to stop.
>
> Remember that as you perform the task, you are to explain what you are doing and why you are doing it.

As in task 3, participants started on the ‘x’ marked on the floorplan shown in figure 3.8.

The robot tour guide which the participants were to follow behaved in exactly the same way as before, following the same path and moving at the same speed. The distractors however each had a predefined path of their own. There were 4 distractors, each with a path that would intersect or approach the path of the tour guide (focus) robot one or more times at about the time the tour guide was at that point in its path. Thus each of the distractor robots was visible to the participant (if the participant was still following the focus robot) at least once for several seconds at a time.

3.3.9 Task 5: Following a person with distractors that have purposeful movement.

Task 5 was a control task. Many participants in task 3 tried very hard not to notice anything but the focus robot. These participants actively reported focusing on their robots, trying not even to notice other robots or other scenery in the room. I therefore designed a following task with perceptually distinguishable objects. In task 5 the tour guide is a virtual person, and there are other (virtual, simulated) people as the distractors. See figure 3.9 for a view of this experiment.

Participants.

There were twelve adult participants in this task.

Figure 3.9: An example scene from task 5 showing two perceptually distinguishable people, a distractor in the foreground and the focus person in the background to the right.

Procedure.

This task was set up exactly the same as task 4, described above, except that the robots were replaced with people. These people were perceptually *distinguishable* from one another. The tour guide and distractor people followed the same set of paths that the tour guide and distractor robots followed in experiment 4.

The participants were given the following instructions:

> When the program starts you will find yourself in a suite of rooms.
>
> For this task, a tour guide person will appear in front of you when the experimenter presses the key to begin the task. You are to follow *that* person through the suite of rooms until the experimenter tells you to stop.
>
> Remember that as you perform the task, you are to explain what you are doing and why you are doing it.

3.4 Analysis of Quantitative results.

3.4.1 Quantitative Measures.

The transcripts from the experiment were used to generate the quantitative data discussed in this section. A fifteen-field record was established for each experimental participant. Each record contained the following fields:

1. A numeric participant identification number;
2. Counting-task identification number (the counting-task this participant performed);
3. Following-task identification number (the following-task this participant performed);
4. Task order (whether the participant performed a counting-task first or a following-task);
5. Participant's self-reported computer game experience (3 discrete levels);
6. Counting-task success, yes or no;

7. Following-task success, yes or no;
8. The number of words in the counting-task protocol;
9. The number of words in the following-task protocol;
10. Number of times the participant referred to a room in the virtual world during the counting-task;
11. Number of times that the participant referred to a room the in virtual world during the following-task;
12. Time in seconds taken to complete the counting-task;
13. The ratio of the number of room-mentions to words in the counting-task, (i.e., Field 10/Field 8);
14. The ratio the number of room-mentions to words in the following-task (i.e. Field 11/ Field 9) ;
15. Ratio of the number of room-mentions to time taken to complete the counting task (i.e. Field 10/Field 12).

The following-task took a fixed time for successful participants so no times were kept for the following task. Thus field 15 is only concerned with the counting task time. There were a few empty cells, due to recording errors or operational malfunctions.

3.4.2 Statistical Analyses .[2]

Differences among the treatment groups.

The different groups were compared using the measures enumerated above by using analyses of variance. Most of the participants performed both a counting and a following task; the counting and following tasks were (first) analyzed separately. The measures most relevant for the counting tasks (including time to complete the task, the number of words in the task transcript, number of room-mentions, and ratio of room-mentions to words in the transcript) were surprisingly similar across tasks and subtasks. There were *no*

[2]**I thank and acknowledge Erwin Segal of my committee for using his copy of SPSS to generate the numerical results used in this section.**

significant differences between any of the measures; in all cases $p > .10$. This means that the counting of immobile objects did not take significantly longer when the rooms themselves were perceptually indistinguishable; likewise, the perceptually indistinguishable rooms did not have a significant effect on the success rates of participants. The participants' ability to count moving robots was not affected by having two groups of perceptually indistinguishable robots rather than only one, and the measures did not even differentiate counting moving robots from counting immobile glasses by the participants.

The measures of the following tasks also showed similarities in performance among participants. There were no differences among the three subgroups based on the number of words vocalized, the number of room-mentions the participant made, or a ratio of the two. Among these subgroups, there were no significant differences on any of the quantitative measures of the protocols; in all cases, $p > .50$. However, the participants assigned to the base case task, following a moving object perceptually distinguishable from all other moving objects were more successful than participants whose focus object was perceptually indistinguishable from the moving distractors, $t(28) = 3.068$, $p = .005$.

Because of the lack of differences in the performance across several of the subtasks, the following analyses consider at most three treatment groups. These are the counting tasks, CT, the following tasks excluding the control, FT, and the control following task, CFT.

Success rates.

The counting tasks, CT, were solved by 76% of the participants. The following tasks excluding the control, FT, were solved by 58% of the participants. The control task, CFT was solved by 92% of participants. CT was easier than FT, $\chi^2 = 4.47, p = .0345$. CFT was also easier than FT, $\chi^2 = 4.71, p = .030$ even though the behavior of the object to be followed was exactly the same in these tasks/ The difference between CT and CFT did not reach significance, $\chi^2 = 1.41, p = .20$.

Most of the participants were given both a counting task and a following task; the order in which the two tasks were given was randomized. Overall performance on the second task was better than that of the first task,$\chi^2 = 7.5, p = .006$. Whereas 59% of the first tasks were solved correctly, 83% of the second tasks were so solved. Performance in all three tasks improved from the first to the second task, the CT success rate went from 69% to 84%; the FT success rate improved from 41% to 77%; and the CFT success rate went from 80% to 100% (only one participant failed the control task). In individual task analyses,$\chi^2 = 7.14, p = .008$. Also, although the success rate for CT was still better than FT when they were performed as the second task, it was not longer significant, $\chi^2 = .52, p > .46$.

Length of protocol

The time to perform the task was a constant in the following tasks since the participants were asked to follow a "tour guide" with a predetermined and constant path as it gave the "tour". This tour took 191 seconds. The time to complete the count tasks was a variable under the control of the participants (M=153.5, sd=10.5). This difference was highly significant, $t(63) = 3.91, p < .001$. Although the participants were asked to report their thoughts while performing the task, they did not respond at a constant rate, so the length of the protocols measured by the number of words uttered was a variable in both tasks. Although the number of words uttered was greater for the following tasks (M=284.4) than for the counting tasks (M=254.0), the difference was not significant, $t(63) = 1.368, p > .17$. This fact can be explained in that the rate of speech in the counting tasks (1.74 wds/sec) was greater than that of the following tasks (1.48 wds/sec), $t(62) = 3.18, p < .005$.

Room-mentions.

When the experiment was designed it was hypothesized that in order to count objects, participants would have to reason directly about the virtual space in which the participants were operating. It was expected that the participants in the counting tasks would refer to the relationships between the virtual rooms more often than those in the following tasks. T-tests were run for the following and counting tasks on both the number of times rooms were The number of room-mentions in the counting task($M = 11.56$) was higher than number of mentions in the following tasks, ($M = (9.34)$), but this difference was of marginal significance with a two tailed test, $t(63) = 1.98, p = .052$. The ratio of room-mentions to total words for these two groups of participants was highly significant, $t(63) = 6.86, p < .0001$. Participants clearly mentioned rooms for a greater part of their total speech when they were counting than they did when they were following an object.

After considering what rooms mentions the participants made, the coded transcripts were examined for sections that were coded to mark a participant trying to identify a room. This would be a code of infer(identity (<some room>)) or speculate (identity (<some room>). Sometimes the 'infer' code was omitted, but identity(<room>) was considered the same as infer (identity (<room>)). Identifying a room, (or considering its identity) was very important for those counting immobile glasses. Those who succeeded identified room more often than those who did not $t(33) = 5.904; p < .001$. Identification of rooms was also significant for all of the counting tasks. Those who were successful identified more rooms than those who were not $t(67) = 3.554; p < .001$. Surprisingly, even though identifying rooms was not strictly necessary for the following tasks, participants who succeeded in the following tasks also identified rooms more often than those who failed at the task$t(65) = 7.882; p < .001$. although there were more room identifications in successful counters than successful followers $t(94) = 2.91, p < .01$.

Role of experience.

Prior game experience, as reported by participants, had a significant multivariate effect considering all of the dependent measures, eg $F(20, 96) = 1.88 p < .025$. The linear trend of experience (from least to most in three steps) showed that for the following tasks, the more experienced the participant, the greater number of words uttered $F(1, 62) = 9.005, p = .004$ and the more likely the task was successful, $F(1, 62) = 8.610, p = .005$. The more experienced participants also uttered more words $(F(1, 62) = 9.005, p = .004)$ and mentioned rooms $(F(1, 62) = 6.097, p = .016)$, but the rate of the room mentions was not affected by experience, $F < 1, p > .25$. In the counting tasks experience was a significant only for the average time to complete the task, $F(1, 64) = 4.027, p = .049$. The more experienced were not more successful, $p > .25$, nor did the number of words uttered, number of room-mentions, or the ratio approach significance, all p's>.25.

All but the first three participants participated in both a counting and a following task. The order of the tasks was randomly assigned. The next check that was done was to see if experience with the first task performed influenced the participant's performance of the second. There was little influence on the counting tasks from performing a following task first. None of the measures of the counting tasks showed a significant effect from task order; all $p's > .30$. When the counting task was second the solution rate increased from 69% to 84%, but that did not approach significance, $t(65) = 1.47, p = .15$.

If the counting task was first and the following task second, the performance in the following tasks was affected in a significant way. Participants spoke more words in the following task ($t(61) = 2.76, p = .008$) and had more room mentions ($t(61) = 2.28, p = .026$) if the following task was second. As was noted above, the task time was fixed in the following so these difference imply that the participants spoke more during the same amount of time if this task were second. The solution rate improved, significantly, from 47% to 82% in the following task when it was the second task, $t(58) = 3.10, p = .003$.

3.4.3 Discussion of Quantitative results.

The most obvious conclusion from the quantitative results is that we now have specific data showing that people can identify individual objects that are perceptually identical to others. These results give clear support for anecdotal evidence that people often do identify individual objects that they could not perceptually discriminate from others. We also have evidence that although people might sometimes use the broader perceptual context to aid the discrimination, at most only minimal perceptual aliasing (Kuipers and Beeson, 2002; Kuipers and Byun, 1991), occurred in this study as there were no differences between the groups that had to identify objects in perceptually identical rooms from those for whom all rooms were perceptually unique. Thus one must use factors in additional to perceptual properties for identification purposes. The fact that first person game experience played a role in a number of measures is very informative. The rate of uttering words was a function of previous game experience. In the following tasks in which the time for the task was controlled, participants with more game experience talked more than those with less. In the counting tasks in which time was under the control of the participants, there was no difference in the number of words uttered as a function of previous game experience, but those who had more experience finished the task in less time. These results suggest that those with less experience had to pay more attention to the mechanics of the task and had less capacity available to apply to the cognitive task at hand. With less capacity available the less experienced participants in the counting tasks needed more time to do the task. Also, with less capacity available, the participants in the following tasks were more likely to fail. There is another additional interpretation of why experienced participants were more likely to solve the following tasks. Participants who were more experienced with the mechanics of first-person video games were better able to control their virtual avatar (their in-game persona, the virtual extension of themselves in the simulation). They were more successful at the following task in part because they were able to maneuver themselves better and thus be in position to make more use of the simpler cases of identifying PIOs (see chapter 4 for what those simpler cases are). Performance on a task in this study was somewhat similar to having previous experience with extra-experimental first-person games and probably due to the same factors. Participants talked more in the following task when it was second than they did when it was first and performance success on the following task was better under that condition. The results based on room-mentions and room identifications directly relates to the cognitive strategies that play a role in PIO identification. With PIOs the agent must keep track of where the objects are or where they may be. This often requires generating a mental map of the relative locations. Participants who used this strategy on the task were more likely to solve it than those that used other strategies. One set of results of this study may be important for how one interprets the protocol analysis method methodological import. The result is that the The protocol analysis method allows the researcher to look into the mind of the experimental participant as s/he reports on his/her thought processes.

3.5 Qualitative Results of the human trials.

3.5.1 An in-depth look at the performance of eight participants

In this section, I examine some of the qualitative data from the experiment. For completeness, I will first present, interspersed with an analysis, the complete protocol data of some prototypical individual participants. One transcript of a successful participant and one transcript of either an unsuccessful or an initially unsuccessful participant will be analyzed for each of the major task categories. The analyzed complete transcripts include only the transcript of the participants' performance of the task and does not include the retrospective interview.

In addition to the complete transcripts that are listed below, many of the other transcripts contain some very instructive subsections. The complete transcripts of all participants are available on the web in the University at Buffalo Computer Science and Engineering Department's technical report series as Technical Report 2005-1.

The participants' transcribed utterances are found in the left hand column of the transcript. As mentioned in section 3.2, the utterances were broken up into the smallest unit that would completely express a complete action, inference, or report of the participant. It was intended that the utterances be broken down so that each utterance expressed only a single action, inference or report. However, sometimes it was impossible to do so. For example, in the transcript for participant 42 doing task 2 of the experiment, participant 42 (P42) says, "So this is the third one I am seeing." This utterance encapsulates identifying the currently seen robot as a new robot and at the same time reporting the current intermediate result total. The participant's statement couldn't be broken down any further and still carry the complete meaning on its own.

The utterances in the transcripts were transcribed as faithfully as possible. This includes any false starts and non-word sounds that the participant made. When the participant spoke words or common abbreviations, those are used in the transcripts. When a non-word (for example P55 makes a hissing sound that turns into a dental blowing sound in the transcript below) is encountered, the sounds was transcribed as faithfully as possible using Roman letters with American English rules of pronunciation.

When the participant pauses in speaking, this pause is indicated in the transcript using a series of dots such as '....'. The number of dots indicates the length of the pause. The minimum two dot series indicates a pause of about three seconds. Every additional dot beyond the second indicates approximately one additional second of pause in participant vocalizing. If the transcriber could hear the sound of keyboard keys being typed while listening to the participant's tape, then the phrase "<silent navigation>" is added instead of (or in addition to if the keys come after some period of silence) the series of dots.

3.5.2 Counting Immobile Objects

Task 1 Version 1(counting immobile glasses in floorplan in figure 3.3), participant P55, successful participant.

The transcript for participant P55 is presented below, interspersed with an analysis. This participant counted glasses in the suite of rooms in which the walls of each room allowed the rooms to be identified purely on the basis of their perceptual properties. This participant successfully performed the task in a manner representative of those that were successful at this task.

As a reminder, the codes of the transcripts are of what the participants say. If the participant says something erroneous, a '**' is appended to the code indicating that the code, and hence the participant utterance, represents an incorrect belief on the part of the participant.

Participant Utterance	Coded
Okay	*Begin-task*
I see two drinking glasses on my left.	*See(glasses:2)*
<silent navigation>... <experimenter prompts participant> and I ssssffffffffff	*Move(currentRoom)*
don't see anything else, any other glasses here, in this room.	*See(other glasses:0)*
So I move into the room, on my right.	*Move(Room2)*
I don't see anything here.	*See(nothing)*
No glasses.	*See(glasses:0)*

Here the participant is beginning to search for focus objects, looking in the first two rooms for them.

So, after I move into the room, into the first- second room I move on	*Label(currentRoom)*
to the room on the right of that room.	*Move(Room3)*
And tch-ch-ch-cha. And it's the same room room I think which I saw	*Infer(identity(currentRoom))***
in the beginning,	
having, actually I missed one glass there. There are a total of three	*ReEvaluate(glass-count)*
glasses.	*See(glasses:3)*
One was hiding behind the bottle I guess.	*Rationale:ReEvaluate*

Here the participant has fallen victim to perceptual aliasing. The participant assumes that the glasses encountered in the current room are the same as the ones seen in the original (aliased) room (with the exception of the "extra" glass that P55 sees).

I go out of that room. <silent navigation>..	*Move(Room2)*
three,	*InterResult→3 glasses*
go out of the second room.	*Move(Room2)*
And now I come into a third room which I didn't, which I have not	*Label(currentRoom)*
seen earlier.	*Infer(identity(currentRoom))*
And this room has, one more glass.	*See(glasses:1)*

Here P55 decides he is in a new room and so decides that the glass that he sees is a new glass.

So a total of four glasses is what I have counted at present. <silent navigation> in three rooms. Four glasses.	*InterResult→4 glasses*
I go back. <silent navigation>	*Move(AdjacentRoom)*
Okay, I have four glasses.	*InterResult→4 glasses*
I don't thinnnnk I have any other rooooooms, here.	*Infer(map)*
hmmm yeah, <silent navigation> <experimenter reminds participant of instructions to stop navigating as an indication of a final answer.>	*InterResult →4 glasses*
Okay, okay, yeah okay-kay-kay-kay-kay-kay-kay-kay. hmmm four five. <sigh> I see four <sigh>	
Going back to the first room which I entered.	*Move(room)*
	*Infer(identity(room)) ***
Now I see four glasses here.	*See(glasses:4)*
So total of four plus one. Four glasses in this room and one in the other room on the chair.	*InterResult→4+1 glasses*
Ah total of five. <silent navigation> yeah okey.	*InterResult→5 glasses*
And theeeeen I go, go to another room	*Move(AdjacentRoom)*
which has, a table and two more glasses,	*See(Furniture)*
	See(glasses:2)
so now there's a total of seven glasses, which I've counted as, till now.	*InterResult →7 glasses*

P55 has now returned to the original room; however, he believes it is a new room, and so he considers the glasses seen here to be new glasses and identifies them as such.

I move into the room, another room.	*Move(AdjacentRoom)*
Ahhh okay, it seems to me I've already seen this room	*Infer(identity(currentRoom))*
and counted this one glass here.	*See(glasses:1)*
	Infer(identity(glass))

Here P55 identifies the room as being one that he has seen before, and automatically identifies the glass in the room as being the same one that he had previously identified in the room.

Ok I've counted this one too.	*Infer(identity(glasses))*
okkkkk. OK move into another room.	*Move(AdjacentRoom)*
This rooms seems to be all empty. yeah, <silent navigation>..	*See(nothing)*
ok seven.	*Report(Result)*

At the end of the task, P55 has identified each room and associated the glasses in each room with the room. He vocalizes less and less, simply moving through each room and assuring that the expected glasses are in each room.

Task 1 Version 2 (glass counting with floorplan from figure 3.4), Participant P3, An unsuccessful participant.

The transcript for participant P3 is presented below interspersed with an analysis. This participant counted glasses in the four-room suite of rooms in which the walls, floors, ceilings, and contents of the opposite rooms were identical. There was no perceptual cue to distinguish between the two rooms. Participants would have to make the determination on the basis of information about location possibly gained from "historical context" (as described by Kuipers and Beeson (2002)) information or some other external cue such as seeing two indistinguishable rooms at once.

There were so few failures in the counting of immobile objects that there was really no such thing as a "typical" transcript of an unsuccessful participant. This participant, however, unsuccessfully performed the task in an instructive manner representative of many those who had difficulty with the counting tasks in general.

Participant Utterance	Coded
ok,	*Begin-task*
<silent navigation> <experimenter prompts participant> ok, I'm looking around the table. Looking towards, the other corners of the room.	*Search(Room1)*
.. now I'm going into this other room. <silent navigation>	*Move(Room2)*
I'm checking out this room.	*Search(Room2)*
I don't see any.	*See(glasses:0)*

P3 begins the task much the way that P55 did above, by searching the current room for glasses and then moving to a new room adjacent to the current room. Though it is not yet clear from the transcript, P3 did see two glasses in the first room, and is counting from there.

So I'm gonna go out.	*Plan(Move(AdjacentRoom))*
<silent navigation> <experimenter prompts participant> oh oh I'm looking around this other room,	*Search(room2)*
which appears to be empty.	*See(nothing)*
oops, trying to find the door. ...	*Move(AdjacentRoom)*
	Navigational-Hindrance
looks like the same room. <silent navigation>	*perceptual-props(currentRoom, room1)*
	*Infer(identity room)***

In this section of the transcript, P3 searches the second room, finds nothing and then eventually entered the third room, the one diagonally opposite the starting room. P3 notices that the room looks like the original room and indicates that she believes that it is probably the same room in P3's colloquial speech.

oops, .. well <experimenter prompts participant> I'm going into this other room	*Move(AdjacentRoom)*
because, it looks exactly like, .. the same uh...	perceptual-props(currentRoom, room2)
ok I'm done. two glasses.	*Report-results: 2 glasses.*

P3 completes the square, visiting the forth and last room in the suite during this section of the transcript, but has perceptually aliased each of the rooms with the perceptually indistinguishable room opposite it. The suite for this version of the task actually has 4 glasses, two in the first room and two in the third. P3 aliased the two rooms and believed that the different glasses were the same as well.

3.5.3 Counting Moving Objects.

Task 2 Version 1 (counting mobile robots all of whom share the same appearance), P48, A successful participant.

P48's transcript of task 2, the robot counting task, is presented below interspersed with an analysis of the transcript. Participants used several strategies while counting moving objects, and P48 used strategies that were common to many successful participants, making this close to a prototypical transcript for this task.

Participant Utterance	Coded
OK	*Begin-task*
now I'm just, checking out the room to see, how big it is, where are	*Search(CurrentRoom)*
the openings.	*Search(map)*
I can see a table	*See(Furniture)*
and I can see a robot.	*See(robots:1)*

Here Participant 48 begins the task by creating a map of the starting area and noticing the objects in the starting area, including the objects that the participant is supposed to be counting.

Which is moving	*Report(Robot movement)*
and I'm trying to, <sigh> see what the robot is trying to do.	*Strategy(understand robot movement)*
It's hitting the walls, and it's, responding to that.	*Report(Robot Movement)*

Here the participant reports trying to understand the robot's capabilities and motivations. In this multi-room environment wherein the participant will count moving objects, the participant cannot rely completely on any truly simple cases in identifying objects, so the participant is trying to understand the capabilities and motivations of the target object in order to understand what its movement might be like while out of sight. The participant determines that the robots are moving mostly randomly.

and I'm leaving the room.	*Move(adjacentRoom)*
There are, ... hmmm .... ok ... I'll give the count later.	*InterResult→??*
<silent navigation> <experimenter prompts participant>	*Difficulty talking*
yeah, um I'm actually, trying to move fast into the rooms, before the robots can actually move out of the rooms so that uh, I can get a, an, get an accurate count of how many robots are there.	*Strategy(move fast enough to allow association of robot to a room.)*
<silent navigation> oooookay. oneeeeee, twoooo. Three. yeah, I'm still, four okaaaaaaaaaay,	*Difficulty talking* *see(robots)* *InterResult→4 robots.*
yeah I'm- I was just moving fast in and out of the rooms before the robots can actually move out of the rooms	*Strategy(move fast enough to allow association of robot to a room.)*
so that I get a- I get the correct count of robots.	*Rationale:Correct(results)*

After P48 determined that the robots seem to move randomly, the participant determines the speed of the robot to be less than his own. P48 then uses this fact to attempt to identify the robots based on their locations within a given room. P48 appears to be trying to effectively reduce the problem of identifying and counting moving objects to the task of identifying and counting immobile objects. At very least the participant has attempted to confine the objects to a single room-wide location.

I can see that there are four rooms. Each opening into two others, and there are two tables and, in in ah diagonally opposite rooms.	*Infer(map)*
And, in total, there are, about, <silent navigation> five robots in four rooms, which are hitting the walls and uh, uh responding to that and changing their directions, because of which they move in from one room into the other.	*Report(results)*

At this point the participant has completely (mentally) mapped out the rooms in the suite. The participant has also understood the capabilities and limitations of the robots the participant is identifying and counting, and successfully uses the speed and location strategies previously discussed to successfully complete the task.

Task 2 Version 1 (counting mobile robots all of whom share the same appearance) P37, an unsuccessful participant

P37 is fairly typical of those participants who failed at the task. I initially expected that those who failed would be using fundamentally different strategies than those participants who were successful. While a small number of participants did use either noticeably different or limited strategies, most unsuccessful participants used most or all of the same strategies as successful participants. Some participants, like P3 discussed previously as counting immobile objects, fell victim to perceptual aliasing. Some, like P37, simply failed to connect all of the dots or follow up on some effective sub-strategies with one that would tie them together to allow the participant to successfully solve the task.

Participant Utterance	Coded
I got two robots in this room	*See(robots:2)*
. .... Let me check out.	*Search(?)*
Two plus one. Aha. One in this room.	*See(robots:1)*
Okay..... Yeah. Yeah. I'm moving to another room.	*Move(AdjacentRoom)*
Ah. Here, I can see one more robot.	*See(robots:1)*
So this is the third one I am seeing.	*Infer(identity(robot))*
	InterResults→3 robots

P37 starts out in a fairly typical manner by searching for robots in the initial room and then moving to other rooms and repeating. However very quickly P37 starts to show the kinds of inconsistencies that will cause trouble in solving this task. In the above section of the transcript, P37 first mentions seeing "Two plus one" robots and then shortly afterward "I can see one more robot." but still reports an intermediate result of only three robots. This sort of human memory failing is one of the reasons for an unsuccessful result.

Hmm, okay. There are two robots.	*See(robots:2)*
Ahh this room is not the same as the previous one, I suppose.	*Infer(identity(currentRoom))*
Okay. Let me check out.	*Move(AdjacentRoom)*
Mmmm, this room has no robots.	*See(robots:0)*
Maybe, these robots must have moved from this room to that one. That might be a possibility.	*Speculate(robot movement)*

Here P37's expectation of finding a robot in the room has been violated, and so he formulates a theory about the robot's capabilities and movement that accounts for the violated expectation. Use of this knowledge of the robot's capabilities and movements would have proved very useful to P37 if he had used it later.

Let me see. Okay, there are two robots.	*See(robots:2)*
Let me check on the other one	*Move(AdjacentRoom)*
	Plan(Search (AdjacentRoom))
.......mmm, in all there are three rooms.	*Infer(Map)***

P37, has built a bad (mental) map of the suite of rooms that he is wandering in. (The floorplan of the actual map is seen in figure 3.3.) There is no direct evidence of perceptual aliasing from P37's vocalizations at this point; however, it is clear that he has built an incorrect map of the world, if not through perceptual aliasing, then at least through the kind of carelessness that he showed in counting robots in the first section of this transcript.

One, one robot is in here.	*See(robots:1)*
Two in here.	*[Move(AdjacentRoom)]*
	See(robots:2):
Okay, three robots.	*InterResult→3 robots.*
Oh man. Am I getting confused?	*doubt*
One with ... one in the room with the table.	*See(robots:1)*
	Label(CurrentRoom)
Ahh ..one, two, three....	*See(robots:2)*
	InterResult→3 robots.
And the room with white floor.....	*[move(AdjacentRoom)]*
	Label(CurrentRoom)
and one in the room with table.	*[move(AdjacentRoom)]*
	*Infer(identity(currentRoom))****
	Label(currentRoom)
Yeah. There is a room without table, right?	*Speculate(map)*
..... there's one room without table,	*Infer(map)*
one with white floor,....	*Label(room)*
okay, one, two, three,...three, four... <participant counts robots while moving rapidly through all rooms> <experimenter prompts participant>	*See(robots:4)*
	Strategy(robots seen in rapid succession are probably different)
	InterResult→4 robots

In this section of the transcript, P37 has trouble vocalizing and runs around a lot, mostly confusing himself even more about the layout of the map and the robots wandering in it.

yeah.. and I'm just trying to find them.	*Plan(searchfor(robots))*
I am trying to see them simultaneously,	*Strategy(see simultaneously →*
so that I'd be sure how many robots I've seen.	*different robots)*
Ahh..two at a time.	*See(robots:2)*
Once, I saw three at a time.	*(Memory(See(robots:3))*
Eh, this is the third one.	*See(robots:1)*
In some other room.	*Move(AdjacentRoom)*
Ahh fourth one.	*Infer(identity(robot))*
I'm very sure there are greater than or equal to four.	*InterResult→* $\geq$*4 robots*

In this section of the transcript, P37 uses the strategy that objects seen at the same time are different objects. This strategy has been used by other participants with good success as a sub-strategy in the counting tasks.

Let me check out.	*Plan(recheck)*
One must have moved in another room.	*Infer(robot movement)*
Two... First.....one...hm........	*[move(rooms?)]*
<participant counts robots while moving rapidly through rooms>	
<experimenter prompts participant>	
yeah yeah... I'm just trying to see them all at one time	*strategy(see robots simultane-*
so that I'm pretty sure about how many of them are there.	*ously)*
Well, I'm checking out in the rooms,	*Move(map)*
at the same time they are moving into other rooms,	*infer(robot movement)*
trying to confuse me.	*Ascribe(intention, robots)*

In this section of P37s transcript, he reiterates his desire to use the strategy of using simultaneous sighting to identify robots as different. He also notices for the second time in the task that the robots move between rooms, something that will potentially cause difficulty with his stated plan. He also ascribes an intention to the robots; however, it is not an intention that could be of any help to him in solving the task, so he justifiably doesn’t pursue the idea later in the transcript.

Let’s see. I’m pretty sure that they are greater than or equal to four ’cause I’ve seen three of them at a time, and the fourth one in some other room. It’s not possible for any three of them to move into that room in that short span of time so I’m pretty sure there are four.	InterResult$\rightarrow$ $\geq$4 robots *Strategy(see simultaneously $\rightarrow$ different robots)* *Strategy(identify new robot based on how far apart robots are and how much time is elapsed)*

Here P37 has hit upon a strategy that many other participants, including the successful P48 above, used successfully in performing the robot counting task, namely to move faster than the robots and identify the robots based on the idea that they could not move fast enough to get from the location where one robot was seen to the location that the other robot was seen. As will be seen below though, P37 does not successfully use this strategy to build on his earlier strategies, and his chances of success are therefore limited.

One, two.. I’m just trying to search	*search(map)*
Just trying to make sure whether they are not more than four?..	*[Move(map)]*
	Plan(search (map))
Yeah! They are four.	*InterResult$\rightarrow$4 robots*
.....<silent navigation>[experimenter prompts] okay sure sure..[experimenter says something] yeah sure.	*Difficulty Talking*
Two two twoooo,	*See(robots:2)*

Here P37 seems to be trying to use the strategy of moving fast enough that one robot could not arrive at the position of another robot between sightings, however to minimal effect. P37 self reports as having intermediate game experience. Perhaps he was unable to control his virtual-avatar well enough to make use of this strategy. However, P37 does not have the same kind of problems with navigating through the virtual world that many other game-novice participants had.

let me get into a corner and have a look at them.	*Strategy(see simultaneously → different robots)*
I'll do nothing. Let them move about.	*Plan(passive watching)*
One... One.	*See(robots:1)*
There were two in that room.	*Memory(see(robots:2))*
Okay. Three at a time. ... I saw three at a time.	*See(robots:3)*
I'm not being able to see more than three.	*Fail(plan)*
Ohhhh let me rush.	*Strategy(move faster than robots)*
Yeah! Fourth one is in here. I'm sure?.	*See(robots:1)*
	Infer(Identity(robot))
	InterResult→4 robots
Okay. Weeee....Yeah... mmmmmm,	*Difficulty Talking*
nobody movin' in here.....	*See(robots:0)*
okay. I'm sure. I'm sure (sigh) four robots are there.	*Results: 4 robots*

After trying the simultaneous-sighting and the move-fast strategies once again, the participant ends the task and give the incorrect answer of four robots (there were five in the task). P37 used several typical strategies that successful participants used, but was not able to put them together in order to successfully complete the task.

3.5.4 Following a Robotic Tour Guide

Task 3 (following a robotic tour guide with randomly moving PIO distractors), P49, a successful participant.

The following transcript from participant P49 is typical of many participants who successfully performed the robot-following task. The transcript is presented in its entirety, interspersed with analysis, in the same format as the counting transcripts.

Participant Utterance	Coded
I'm just following my robot,	*Follow(focus-bot)*
	[Move(adjacent-room)]
and trying to, approximate which of these, is mine by the... speed.	*See(PIO distractor)*
	See(focus-bot)
	Strategy(identify based on mobility and position)
Since I entered almost immediately, after the robot, I guess the one who is in front of me to be my robot.	*Strategy(identify based on mobility and position and time lapse)*

The robotic tour guide that P49 was following in this segment entered another room before P49 did. When P49 arrived in the room, there were two PIO robots in the room. Here P49 has to identify the tour guide to proceed. By the above statements P49 seems to be identifying the tour guide based on the observed speed of the robot and the amount of time since P49 last saw the tour guide and now sees a robot that might be the tour guide.

<silent navigation> so... ss I'm following where he's– this guy's taking me.	*Follow(focus-bot)*
What I'm trying to do is, keep the distance between myself and the robot to be the minimum so that, I can guess, where,	*Strategy(follow closely)* *Strategy(object continually viewed is same object)*
<silent navigation> .... <experimenter prompts participant> yeah I'm still in track of my robot,	*Infer(identity(focus-bot))* *Strategy(mentally track focus-bot)*
so I'm plainly following him. Always keeping the distance between me and him as less as possible	*Follow(focus-bot):closely*
so that, I know,	*Strategy(object continually viewed is same object)*
<silent navigation> at least that's what I'm trying to do,	*comment*
and hopefully I'm still with the right robot <chuckle.>	*Infer(identity(focus-bot)*
So I'm just following him.	*Follow(focus-bot)*
Keeping him in m-my line of vision. So that I don't lose him.	*Strategy(object continually viewed is same object)*

In this section of the protocol, P49 follows the focus robot through several rooms, keeping it continuously visible at all times. So long as the object is continuously in view, according to Pollock (1974), the object can be properly identified at a later time as the same object seen at an earlier time. P49 seems to use this technique heavily. (Other participants use it similarly heavily.)

I'm anticipating where he would go, so that,.... I'll always stay in close with him.	*Strategy(anticipate(focus-bot movement))*

Here P49 takes the previous strategy one step further. In an attempt to keep the focus robot under continuous observation, P49 tries to anticipate the movement of the focus robot and put himself in a position to continuously observe the focus robot after each of the focus robot's movements.

<silent navigation> ... I'm hardly paying any attention to any of the other robots since,	*See(PIO distractors) Ignore(PIO distractors)*
specifically concentrating on this robot.	*Strategy(mentally track focus-bot)*

Here P49 talks about specifically tracking the focus robot, perhaps through a FINST-like mechanism. P49 also claims to ignore distractors as much as possible in order to focus processing power on identifying the focus robot.

Oh, ... all right,	*Doubt*
	[See(distractor)]
he's still who I think he is <chuckle> <silent navigation>	*Infer(identity(focus-bot))*
so yes I'm following him. <silent navigation>	*Follow(focus-bot)*
... except for the first scene when he entered the first room, when	*Memory*
I temporarily lost sight of him, since then I think I'm following the same robot,	*Infer(identity(focus-bot))*
but, the first scene, in which, I like missed him by a couple of seconds,	*Doubt*
it's quite possible that some other guy m- might have crossed him and I might be following him.	*Speculate(fail(infer(identity(focus-bot))))*
<silent navigation> but otherwise I'm pretty sure I'm following the right person.	*Infer(identity(focus-bot))*

Here P49, after dealing with a somewhat troublesome distractor who was coming up from "behind" his virtual avatar, explains further that by continuously viewing the focus robot, he believes he has identified it correctly throughout the trial with the possible exception of a time when the focus robot was not continuously viewed.

There's nothing to distinguish, the- these robots from each other. Because, they look almost similar.	*Report(Perceptual-props(distractors)=Perceptual-props(focus-bot))*
<silent navigation> so <silent navigation> I'm still following him.	*Follow(focus-bot)*
<silent navigation> he's taking me into so many rooms,	*Report(focus-bot movement)*
I know know, <chuckle>	*[task ends]*

In this final piece of the transcript, P49 implies that he has tried to use the perceptual differences between the focus robot and the distractors to identify the focus-robot when he mentions explicitly trying to find a distinguishing visual difference between them.

Task 4 (following a robotic tour guide with path-following PIO distractors), P42, an unsuccessful participant.

Participants who were unsuccessful at the following task usually fell into two groups. The first group were those participants who knew that they had lost the focus robot (or at least knew that something was wrong). These participants had to have the simulation ended early because of this. The other group, nearly as plentiful especially in task 4, were those who never realized they were following a distractor robot in the course of the task. This sort of participant can be very instructive of the sort of mistakes that humans are very likely to make at this task. P42 is one such participant.

Participant Utterance	Coded
He's gone fast.	*Report(focus-bot movement)*
Ahhhhh?.. I can't catch up with you.	*Report (fail(follow(focus-bot)))*
Where are you goin'?	*Query(focus-bot)*
He came up that way.	*See(PIO distractor)*
[transcriber's note: This PIO distractor will be referred to as the focus-bot for the rest of this transcript since the participant has made this erroneous inference]	*Report(PIO distractor movement)* *Infer(identity(PIO distractor=original-focus-bot)***
How did he come up that way?	*Doubt*

Here the focus robot, which starts out in the hallway with the participant, moves fairly quickly into a room on the left of the hallway (the room labeled 1 in figure 3.8). P42, a novice with this sort of game-like environment, has not been able to follow quickly enough. A couple of seconds after the focus-bot entered the room on the left, a distractor emerges from a room a little further up the hallway and on the right of the participant (the distractor emerges from the room labeled 2 in figure 3.8.) P42 says in her retrospective that she believed that there was only one robot at this point in the task. P42 therefore assumes that this other robot must be the same one even though it seems unlikely that it would be able to move so far so quickly. The effect of a belief that an object is unique appears to be a very strong one with significant effects on at least some participants. P42 is representative of several participants who made this same mistake.

I can't catch up with him. He's going too fast.	*Doubt*
	Navigational-hindrance
(chuckles) ahhhh okay. That's how you do it. Okay okay. Okay okay.	*Understand(game-controls)*
That way we can pass her. Go turn to look.	*Move(near focus-bot)*
Now I'm goin' good	*comment*
well, I pressed the wrong button.	*Navigational-hindrance*
Huh.. follow her left	*Follow(focus-bot)*
ahhhhh?.. ahh, I can't get in there	*Navigational-hindrance*

In this section of the transcript, the novice game player P42 learns to control her avatar and follow the robot. The only oddity here is the way P42 switches back and forth between the genders when referring to the robot she is following.

Okay. Where did he go?	*Query(self)*
He must have went out the door.	*Strategy(infer(focus-bot path))*
I don't know where he went.	*doubt*
.... (giggles) maybe he went that way.	*Speculate(focus-bot location)*
Now I got to go forward, I think.	*Plan(Move(near Focus-bot))*
I don't know where he went.	*doubt*
Where did the guy go?	*Query(self)*
Where did you go? ...	*Query(followed-bot)*
here we are.. heeeeere we go.	*[See focus-bot]*

In this section of the transcript, P42 loses sight of the robot she was following and is forced to search for it. She forms a hypothesis about where it might be and searches for the robot in that location. Interestingly she tends to query to robot itself when she has lost it, as if the robot could hear her. Several other participants spoke directly to their robot as well. At the end of this section of the transcript, P42 spots the robot she is following.

Okay. He's going this way.	*Report(focus-bot movement)*
That means I gotta go backwards.	*Plan(move(backwards))*
Okay. ....No, I am doing right 'cause I'm going back behind him or should I turn around and get in- so I'll turn around.	*Plan(follow(focus-bot))*
Ah, this is okay? Okay [experimenter prompts participant] Sure.	*difficulty Talking*
He's going. okay okay. Okay. He's going back out .	*Report(focus-bot movement)*
I'll turn this way. Okay.....okay.	*Move(in currentRoom)*
Here we go.. Now I go. Okey.Here we go.	*BothMove(Focus-bot)*
Oh, I'm gonna miss him. Hah.....can't get in there!	*Navigational-hindrance*
I'll catch ya.	*Talkto (focus-bot)*
There's two of them!	*See(focus-bot)*
	See(PIO distractor)

At the end of this section of the protocol, P42 finally sees two robots at the same time, forcing her to given up her previously held belief that there is only a single robot in the suite. This indicates that seeing two objects simultaneously is sufficient to convince even people with very strongly held beliefs about the uniqueness of an object that there are actually more than one.

Now he's coming back.	*Report(focus-bot movement)*
I can go back. I'll catch this one .... okay!	*Follow(focus-bot)*
.... Will I just follow just one of them?	*Query(self|Experimenter)*
(giggles) .... I follow that one.	*Infer(identity(focus-bot))*
I think this is the right one..... all right.........there we go.	*Speculate(identity(focus-bot))*
Okay. [experimenter stops experiment] it wasn't it the right one? (giggles)	*Finish-task*

P42 now follows her robot (not the original robot, but the one that she has followed for some time) to the end of its path. Though her reasoning is not evidenced in her transcript here, she says in the retrospective that her reason for choosing the robot to follow was that she saw hers come out of a room first and decided that must be the one. P42 failed to successfully complete the task, but she did so in a way that provided an interesting insight into the kinds of assumptions that humans make and the effects on the identification of PIOs when those assumptions are wrong.

3.5.5 Following a virtual person as a tour guide.

Task 5 (Following a person), P65, a successful participant.

In task 5, participants followed a virtual person through the suite of rooms. The tour guide person and all of the distractors are perceptually distinguishable and can be identified solely on the basis of their appearance. P65 is fairly typical of successful participants in this task. His transcript is presented with analysis below.

Participant Utterance	Coded
<silent navigation> <experimenter prompts participant>	*Difficulty Talking*
There are around... eight to ten objects. <silent navigation> there are chairs. <silent navigation> there are around <silent navigation> sssseven more chairs, a blackboard, and a desk.	*See(Furniture)*
There's another guy shaking his hand.	*See(distractor)*
	Report(distractor movement)
<silent navigation> <experimenter prompts> yeah because I'm following the guy.	*Follow(focus-person)*

As P65 begins following the tour guide, despite early difficulty in navigating, he spends most of the time looking at distractors, and reporting his observations about those distractors. Compare this to P49, whose robot-following task is in subsection 3.5.4. P49's speech early in the task is mostly about the focus object and strategies for following the focus object. P65, in the same amount of transcript, mentions the focus person only once, and that because he was prompted after lapsing into silence. Otherwise P65 only discusses other things in the world. By the protocol analysis assumptions, we can infer that in this task, the participant seems to need much less processing power for identifying the focus object than in the previous task when the the distractors were PIOs.

OK there's a female.	*See(distractor)*
There's a big machine, on a table.	*See(Furniture)*
The female is dancing.	*Report(distractor movement)*
<silent navigation> Ok there's a frame- there's a, bulletin board on the, wall. I see uh, four cabinets.	*See(Furniture)*
There's another guy who's just entered the room.	*See(distractor)*
	Report(distractor movement)

All of the above section of the transcript was said in a single room, the room labeled 2 in figure 3.8. In this entire part of the task, P65 still never mentions the focus object, the tour guide. Even so, P65 is able to follow the tour guide and is still with the tour guide. It would appear that being able to identify an object purely by its perceptual properties makes the identification task much easier for people.

Now I enter into kind of cybrary.	*Move(adjacent-room)*
	Label(CurrentRoom)
There's another guy who's come over.	*See(distractor)*
	Report(distractor movement)
<silent navigation> There are around two comps [comps being the Indian English slang for computers -transcriber] in each desk. With two chairs... per desk. ... there are around...<silent navigation>. Eight rows...	*See(Furniture)*
I'm still following the guy.	*Follow(focus-person)*

Finally, after moving through the room labeled 4 in figure 3.8, P65 once again mentions the focus object, still spending most of his time discussing distractors and furniture in the rooms.

I see the female again. ..	*See(distractor)*
	Infer(identity(distractor))
who was dancing earlier.	*Memory(distractor movement)*
<silent navigation>.. see a, pop vending machine. With a table, and a kind of a microwave or something	*See(Furniture)*
I think it is a cafeteria.	*Label(CurrentRoom)*
.. I see the same girl again	*See(distractor)*
	Infer(identity(distractor))
who is dancing.	*Report(distractor movement)*

In this section of the transcript, P65 twice correctly identifies one of the distractor people. He does so despite the fact that the distractor person was out of his sight between each identification. Between the initial sighting of the distractor and the first sighting mentioned in the segment of the transcript above, the distractor was out of P65's sight for a significant amount of time. When a participant in task 4 (the task where the paths were all the same as this task, but with focus and distractor objects that were PIO robots) saw the robot in the same places that P65 sees the person in this task, the participants in the robot-following task always refer to that robot as "a robot" or "another robot" indicating that the robot-following participants have not identified it as being the same as the robot that they previously encountered. However, here, our assumption that people should all have a unique appearance is strong enough for P65 to identify this person twice as being the same person that he previously saw.

<silent navigation> now I enter a, rectangular, room	*Move(adjacentRoom)*
with nothing place,	*See(Nothing)*
I think it's a... it's a walk way or something it's a, I don't know gallery or something...	*Label(room)*
I enter the same room, with the cybrary from the behind door.	*Move(adjacentRoom)*
	Infer(identity(room))
There are two discrete machines on the wall. And around eight rows of machines, of tables with two chairs again. . around two machines per row.	*See(furniture)*
I see two guys,	*See(distractors:2)*
one wearing a blue suit. <silent navigation> and the other one, wearing a white shirt.	*Perceptual-props(distractors)*
He's, He's giving some kind of directions.	*Report(distractor movement)*

In this section of the transcript, P64 has moved back into the room labeled 4 in figure 3.8. He sees two more distractors and conceives of them, uses the perceptual properties of their clothing to distinguish them, and thinks about them for the first time.

I again enter the room	*Move(adjacent-room)*
	infer(identity(room))
with the bulletin board, there are two bulletin boards.... and some kind of a machine, something.	*See(Furniture)*

And at this point P64 follows the tour guide person to the end of his path and successfully completes the task.

Task 5 (Following a person), P37, the only unsuccessful participant in this task.

Task 5 was, as has been mentioned, the control task. Participants were able to identify the relevant objects based purely on perceptual properties. Though several had trouble with the task at one time or another, only one participant failed at the task in the end. P37 cannot be said to be prototypical of those who failed the task, therefore; however, he was very typical of some of those that had trouble initially.

Participant Utterance	Coded
I have to follow a person? Right? [experimenter confirms instructions]	*Query(experimenter)*
yeah.	*Begin-task*
He's entered this room.	*Move(focus-person, room2)*
Let me enter it. ??..	*Move(room2)*
ah there are chairs and everything.	*See(furniture)*

This participant starts out the task well enough, by following the tour guide person into the room labeled 1 in figure 3.8. There appears to be no significant differences between this participant and the successful one at this point in the transcript.

But I don't need to mention them right? Okay. [experimenter reads section of task instructions] Okay.	*Query(experimenter)*
I'm following him.	*Follow(focus-person)***
I don't know what he is doing??maybe, showing me some kind of direction or..maybe, I'm a student????.	*Speculate(intentions(person))*
maybe, this is not the person I'm following???	*Speculate(identity(person))*

In this section of the transcribed task, P37 incorrectly identifies a distractor person as the focus person and follows the distractor person instead. The distractor person's path becomes somewhat repetitive relatively shortly after the focus person leaves the room, however, and so P37 begins to doubt his choice. In particular, the actions of this distractor do not appear to be those of a legitimate tour guide, so P37 begins to doubt his identification of this person as the tour guide person. So even though P37 is unsuccessful, he still is aware of several strategies that can help him out in this task, strategies that have been used successfully by other participants. And thus he notices that something is wrong.

I'm moving out of this room.	*Move(room1)*
Let me follow this one......	*Follow(distractor)*
aha.......mmmm,	*doubt*
Let me see the computer lab in here....yeah..	*Move(room5)*
	Label(room5)
Is this the person I was looking for...	*Speculate(identity(person))*
I don't remember his face and structure.......	*Fail(memory(perceptual-props(focus-person))*
	Fail(identify(person))

[experimenter stops the experiment]

P37 continues to try to find the focus-person for a few moments more, seeing several people and not making decisions to following one of them for more than a few seconds. Finally P37 admits that he doesn't remember what the focus person looked like and cannot continue the task. In the retrospective, P37 mentions that he only noticed that the focus person (the tour guide person) wore a suit. Since there were two other suit-wearing virtual people in the suite, P37 was not able to identify the original person after losing sight of him. In this way, the task which could have been a purely perceptual task, became instead a PIO identification task.

3.6 Strategies reported

The coded transcripts revealed that participants reported only a relatively small number of strategies. Though some strategies went completely unreported in the main transcripts, and were only reported in retrospective interviews, (see Chapter 4 for more details) most of the strategies that participants used were reported by at least some of the participants. Table 3.1 shows the strategies that were found in the Strategy() codes in the coded transcripts. Appendix A gives the syntax and semantics of the code used to code the transcripts. Strategy codes contain English text with a rough gloss of the strategy, so that the coder would not be biased and try to make what was reported fit into what strategies were expected. Several unexpected strategies did emerge from an analysis of the strategy codes. The data found in Table 3.1 was generated using the following method. When a strategy code was found, I first checked to see if the text describing the strategy fit into any of the existing categories, if not a new one was added. If the participant was ultimately successful, then the successful participant count for this strategy was incremented. Likewise, if the participant was ultimately unsuccessful, the unsuccessful participant count for that strategy was incremented by one. Each strategy was only counted once for a particular participant. This prevented some strategies from having artificially high numbers since some participants repeated their strategies over and over. Each participant could have any number of different strategies incremented by one by using that strategy, but would only be counted once per strategy. Some participants reported more than one strategy, and some reported none. The numbers in Table 3.1 represent only the strategies reported by the participants during the task itself. Many of these strategies were used mostly successfully. The notable exception being that since there were no unique focus objects in most of the experiments, any use of the strategy of assuming that the focus object was unique usually led to failure. Also of note is the fact that participants who failed at the task tended to report fewer strategies, so it is less clear which strategies are more prone to failure than the data in the table might indicate. Finally, it is worth noting that the most often reported strategy, that of using an understanding of how an object is likely to move and through its motivations and capabilities, has a failure rate of more than a third. This was similar to the failure percentage (38%) of participants who counted or followed moving PIOs.

3.7 Conclusions

The experiment described in this chapter shows that though people can often identify PIOs in a variety of situations, there are also cases where people are not terribly good at it. With overall success rates of the non-control tasks ranging between 58% and 84%, people clearly do much better than chance at the task of identifying PIOs. There are still limitations in our ability to do the task. Furthermore, none of the tasks was set up to deliberately exploit known psychological biases such as having the object to be followed move behind a large object only to have it change directions while a different object emerged from behind the object in the same time it would take for the original to move out from behind the occluding object at its original

Strategy	Number of *successful* participants who used this strategy	Number of *unsuccessful* participants who used this strategy	Percent of Participants who were successful
Continuously viewing an object	10	1	91%
Continually perceiving an object (object out of sight short times, often that the participant was unaware of.	7	1	87%
Location: participant uses object location to identify it	12	2	86%
Speed: Participant reports using the object's speed to help identify it	10	2	83%
Anticipated movement: Participant guesses where the object will be soon, including what turns it will make	4	1	80%
Trajectory: Participant uses object's last known heading to identify object. (a subset of anticipated movement)	4	1	80%
Persistence of objects: participant identifies objects more than one time, if the count of identified objects matches, good, if not, persistence of objects says something is wrong.	4	1	80%
Time: Participant uses time since last identification to identify object	7	2	78%
Perceiving an object, then quickly perceiving it again a short time later	7	2	78%
Simultaneous Perceptions of more than one object	10	3	77%
Locally unique (participant believes that there is only one such object in the area)	3	1	75%
Understand movement/motivations of object.: participant reports using the objects perceived motivations, and/or the way the object moves to identify it.	17	8	68%
Unique objects (participant believes that object is truly unique - note that this was only true in the control task)	1	2	33%
random guess: Participant reports guessing randomly to identify object.	1	3	25%

Table 3.1: The strategies found in `Strategy()` codes

speed. Without deliberately making the task difficult, the success rates are still quite varied.

People use many strategies in identifying PIOs. Some of the strategies were those that I initially hypothesized, such as the speed/mobility of the object to be identified, its location, the time since an object's last sighting, and how common such objects are. Other unexpected strategies also emerged from the study. Participants occasionally used their beliefs about an object's intentions (as P37 did in section 3.5.5) to identify the object. When a participant deduced an intention and used it, that participant was often at least locally successful in the task. Participants also used strategies that were so simple that I failed to hypothesize about them. For example, some participants identified an object they were to follow simply by never losing sight of it; similarly, some participants lost sight of the object for such short and predictable times that the participants themselves often didn't notice that the objects had been out of sight at all. Other participants got a head start on the counting task by viewing several objects at the same time and thus inferred trivially that there were at least that many objects in the suite. Participants counting glasses used the fact that the glasses didn't move to rely on the glasses' location completely. Some participants in the following tasks also showed a strong bias toward believing that the robot that they followed was the only robot in the virtual world. This lead to both an ease of identification when these participants got lucky and followed the right robot, and to some interesting mistakes when they saw a different robot for the first time.

The study also showed that identifying PIOs is not a completely distinct task from identifying objects with a unique appearance. Identifying an object based on intrinsic perceptual properties appears to simply be one way of identifying an object, albeit one that many participants found relatively easy. As the transcript of P37's attempt to complete the control task (Task 5) showed, when an object that otherwise has a unique appearance is incompletely cognized, the act of identifying it later may well become an exercise in the more general problem of identifying a PIO.

Chapter 4

A Computational Theory of Identifying PIOs.

4.1 Introduction.

In the previous chapter, I discussed some experiments that showed how humans identify PIOs. In this chapter, I will present a computational theory of how an agent identifies objects in general, and PIOs in particular. The theory was developed from the human participants' transcripts.

It is important to note that no single transcript can be thought of as the prototypical transcript encapsulating the entire theory. No single participant was able to vocalize everything that he or she was doing. Also, no single participant found himself in all of the situations that a complete theory must account for. Some participants counted (and thus had to identify) stationary objects, while others had to count and identify moving objects, and so on. The theory is developed from the strategies gleaned from all of the successful participants' transcripts. Parts of an unsuccessful participant's transcript is also used if it meets two criteria:

1. Some successful participant also used the same strategy at some point.

2. The part of the transcript used represents a locally successful part of the participant's activity.

What I mean by number two is that, during that part of the transcript, the participant must be successfully identifying objects. This is most often the case when the participant initially counts the objects correctly (in a counting task) and then later makes an incorrect count during a recount. It might also happen if a participant initially follows the correct tour guide, but later loses the tour guide and begins following a distractor. In such a case, the part of the transcript recorded while the participant followed the correct tour guide contains data that is worth using in the formulation of the theory.

4.2 Intensional Representation.

In this dissertation the agent's beliefs and reasoning are based on an intensional representation (Maida and Shapiro, 1982). Intensional representations model the sense (Frege, 1892b) associated with an object rather than the object referent itself. The terms of the representation language used, SNePS (Shapiro and Rapaport, 1992; Shapiro and the SNePS Implementation Group, 2004), denote mental entities. Some such entities are propositions others are abstract ideas and others are the agent's "concepts" or "ideas" of objects in the world. This is important for the task of identifying PIOs because, before the identification task is complete, the agent may have two mental entities, e_1 and e_2, that it might or might not conclude correspond to the same object in the world. It is similar to the situation of George IV who, "wished to know whether Scott was the author of Waverly" (Russell, 1905, p 108). Or, if you see an elephant walking down the street with a parade today and then go to the zoo tomorrow and see an elephant, you might wonder if the elephant you see there is the same one you saw in the parade. In order to wonder about this, you must have two mental entities for the elephant, one for the elephant seen in the parade and one for the elephant seen in the zoo. You must have these two mental entities even if the elephant in the zoo and the elephant in the parade really are the same elephant in the world. In the case when two mental entities correspond to the same object in the world (the elephant that you see and the elephant in the street are actually the same elephant), then the two entities are said to have the same extension; we can say that the two entities are coreferential. Mental entities are the denotations of the symbols described by Coradeschi and Saffioti (2003) as part of the symbol anchoring process.

I will use "object" to refer to an object in the world and "entity" to refer to a mental entity that is the denotation of a SNePS term. In the task of "identifying perceptually indistinguishable objects", the agent may perceive an object in the world that might or might not be the same as a previously perceived object in the world. The agent's task is to decide whether the entity e_2 (think of "the author of Waverly") corresponding to the newly perceived object is coreferential with an entity e_1 (think of "Scott") that corresponds to a previously perceived object.

4.3 The Base Cases in the Identification of PIOs.

4.3.1 What makes a base case.

The experiment with human participants described in chapter 3 showed that there are four conditions under which human agents find the identification of perceptually indistinguishable objects to be very easy. I call these four conditions the base cases of the identification task. Participants in the experiment actively tried to put themselves into a position where they could use one or more of these base cases to identify the PIOs in the simulated world of the experiment.

When the computational agent identifies a perceptually indistinguishable immobile object using a base case (see below), it does not form a new mental entity for the object and then try to find (or "remember" to use a commonsense colloquialism) an existing entity with an equivalent extension. The agent only creates new entities as needed for cognizing information (Maida and Shapiro, 1982). The object that the agent is perceiving is either *the one* that it has seen at this location before, or a new, never-before-perceived object. If the object is *the one* that it has seen here before, then the agent ought to use the original mental entity for it and not conceive of something new that the agent believes is really the same thing in the world. If the object is a newly encountered one, a new mental entity is created for the newly encountered world object that the agent conceives of.

Human participants support the claim that new entities are not created when using base cases to identify PIOs. As an illustrative example, participant P55 is using a base case to identify PIOs when he says: "ahhh OK, it seems to me I've already seen this room and counted this one glass here." Upon seeing the glass, P55 immediately identifies it as *the one* he has seen before. Contrast this to P33 who is not using a base case to identify his PIO in the following utterance: "Where did the robot go? I think *this* is *the one*." In this statement, P33 indicates that he has one entity for the robot he is looking at now (vocalized as *"this"* in the transcript) and a separate entity for the robot he is looking for (vocalized as "*the one*" in the transcript.)

Note that these base cases are all based on the beliefs of the agent rather than facts about the world. For example when using the immobile objects base case, it is the agent's belief that the object is immobile that allows the case to be used. Each of the other base cases is based on the agent's beliefs about the objects that it is seeing. A last characteristic of base cases of identifying PIOs is that so long as the agent's beliefs that the base case holds is correct, the base case is a nearly foolproof method of identifying the PIO.

Each of the four base cases is described in its own subsection below.

4.3.2 Base Case 1: Simultaneous perceptions.

The first base case in the identification of PIOs is simultaneous perceptions of objects. Simultaneous perception is used to identify two PIOs as being different objects in the world.

If an agent perceives two perceptually indistinguishable objects in its sensory field at the same time, the agent can trivially conclude that the two are not the same object.[1] Unlike some of the base case strategies, participants were conscious that they were using this strategy of simultaneous perceptions and discussed its use while they used it. While counting moving robots, P37 states "I'm trying to see them simultaneously." P4, while doing the same task, is even more explicit when she states that she saw "The same two robots at the same time, so I know that there are at least two robots here."

4.3.3 Base case 2: Objects with a unique appearance.

If the agent believes that an object has a unique appearance and there are no other PIOs in the world, then the agent can instantly identify the object. The agent has only one entity for an object of this appearance and believes there is only one such object, so the agent should immediately use this entity to refer to the object whenever the object is encountered. Thus, like other base cases, the agent can and ought to use its original entity for the object in this case.

Participants were often aware enough of their use of this assumption of unique appearances to try to verify the assumption when possible. P15, when counting robots when there were two groups of perceptually indistinguishable robots, says "And I see the clown dalek here. aaand the little black and white one I don't.. annd a clown here - is that the same clown?"

The belief that an object is unique is subject to being "overruled" by a case of simultaneous perceptions. If an agent believes that an object has a unique appearance, but then sees two PIOs with that appearance, the agent will have to put aside the belief that there is only a single object with this appearance. For example P9 in the tour guide following task ended up following a distractor after never seeing more than one robot

[1] In this dissertation I am ignoring the use of illusions with mirrors and other deliberate attempts to make a single object appear to be multiple objects.

at a time. When P9 sees another robot, he makes the following statement: “So which one am I supposed to follow? There are two robots now....”, Indicating that he clearly abandons his belief in the unique appearance of the robot. That P9 had that belief is born out in the retrospective interview done a few moments later with the following exchange:

> **Experimenter:** What strategies did you use and why did you choose the robot that you chose to follow?
>
> **P9:** Well I had no clue that it’s a different robot. If I had known that there were more than one robot, I probably would have been more careful but I didn’t know.

The use of a single mental entity for an object believed to have a unique appearance was particularly noticeable when the participant’s assumption that an object has a unique appearance turned out to be incorrect. While trying to follow a robotic tour guide who turns into a room on the left of a corridor P42 says “I can’t catch up with you. Where are you going?!” And then a second later as a second robot emerges from a room on the right of the corridor a little further from the participant “He came up that way. How did he come up that way?” The participant clearly seems to be using the same mental entity for both robots and believes that there is only one.

4.3.4 Base case 3: Immobile objects.

Immobile objects are defined here as those objects which cannot move or be moved. I’m also including those objects which humans expect cannot be moved, even if such an object might be moved by using a rarely used technique. For example, people do not expect things like houses and other buildings, or even large trees, to be moved intact from one place to another, even though it is possible. I will also include in the category of “immobile objects” those which are prevented from being moved in the ordinary context and are considered such by the average person. For example a park bench that is bolted to the ground can be considered immobile by most people. Likewise a projection computer that is secured to a cabinet which is in turn secured to the floor is considered immobile by most people, though not to a thief with bolt cutters and a hammer. Specialized environments like computer simulations can also make objects immobile that might not otherwise be considered immobile.

Since the location of an immobile object does not change, location is the most important feature that allows an agent to identify immobile PIOs. In order to identify an immobile PIO, the agent must first recognize what kind of object it is perceiving. Next, the agent needs to reason, or realize, that objects of this kind are immobile. Then the agent cognizes the location of the object. At this point the agent can identify the object. Either the agent knows about an immobile object of this kind at this location, in which case it now identifies the current object using the entity that denotes that previously seen object, or the agent has never encountered one of this kind of object at this location, in which case the agent identifies the object as a newly encountered object and creates a new entity to refer to the object which has that description and is at that location.

The human-participant experiment supports the claim that location is of paramount importance in identifying immobile PIOs. Human participants find the use of location information so intuitive that they rarely notice it at the conscious level. When human participants were asked to discuss what they were doing and why while counting immobile PIOs, they never mentioned the object's location as being important during the task, even if they were clearly using location information. However, when asked in a retrospective interview, participants were able to articulate that location information was what they were relying on. The following exchange is representative. It was taken from a retrospective interview following an experimental task in which participants were asked to count glasses; The glasses were immobile in the simulated environment and recognized as such by participants.

> **Experimenter:** how were you able to distinguish between the glasses even when they looked the same?
>
> **P33:** ah because they are lying in the different rooms. That's why. They are different.

An agent cannot use only an object's location to identify that object as the appropriate mental entity. The agent must still recognize the object as belonging to the same class of (perceptually indistinguishable) objects as the previously encountered object. There are two reasons for this. The first reason is that any object might be destroyed: a house might burn down; a tree might blow down in a hurricane. Since an object might be destroyed and some other object take its place (perhaps a gazebo in the place of an ancient tree), the object itself needs to be identified as being perceptually indistinguishable from the previously seen object. For example, an agent with a perfect dead reckoning mechanism, if told "go to the Civil War monument.[2]" cannot simply turn off its sensors and go to the place that it last saw the monument. Location is only sufficient to identify immobile objects. So an agent must recognize the object as being of a class of objects that are immobile in order to take advantage of location as the distinguishing factor for identifying an object.

The use of the original entity is supported by the human-participant data in the immobile object case as well. While performing the glass-counting task, no participant who was sure about what room he/she was in expressed doubt about the identity of a glass. The glass was either referred to as the same one seen previously or it was referred to as a new glass. This contrasts with mobile objects, where participants often clearly seem to have more than one entity for an object and can talk about both entities. To reuse the example from subsection 4.3.1 above in more context, when following a robotic tour guide in a suite with several perceptually indistinguishable distractors, P30 briefly loses the tour guide robot and then makes the following statement "Where did the robot go? I think this is the one".

Earlier it was noted that the use of a single entity is contingent on an agent correctly identifying its current location. Our participants were vulnerable to mistaking one room for another if the two looked similar. Kuipers and his colleagues (Kuipers and Byun, 1991; Kuipers and Beeson, 2002) call this sort of mistake "perceptual aliasing" and have discussed the problem and a solution for robotic agents. When our participants fell victim to perceptual aliasing, use of location information to identify immobile objects was fallible. Sometimes participants would notice the possible aliasing, such as when P20, while counting glasses, says "I'm just, just curious to whether or not this is the same room. So I'm going to go back and retrace that, my steps." Participants who fell victim to perceptual aliasing and never realized it generally failed at the identification and thus the counting tasks.

[2]Civil war monuments throughout New England are almost universally based on the very same statue and are for most people, perceptually indistinguishable from one another.

4.3.5 Base Case 4: Continuous viewing.

Pollock (1974) has discussed reidentification of objects, a problem related to that of identifying PIOs. He notes that an object under continuous observation can be reidentified at a later time as being the same object, in particular, that "continuity of appearance is a logical reason for reidentification."

Continuous viewing of an object also appeared in the human-participants trials as a base case for identifying PIOs. Continuous viewing, like location, is used to identify an object as being the same as a perceptually indistinguishable object seen earlier (Pollock's reidentification). This ease of identification of object while under continuous observation seems to be implicitly assumed in Coradeschi and Saffiotti's 2003 *Track* functionality.

More concretely, the continuous viewing case applies if an agent views an object at position p_1 and later observes an object that is perceptually indistinguishable at position p_2. If the agent has continuously viewed the object as it moves from p_1 to p_2, the agent may assume with great certainty that the object it is currently seeing at p_2 is the same object that it originally saw.

Human participants tried to use this base case as often as possible when asked to follow a virtual robotic tour guide through a suite of rooms that also contained several perceptually indistinguishable robots serving as distractors. Like the simultaneous-perceptions case, participants were aware enough of this strategy to report it while performing the task. P7, after an early bit of difficulty, says "And I am following him very closely. And I am not going to lose sight of him this time." P23, is also very specific about using continuous viewing: "So I'm just staying, uh, close to this robot keeping my eye on him."

4.4 Intermediate cases of PIO identification.

4.4.1 What makes an intermediate case.

It has been pointed out[3] that the base cases described in section 4.3 represent primarily perceptual cases of identifying PIOs and that there were likely to be simple cases that do not rely on purely perceptual mech-

[3] My thanks to the anonymous reviewer of a paper based on this dissertation who did so.

anisms for the identification of PIOs. Looking at human performance during the experiment described in chapter 3, one can see evidence of non-perceptual cases that are similar to the base cases. In fact, for every perceptual base case, there is at least one non-perceptual simple case which can be closely identified with the base case. I will call these associated non-perceptual cases “intermediate cases”. They are so named because they are between the largely perceptual base cases and a mostly cognitive general PIO identification mechanism. Like the base cases, intermediate cases are chosen based on the beliefs of the agent, not something that actually occurs in the world. Therefore, like the base cases, the intermediate cases might lead the agent to make an incorrect identification if the belief that triggered the use of an intermediate case was erroneous.

4.4.2 Intermediate Case 1: Rapid perceptions.

The first intermediate case is related to the base case of simultaneous perceptions. In that case, seeing multiple objects at once was sufficient to assure that there are multiple objects in the world. In the rapid perceptions case, on the other hand, the objects (usually two of them) are not perceived at the same time, but rather in rapid succession, with no PIO encountered between the two perceptions. The Rapid Perception case assumes directional sensors since if the agent had omni-directional sensors, the agent would have simultaneous perceptions of object in front and behind it. As in the case of simultaneous perceptions, the rapid perception case is used to prove to the agent that two objects are not the same.

Participants in the experiment sometimes used rapid perceptions to disprove a hypothesis of unique appearance, as P18 does in the following transcript excerpt.

> Going into the next room, there is a multi-colored robot, and one who looks like the last one. I’m turning back, that robot is still in the other room so I know that these are two distinct robots.

Prior to this excerpt, P18 has seen only one robot, a silver-gray robot. As he enters another room, P18 sees a “multi-colored” robot as well as a silver-gray robot. In order to identify this silver-gray robot as a new, never before seen robot, P18 looks back toward the place where he last saw a silver-gray robot. When he sees a silver-gray robot in the previous location as well, P18 assumes (correctly in this case) that the robot seen in the current room is different from the one he looked back to see.

In order to take advantage of this rapid-perceptions case, an agent must see an object Ω, then must turn at least as fast as objects of type Ω can move, turning no more than 180°, and must see another object that looks like Ω. The agent must turn at least as fast as the object can move because if the agent turns more slowly, there is the chance that the object will be able to move to the new position before the agent views it. For example, if the agent turns so slowly that the object can move ahead of the agent's gaze, then this intermediate case does not hold. Likewise, if the agent turns more than 180°, then the object Ω could possible move around behind the agent and be seen again when the agent stops turning. In this case, again, the agent cannot use the intermediate case of rapid turning. However, if all of the conditions hold, the agent can determine with a very high degree of confidence that there are two different PIO objects in the world.

4.4.3 Intermediate Case 2: Locally Unique Objects.

An agent can often easily identify an object without the object being truly unique in the world, or even believed to be by the agent. It is only necessary for the agent to believe that an object is unique in the current context. For example, suppose you know identical twins, but you know that one of them is in the army posted abroad for the next six months. If you see someone that looks like these twins in town tomorrow, you can immediately assume that this person is the second twin. Of course, as with the unique items base case discussed above, the simultaneous perceptions base case described above will trump a belief that an object has a a locally unique appearance. In such a case, the agent must put aside the belief that an object has a locally unique appearance. Continuing the above example, if you knew that one of the twins was stationed abroad, but you saw two people that looked just like the twins, you must then put aside the assumption that the twins' appearance is locally unique. You may realize that something unusual happened and ask the twin about his sudden return, but you must give up the assumption of of locally unique appearances.

Participants seemed to use this assumption that locally unique objects can be effortlessly identified as the same thing seen previously (using the same mental entity) when they could. Sometimes the assumption of local uniqueness of appearance would be limited to a single room. For example, P12, while following a robotic tour guide in a suite of rooms with PIOs as distractors says "I'm stuck, OK but there is only one robot so I can follow *it*." P23, doing the same task, says something similar "There aren't any other robots in this room so *it*'s a little easier to follow." In both cases, the participants thought that the robot that they were following was the only object with the robotic appearance in the room. When entering the room they see only one object with that appearance and so they automatically identify this robot as the one they have been following.

4.4.4 Intermediate Case 3: Stationary Objects.

The next intermediate case is related to the base case of immobile objects. Stationary objects are those objects that cannot move themselves and are not easily moved by a breath of air. A helium-filled balloon is not a stationary object, even though it cannot move itself. On the other hand, many of the objects that we come into contact with in our daily lives are stationary: Lamps, computers, textbooks, and similar objects are all stationary objects. Their position will not change (or at least people do not expect it to change) unless there is an animate object to move the stationary object. P31 explicitly pointed this out in a retrospective after counting glasses in task 1 of the experiment:

> **Experimenter:** What strategies did you use to do this task?
>
> **P31:** Mmm I guess I just kind of based it on the fact that they would be stationary throughout the rooms and there was nobody else in there.

In the absence of a mover, stationary objects can be treated just like immobile objects; that is, location becomes the paramount criterion for identifying the object. The lack of another agent capable of moving a stationary object is something that a PIO identifying agent must reason about.

4.4.5 Intermediate Case 4: Continually 'Perceived' Objects.

It is well known (Johnson, 1998) that young children will identify briefly occluded objects as being the same objects. The participants in my experiment overwhelmingly did likewise. Though participants may have briefly lost sight of the focus object by looking away or having the object occluded, the participants nonetheless knew where the object was and looked for it "where it ought to be" when they viewed the object again. Most of the time, participants were not even aware that they had lost sight of the object in question.

4.5 Identifying PIOs in general.

Identifying PIOs is trivial and intuitive when one of the base cases can be applied, and can often be done without great mental effort when one of the intermediate cases can be applied. When none of the base or intermediate cases holds, the task can be much harder. An agent usually requires several more pieces of knowledge to identify mobile objects which are not continuously viewed. If people need to identify an object represented by the mental entity e, experiments show that they use knowledge of how rare or common they believe objects that look like e are. They will also use their beliefs about how fast the objects like e can move and the time between the time, t_1, that the agent last encountered an object it thinks might have been e and the time, t_2, that the agent sees e itself. Humans will also use the motivations of the object being identified if they can infer any.

Humans participants seem to use general beliefs formed from observations of the world. The most salient is information about the class of objects to which the PIOs being identified belong. These include things like: how fast or slow do objects of this kind move? [For example, P8 while counting moving robots: "I think that's the guy I counted already because, ah well he- uh couldn't have moved that fast"] Has an object of this kind ever been known to change speed? [P6 asked in a retrospective why he chose to follow a particular robot: "It's possible that it changed speeds, but it didn't really appear to do so throughout the game"] Have I ever identified more than one object that is perceptually indistinguishable from this one? [P18 while counting robots in a condition with two distinct groups of perceptually indistinguishable robots: "Because I thought maybe the multicolored robot had traveled, into that last room that I just searched, but it looks like there are two multi colored robots."]

Human participants also use information from observations of the specific objects being identified. Beliefs formed from these observations include beliefs about where and when the agent last encountered a PIO that the participant believes might be the PIO that the participant is currently looking at. [P25 counting robots with two distinct groups of perceptually indistinguishable robots: “I am entering the third room .... I can find the third robot, but I guess this is the same one as the first one but the room is different”] Another belief formed about the object itself is the answer to the question: “Does the object appear to have a particular purpose or motivation and if so, what is it?” [P10 following a tour guide “There are a total of three robots in here now. But...[4] and they seem to be moving randomly.”] The direction or trajectory that the object is moving in is important when an agent is trying identify a PIO only a relatively short time after encountering another PIO [P18 following a robot “He hasn’t changed directions, so I can still tell which one is him”] It is also important for the agent to have some awareness of where other PIOs are in the area to make sure that it doesn’t get unnecessarily confused if the object it is focusing on moves too close to one of the others.[P23 following a robot “So I just cut in front of that robot, in order to keep following mine.”] Successful participants like P23 would often keep some awareness of nearby PIOs and act to avoid occlusion of their focus object by other PIOs.

4.6 An algorithm for recognizing objects.

4.6.1 Assumptions

Let us suppose that our agent has seen an object with appearance d at location p at time t. The agent needs to identify the object. Either the object is a new one, or it is one that the agent has seen before, in which case the agent already has an entity representing the object. When the agent already has one or more entities representing objects with description d, the agent must recognize which entity, if any, represents the same PIO using reasoning.

We assume for simplicity that the object’s speed, if known, will be constant. If the agent doesn’t know the speed of the object, it will probably not be able to decide if the object it is perceiving has been previously encountered or not.

[4]When sequences of two or more dots appear inside of a quote from a participant, it indicates that the participant gave a noticeable pause at that point. The number of dots indicates the length of the pause.

4.6.2 Helper Functions

The PIO identification functions assume that several support functions are available. In the functions below, the following convention is used for the types of the parameters to these helper functions. A parameter name that ends with 'List' is a list containing list elements of a type specified by the first part of the parameter name. Parameters named E (or $E_{\text{<number>}}$) are mental entities. A parameter name with 'E' prepended is a data structure of mental entities. P and $P_{\text{<number>}}$ are places in the world. T and $T_{\text{<number>}}$ are times. Parameters named 'Class' must be a class of objects. The type of a parameter named 'Desc' must be a perceptual description of an object.

AllPairsWithFirst(PairsList) **returns:** *A set of pairs that all have the same first element as the first pair in PairsList*

effects: *as a side effect, removes the returns set of pairs from the* PairsList *input.*

Believe-equiv(E_1, E_2) **effect:** *creates a new belief that E_1 and E_2 refer to the same object in the world. (Trivially implemented with a SNePS believe act)*

BelievePossiblyEquiv(EPairsSet) **effect:** *creates the new belief that the first element of each pair in the set (the first element of each equivalence pair must be the same entity) possibly represents the same object in the world as each of the entities which are the second elements of* EPairsSet. *This procedure is used when the agent is unsure of an identification. Rather than losing all information achieved by the algorithm below, making this belief of a possible equivalence allows the agent to later use reasoning to determine the actual answer if the agent chooses*

ClassOf(Desc) **returns:** *the class of objects that "look like" Desc. (implemented by checking the alignments list in the SNePS/GLAIR model)*

ClosestToActualPosition(PairsList) **returns:** *an entity which is the second entity in one of the pairs of the pairs list. (The list must be a list of pairs of entities; the first entity must be the same in all of the pairs.) The entity returned is the one second element of a pair that the agent believes is most* *likely to have traveled from its previous location to the current location of the entity that is the first element of each pair. If the agent cannot find any reason to believe that one entity is more likely than another, then this function returns* null*, the non-entity.*

ContinuouslyPerceivedPIO(Desc, ESet, P) **returns:** *an entity from the set of entities ESet. The returned entity corresponds to the object with the description Desc that the agent has perceived as being the same from some previous place to the place* P. *This function is similar to* ContinuouslyViewedPIO *(see below); however, there is no requirement that the object be viewed continuously. Rather, if sight of the object is lost for a few seconds, the tracking mechanism will continue to track where it "ought" to be and continue to accept the object perceived there as the one being tracked. If there is no such entity, then the function returns* null*, the non-entity.*

ContinuouslyViewedPIO(Desc, ESet, P) **returns:** *an entity from the set of entities ESet. The returned entity corresponds to the object with the description Desc that the agent believes it has viewed continuously from some previous known place, to the place P. If there is no such entity then the function returns* null*, the non-entity. This function will rely on a FINST-like* (Pylyshyn, 1989) *mechanism for tracking continuously viewed objects.* Sandewall (2002) *has implemented such a system using* tracker objects *and* activity demons *to supervise the tracker objects.* ContinuouslyViewedPIO *will return true if there is a tracker object that has continuously viewed the object at* P *from some previous spot.*

CouldReach(E, P, T) **returns:** true *if the agent believes that the object corresponding to entity E could have arrived at the place P by time T. This function should only be called from the function* IsSameSmallD *(explicated in section 4.6.3) That function assumes that the distance traveled between the previous sighting and the current one is relatively small. A relatively simple motion planning algorithm should be sufficient in this case. This function can be thought of as a simple application of the* ShortestKnownPathBetween *(see below) helper function, followed by a quick calculation to see if* E*, at its rate of movement, could travel the shortest path from its previous* *sighting point to* P *by time* T.

Disallow(MSet, P_1,P_2,T_1,T_2) **returns:** true *if the agent believes that any element of the set of motivations, MSet, of an entity would disallow the entity from being at location* P_2 *and time* T_2*, given the agent's belief of a previous encounter at time* T_1 *and location* P_1*; otherwise the function returns* false.

HeadedToward(E, P) **returns:** true *if the object corresponding to entity E was headed in the direction of the place* P *when the agent last observed the object; otherwise returns* false.

Immobile(Class) **returns:** *true if the agent believes that members of Class are immobile and false otherwise. (Trivially implemented with a SNePS query)*

IsUnique(Desc) **returns:** true *if the agent believes that there is only one object in the world that has the appearance Desc; otherwise the function returns* false. *(again a simple SNePS query)*

IsUniqueInContext(Class) **returns:** true *if the agent believes that there is only one object in the current context that has the appearance of the class of objects Class; otherwise the function returns* false. *(Trivially implemented with a SNePS query)*

JustSeen(E) **returns:** true *if the entity E was seen before a turn that was faster than E's rate of movement and less than 180° from E. This is an implementation of the intermediate case of rapid perceptions using reasoning to make the determination.*

Loc(E) **returns:** *the location where the entity E was last encountered. (Trivially implemented with a SNePS query)*

MakeEntity(Desc, P, T) ***returns:*** *a new mental entity to represent the object with the description Desc, which is believed to be at location P at time T. (Trivially implemented with a SNePS believe act)*

MotivationsOf(E) **returns:** *a set containing any motivations that the agent believes the entity E has.*

MustIdentifyNow(E) **returns:** true *if the agent believes that it must identify the entity E at the current now (see (Ismail and Shapiro, 2000) for the treatment of time and 'now' being used for this agent) and* false *otherwise.*

NotAutoMobile(Class) **returns:** true *if the agent believes that members of Class cannot move on their own.*

OtherKnownPIOs(E_1, E_2) **returns:** true *if the agent knows about other entities that are perceptually indistinguishable from the two PIOs E_1 E_2 and which are not coreferential with either; otherwise returns* false. *This function is implemented through a combination of SNePS queries (for entities that belong to the same category and have the same appearance as* E_1*) followed by two calls to remove-equivs on the resulting set of entities; the first call will remove those entities that are equived to* E_1 *and the second call will remove those entities that are equived to* E_2.

OtherPossMover(Class) **returns:** *a possibly empty set of all of the entities that the agent knows are in the area that could have moved something of the class* Class.

PIO-AtLoc(Class, ESet, Loc) **returns:** *If an object*[5] *of Class has been seen at location Loc before, then the entity (from the set of entities ESet) denoting that object is returned; otherwise null (the non-entity) is returned. Note: The agent may still fall victim to perceptual aliasing* (Kuipers and Byun, 1991; Kuipers and Beeson, 2002)*, and so if the agent is mistaken in its beliefs about the location, then the agent is likely to reach the wrong conclusion about the identity of the object. (Trivially implemented with a SNePS query)*

RandomSecond(PairsList) **returns:** *An entity which is the second element of a randomly chosen pair from the parameter* PairsList.

RateOf(E) **returns:** *the speed or rate of movement of the entity E. In a real hardware robot this can be found with a simple radar gun. In a simulation the radar gun can also be simulated. The agent will then believe that the object denoted by E moves at the appropriate rate.*

[5]Only one object of a particular class can be at a single location at any one time. One might think of a cabinet and the fancy china plate inside it as being at the same location, however you cannot have two plates at the same location.

RemoveEquivs(E, ESet) **effect:** *removes from the set of entities ESet, all of those entities that are coreferential with the entity E. (The existing SNePS function remove-equivs)*

RemoveEquivsFromList(E, EPairList) **effect:** *removes from the list of pairs of entities EPairList, all of those pairs whose second element is coreferential with the entity E. (Implemented with a map over a list, removing the appropriate elements from the list.)*

ShortestKnownPathBetween(P_1, P_2) **returns:** *the length of the shortest route (that the agent knows about) between the two positions P_1 and P_2. This function is done at a sub-cognitive level. The path is calculated by using a simple path planning algorithm like the one proposed in (Lozano-Perez and Wesley, 1979). Once the path is available, the length of the path is calculated from the lengths of the path's straight-line segments.*

Time(E) **returns:** *the time when the entity E was last encountered. (Trivially implemented with a SNePS query)*

4.6.3 PIO Identification Functions

Below is an algorithm based on human-participants experiments for recognizing if a currently perceived object is new or is the same object as a PIO that was seen earlier. The algorithm is given below as five functions written in pseudo-code. The five functions, in the order presented, are Recognize, IdentifyNonMovingObjects, IsSame, IsSameShortD, and DoubleCheckUnCertian.

The function Recognize takes an object description, a set of positions of all of the current sightings of objects with that description, the time of sighting, and a (possibly empty) set of entities representing objects with this description that have already been encountered. The function returns a set containing mental entities corresponding to each object currently seen. Due to the simultaneous perception base case, there will be one entity returned for every element in the set of places where such an object is currently viewed.

function Recognize(D, P, T, E) **returns** *a set of entities*

inputs:

D: description of an object in the world,

P: set of places that agent currently perceives objects that have description D,

T: the time of the perception, and

E: set of previously created entities that have description D

eSet ← { }

First check the case of no PIOs - first time the agent sees something with description D

if |E| = 0{

for each ($P_i \in$P)

eSet ← eSet + MakeEntity(D, P_i, T)

return eSet}

Next check the base case of unique objects and

the intermediate case of those unique in the current context.

if (|P| = 1 & |E|=1) & (IsUnique(D) $\bigvee$ IsUniqueInContext(D))

return E

Neither immobile nor stationary objects move on their own, both the

intermediate (stationary) and base (immobile) case will be

considered in IdentifyNonMovingObjects

if NotAutoMobile(ClassOf(D))

return IdentifyNonMovingObjects(D, P, T, E)

Next check the base case of continuously viewed objects

for each ($P_i \in$P) {

E_n ←ContinuouslyViewedPIO(D,E, P_i)

if not the base case, consider the intermediate case of continuously perceived objects

if(E_n = null)

E_n ←ContinuouslyPerceivedPIO(D,E, P_i)

if E_n != null{

eSet ←eSet + E_n

P ←P - P_i}}

Every remaining object could not be identified using a base case or intermediate case

For each remaining object, make a new entity and reason if that entity is the

same as something seen before.

for each (P_i ∈P)

E_n ←MakeEntity(D, P_i, T)

for each(E_m ∈E){

sDesc ←IsSame(E_m,E_n,Loc(E_m),P_i,Time(E_m),T)

If the agent decides that the current object E_n is the same as the previously

seen object E_m then it should believe so and remove equivalent entities

from further consideration, both from the regular set and those considered

possible matches for entities the agent is not sure about (see below)

if sDesc = true

{Believe-equiv(E_m, E_n)

RemoveEquivs(E_n, E)

RemoveEquivsFromList(E_n, unCertain)}certainty

Otherwise if the agent is not sure, remember the pair till all the entities have been

considered.

else if sDesc = unknown

unCertian ← pair(E_n, E_m) + unCertain

eSet ←eSet + E_n}

At this point, all of the objects that the agent can identify with certainty have

been identified, now try to take can of the uncertain ones, if any.

if |unCertain| > 0

```
    DoubleCheckUnCertian(unCertain)
  return eSet
```

function IdentifyNonMovingObjects(D, P, T, E) **returns:** a set of entities

inputs:

D: description of an object in the world,

P: set of places that agent currently perceives objects that have description D,

T: the time of the perception, and

E: set of previously created entities that have description D

In this function, the agent considers those objects which cannot move themselves.

This includes immobile objects and stationary objects.

```
 if Immobile(ClassOf(D)) ⋁
 not OtherPossMover(ClassOf(D)){
   for each (Pi ∈P){
     En ←PIO-AtLoc(ClassOf(D),E,Pi)
     if(En !=null)
       eSet ←eSet + En
     else
       eSet ←eSet + MakeEntity(D, Pi, T)}
   return eSet}
```

The function IsSame checks to see if two entities represent the same object. Three return values are possible. A return value of *true* means that the agent believes that $e_1 = e_2$. A return value of *false* means that the agent believes that $e_1 \neq e_2$. A return value of *unknown* means that the agent does not believe it has enough information to decide whether $e_1 = e_2$. If the agent must make a decision with an answer of unknown, it will probably have to select randomly.

function IsSame(E_1, E_2, P_1, P_2, T_1, T_2) **returns** *three-value-boolean:*

inputs:

E_1*: The previously perceived entity.*

E_2: *the currently perceived entity,*

P_1 & P_2: *the place where* E_1, E_2 *respectively were perceived,*

T_1 & T_2*: the time of the perception of* E_1, E_2 *respectively.*

First check the intermediate case of rapid perceptions. If E_1 *was seen too recently to be the same object as* E_2 *then the agent can make this decision quickly*

if JustSeen(E_1)

return false

Next the agent needs to know the speed of the entities.

rate1 $\leftarrow$ RateOf(E_1)

rate2 $\leftarrow$ RateOf(E_2)

Since the assumption is that an object's speed is constant, if the rates of speeds differ, then the objects differ.

if (rate1 != rate2)

return false

if the agent doesn't know the speed of the objects in this non-base case situation, the agent cannot make a good decision about the identity of the objects.

if rate1 = unknown

return unknown

next check to see if an object with the known speed could have traveled from its previously known position to the currently perceived position. If not, the two must be different objects.

possibleRange ← rate1 * (T_2 - T_1)

if (possibleRange < ShortestKnownPathBetween(P_1, P_2)

return false

If the agent knows a motivation of E_1 *which would prevent* E_1 *from being at the place where* E_2 *is currently perceived, then assume the two are different.*

if Disallow(MotivationsOf(E_1),P_1,P_2,T_1,T_2)

return false

If the distance that E_1 *could have traveled between sightings is greater than some environment-specific large distance then the agent can't decide*

if (possibleRange > LargeD)

return unknown

else

Otherwise use a rule for identifying objects that might have only traveled a small distance.

return IsSameSmallD(E_1,E_2,P_2,T_2)

function IsSameSmallD(E_1, E_2, P_2, T_2) **returns** *three-value-boolean:*

inputs: *Two entities, the place where the second was perceived, and the time of the second perception.*

If the agent knows about PIOs other than the two entities in question then if none of the others could reach the place of the second perception, P_2*, by the time* T_2 *then assume that* $E_1 = E_2$

if OtherKnownPIOs(E_1, E_2)

if not (other-known-PIO) suchthat CouldReach(other-known-PIO, P_2, T_2)

return true

Otherwise if the object represented by the first entity was headed toward the place of the second perception and there are no other PIOs known to be headed in that *location* $E_1 = E_2$

if HeadedToward(E_1, P_2)

if not (other-known-PIO) suchthat HeadedToward(other-known-PIO, P_2)

return true

if there was another PIO headed in the same location, then the agent isn't sure.

else return unknown

If none of the above, there is no evidence that the two entities represent the same object and there has only been enough time to travel a short distance so assume that the two are not the same.

return false}

function DoubleCheckUnCertian(uPairs) **effects:** *possibly results in a belief that the first and second items of one or more of the pair in the uPairs list are coextentional*

inputs: *uPairs: A list of pairs of entities that the agent believes might be the coreferential. The first item in the pair is the entity corresponding to the object currently seen; the second item in the pair is the entity that might be coreferential*

First, if there is only one current object that the agent is unsure about, then the agent should just assume that the uncertain object must be the same as the one seen before (A closed world assumption if the agent gets into an unsure situation.)

if |uPairs| = 1

pair ← first(uPairs)

Believe-equiv(first(pair), second(pair))

return

At this point there should be more than one pair in the list. Either there is *one currently perceived object more than one possible match as a*

previously seen coreferential entity, or there is more than one currently

perceived unknown object

while(|uPairs| $\geq$1)

so for each currently perceived entity, consider the following

oPairs $\leftarrow$ AllPairsWithFirst(uPairs)

Now the agent must consider, does it really have to identify this object now. If not,

Then make a tentative belief so that reasoning can consider this paring again later.

if ! MustIdentifyNow(first(oPairs))

BelievePossiblyEquiv(oPairs)

if the agent has to identify the object, first consider which of the previous entities'

expected position was closest to the current entities' position

bestMatch $\leftarrow$ClosestToActualPosition(oPairs)

if there was no way for the agent to decide based on position

if bestMatch = null

then choose randomly.

bestMatch $\leftarrow$RandomSecond(oPairs)

Believe-equiv(first(oPairs), bestMatch)

The above algorithm is for identification at a single time. After the agent executes the recognize function and any of the other identification functions above that recognize might call, there is still a possibility that the agent will not have a firm belief about the identity of the object. In particular, the agent may believe that an object that it sees *might* be the same as one that it has seen before, but the agent is not sure. The agent will have a “Possibly Equiv” relation about the two entities. At this point, if it is important to the agent to know if the two entities really do represent the same real world object, then the agent should use its reasoning and mobility to check. For example, the agent may want to use its mobility to make a quick turn and use the rapid perceptions intermediate case. The agent could use its mobility and reasoning to decide how to best use a base or intermediate case to help make its decision about the equivalence of the two entities. Once the agent has reasoned about where to go and gone there, it will identify any objects it finds there using the same recognition and identification routines listed above.

When looking at the algorithm above, one notes that it is essentially an polynomial n*m algorithm (Or P*E if referring to the parameters of Recognize, the function that serves as an entry into the algorithm.) I would argue that this is a good thing, since it enforces the notion that only a limited number of objects can be effectively recognized at a time. This algorithm is intended to be cognitively based, and many theories of human cognition show that people have limited capability for simultaneous processing. Miller (1956) first introduced the idea that people could only handle about seven "chunks" of information at a time. Since then others have found similar results, including Pylyshyn (1989) who found, when testing his FINST theory, that people could track a similar maximum number of moving objects at any one time. Because of our own cognitive limits, the limits on the number of objects that this algorithm can realistically handle are, I believe, quite acceptable.

Chapter 5

Crystal Cassie: a 3-D Simulated Robot that Identifies PIOs.

5.1 Introduction

An implemented embodied artificial agent is needed to test the computational model described in chapter 4. This chapter describes such an agent, a successor to earlier embodied artificial agents built by the SNePS research group (SNeRG) at the University at Buffalo.

SNeRG has built several implementations of an embodied computational cognitive agent called Cassie (Shapiro and Rapaport, 1987; Shapiro, 1998; Shapiro et al., 2000; Shapiro and Ismail, 2001; Santore and Shapiro, 2003), based on the Grounded Layered Architecture with Integrated Reasoning (GLAIR) (Hexmoor et al., 1993; Hexmoor and Shapiro, 1997; Shapiro and Ismail, 2003). There has been one major hardware implementation of Cassie, using a commercial Nomad robot, and several simulated versions using various graphical user interfaces for her environment and to display her behavior to human observers. In this chapter, I describe a new implementation, in which Cassie's body and the world are simulated in Crystal Space (Tyberghein et al., 2002), an environment for building 3-D games. In §5.2 the architecture being used is briefly described. §5.3, is a discussion of the implementation of the architecture and how the pieces of the simulation fit together. In §5.5 the implementation of the PIO identification routines in this Crystal Space-based robot simulation is discussed. And in §5.4 I discuss the experience of using Crystal Space as an off-the-shelf 3-D gaming environment.

5.2 The Architecture

GLAIR is a three-layer architecture for cognitive robotics and modeling cognitive agents. It consists of a Knowledge Level, a Perceptuo-Motor Level, and a Sensory-Actuator Level.

The Knowledge Level (KL) is the "conscious" level of the cognitive agent. It is the location of symbols accessible to reasoning and to natural language interaction, including the "abstract-level representations of objects" discussed in (Coradeschi and Saffiotti, 2001a,b; Shapiro and Ismail, 2003). That is, the KL is the location of the concepts that the cognitive agent has, including its concepts of objects in the world.

The Perceptuo-Motor Level (PML) is the level of the cognitive agent's representation of the physical properties of objects. At this level, objects are represented by N-tuples of their physical characteristics such as shape, material, and size (see §5.3.2 for details of the characteristics in this simulation) rather than by their KL concepts. At the PML these physical properties are associated or aligned with the agent's KL concepts of the object with those properties. The PML is also the location of the cognitive agent's well-defined skills, including natural language understanding and generation and the primitive actions of the KL—those skills which do not require conscious reasoning from the agent. The algorithm described in §4.6.3 is implemented at the PML; as is often the case, though, these routines often invoke the KL to reason consciously about some part of the identification problem. In practice, the PML is often separated into three parts (Shapiro, 1998; Shapiro and Ismail, 2003), referred to as the PMLa, PMLb, and PMLc. Details of the implementation are to be found in §5.3, below.

The Sensory-Actuator Level (SAL) is where the low-level control of the cognitive agent's effector and sensors (real or simulated) is located.

5.3 Crystal Cassie: A Cognitive Agent Implementation

5.3.1 The Four Processes of Crystal Cassie

The Crystal Space version of Cassie (Crystal Cassie) is composed of four separate processes. Two of the processes explicitly implement parts of the GLAIR architecture, a third is used for natural language interaction with a human user, and the fourth contains the simulation of the agent's physical body and the world itself. The processes are connected using standard IP sockets. Each of these processes is described in more detail below. Figure 5.1 shows the four processes, their socket connections, and the parts of the GLAIR architecture that they implement. The figure also shows the interface between Crystal Cassie and a human user/interlocutor/observer.

Process P1 implements the KL, and the top two parts of the PML. It is is written in Allegro Common Lisp. The KL is identical to the FEVAHR knowledge level described by Shapiro and Ismail (2003) but with different domain knowledge. It is implemented using the SNePS knowledge representation and reasoning

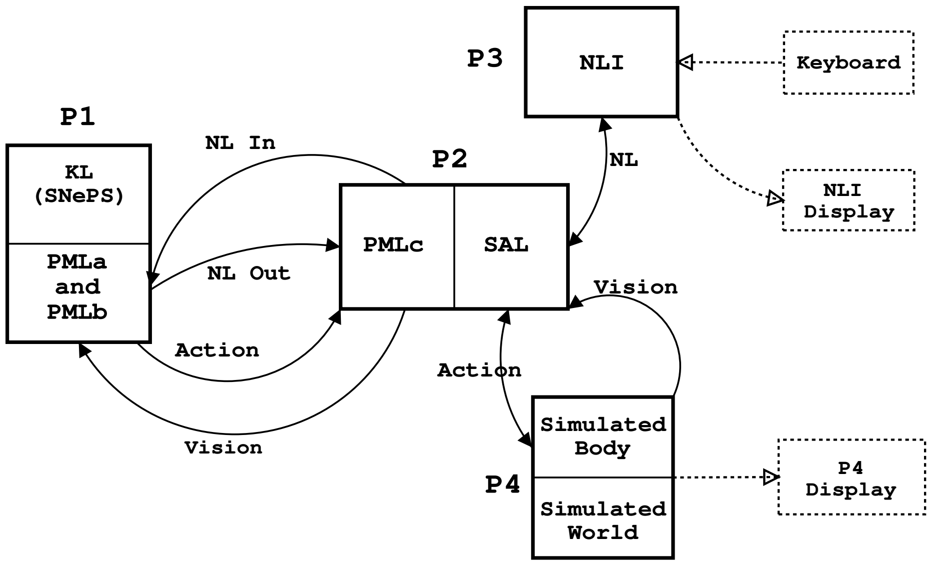

Figure 5.1: A schematic of the four processes, showing how they communicate with each other and with the user, and how the various parts of the GLAIR architecture are distributed among them.
The solid boxes represent the four processes, labeled P1 through P4. The solid lines represent socket connections with communication flowing in the direction of the arrows. The boxes represent a user interaction modality. The dotted lines represent either input from or output to the user depending on the direction of the lines.

system (Shapiro and Rapaport, 1987, 1992; Shapiro and the SNePS Implementation Group, 2004). Symbols in the KL are implemented as terms in the SNePS logic (Shapiro, 1993, 2000).

Process P1 also contains parts of the PML. The top sub-level of the PML (the PMLa) is where the agent's well-defined skills (KL primitive acts, including the PIO recognition routines) are implemented. It is also the place where natural language understanding and generation occurs. Natural language interaction is implemented using a GATN[1] grammar (Shapiro, 1982, 1989).

The second sub-level of the PML (the PMLb) implements the connection between the PMLa and the rest of the simulated agent. In previous simulations, (Shapiro, 1998; Shapiro and Ismail, 2003), the connection was made using the Lisp foreign function interface. In Crystal Cassie, the connections are made using IP socket connections. The PMLb has four sockets used as one way connections to the PMLc (described below). These connections each represent one of the agent's cognitive modalities (Shapiro and Ismail, 2003). The connections are described next from the point of view of the PMLb.

[1]Generalized Augmented Transition Network

The first connection is a one-way natural language *(NL) in*put connection (which can be thought of as a "hearing" modality, for receiving the natural language that the sensors received at the SAL layer), over which sentences are sent as strings. These strings are then sent to the GATN parser at the PMLa level, which translates them into the SNePS KR language.

The second connection is a one-way *NL out*put connection (which can be thought of as the agent's "speech" modality) from the PMLb to the PMLc. Sentences and phrases from the GATN generator are sent through this connection.

The *Action* connection (the movement modality) is used to send action requests from the PMLb to the PMLc/SAL layer. It is again used as a one-way connection.

The *Vision* connection (the vision modality) is a one-way connection from the PMLc to the PMLb, over which is sent a series of n-tuples with the relevant features of all of the objects that are currently visible to the agent's simulated vision system. See §5.3.3 for more about the simulated vision system.

The remaining three processes all currently use Crystal Space tools (Tyberghein et al., 2002) and are written in C++.

Process P2 implements the PMLc and the SAL. The PMLc mediates between the PMLb and the SAL. This process controls Cassie's (simulated) sensors and actuators and connects all the other processes to each other. The PMLc has the four socket connections to the PMLb discussed above. The SAL has three more socket connections to the remaining processes. Each of these connections and its function is described below in the paragraph devoted to the process it is connected to.

The Natural Language Interface (NLI) process (P3) uses the Crystal Space console window as a mechanism that the human interacting with Cassie uses to type natural language text to the agent and to read Cassie's responses. This process connects to the SAL via a socket connection, used as a two-way connection, using the Crystal Space C++ socket wrappers. The NLI sends one sentence at a time, as a single line, to the SAL, and listens to its network connection for replies. It prints the full text of any text it receives prepended with the string "Cassie replies:" to its display. Figure 5.2 shows the NLI display.

The fourth process, (P4), implements the simulation of Cassie's body and the simulation of the environ-

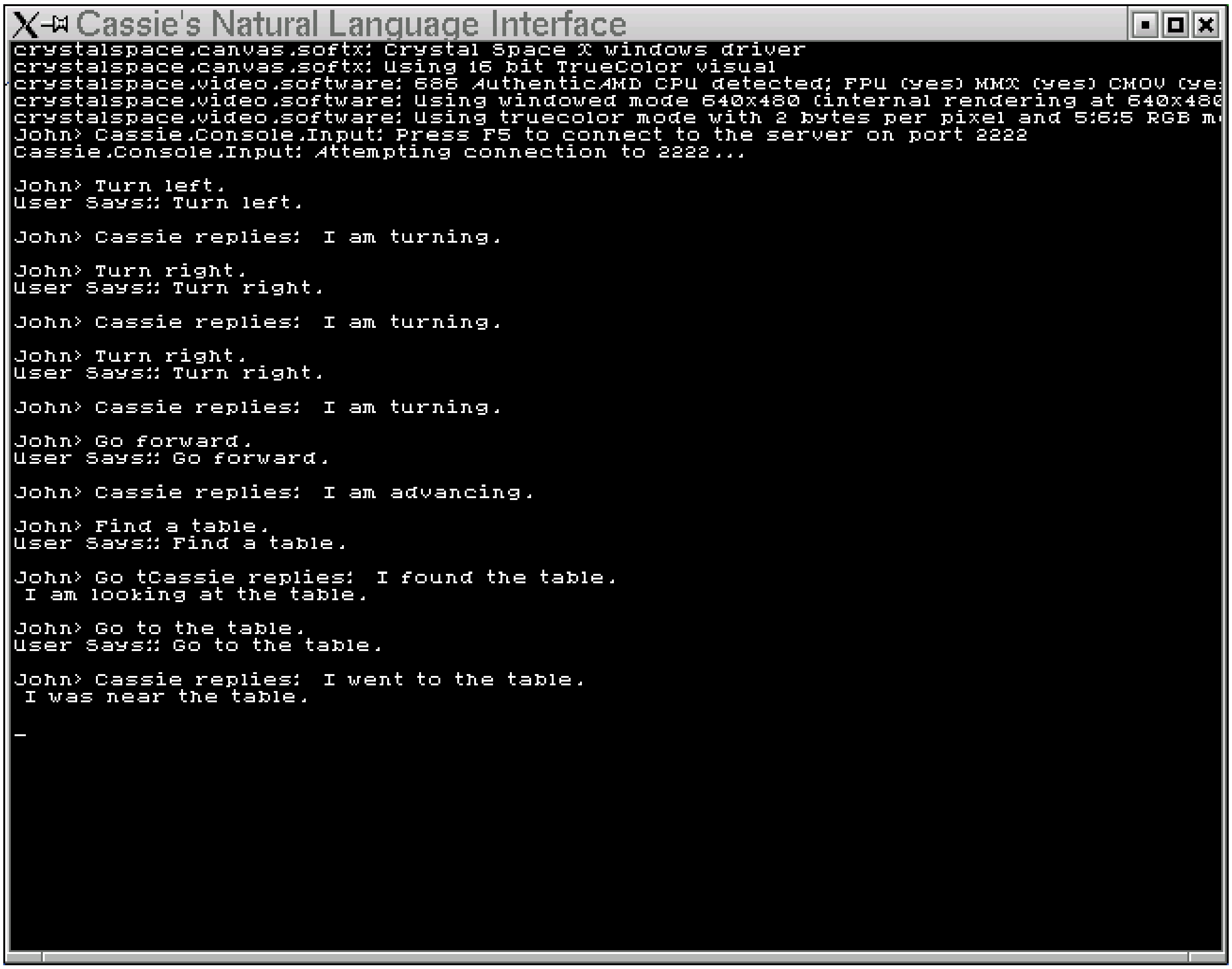

Figure 5.2: The user's view of the NLI process during a sample interaction. The 'John>' prompt is a prompt printed by the Crystal Space terminal library before user input.

ment that Cassie interacts with. P4 displays what Cassie can see, showing the results of her interactions with the environment. See Figures 5.3 and 5.4 for views of what P4's display looks like. The simulated body connects to the SAL with two socket connections. The Action Connection is a two-way connection. The simulated robot body receives action requests (requests to turn motors on or off, forward or backward) from the SAL. Cassie's body will then act on those requests. These actions may or may not produce a change in the world. (E.g. asking Cassie to move forward when she is up against a table will not accomplish anything.) In order to better model actual robots, there is some uncertainty built into the simulated robot. When the simulated body receives action requests, the actual distance traveled may or may not be the amount requested by the higher levels. When an action is finished (with either success or failure) the simulated body sends the SAL a short "action complete" message over the Action Connection. The success or failure of the action request must be determined by sensing just as in a real robot; no "cheating" information is sent back through the action socket.

The Vision Connection is a one-way socket connection from the simulated robot body to the SAL. The body sends vision information in the form of a string consisting of a set of space-deliminated 4-tuple of {material, shape, location, size} values. The SAL translates this simulated sensor data into the "physical-level representations" (Coradeschi and Saffiotti, 2001a,b; Shapiro and Ismail, 2003) appropriate to the PML. See §5.3.3 for more on the simulated vision system.

Cassie's simulated vision system can only see and send visual information about objects which appear within her first person perspective of the world. See Figures 5.3 and 5.4 for samples of what Cassie can see. Cassie's PML only receives feature tuples for those objects within her visual field, that is, those shown on P4's display. P4's display is alike in every way to the display of an equivalent scene in the human-participants' test application described in Chapter 3.

Together, these four processes implement a working cognitive agent based on the GLAIR robotic architecture.

Figure 5.3: The P4 display showing a sample view of what Cassie can see: a computer lab with two people in it.

5.3.2 Crystal Cassie's Environments

Crystal Space environments are defined in a Crystal Space-specific XML (text) file format. The file defines the geometry of all of the objects in the environment, including the rooms, and materials associated with those objects.

Objects are first defined by a series of vertices given by <x, y, z> world-space coordinates; the x, y and z values of the coordinates are each floating point numbers. These vertices are then used to define polygons. In Crystal Space, as an optimization, only one side of a polygon is visible: the visible side is the one defined by the list of vertices in clockwise order. Polygons are then grouped together to form objects—either rooms or objects in the rooms.

In order to be visible, each polygon must also have a material associated with it. Materials are images of the visible features of objects in the world. For example, in Figure 5.3, "wood" is a material on the table object. A material is generated from an image file in a Crystal Space data library . Most standard image file formats are supported. Crystal Space data libraries are implemented as zip archives referenced in a Crystal Space-specific configuration file. The height and width of material images must be some power of two, though the height and width need not be equal. Crystal Space will then tile the material across the associated polygons. The syntax allows materials to be scaled before being tiled on the polygon, so that users can create better looking worlds.

Crystal Space calls objects in rooms meshes. Meshes are prototyped once as a group of vertices, polygons, and associated materials, and can then be used in any or all rooms defined in the file. Meshes and rooms can both have "keys" associated with them. These keys are ignored by the Crystal Space engine, but can be queried by user programs. This way, objects in the world can have customized user data associated with them. Each object in our environments has two keys associated with it, one for shape and one for material. We use these keys in our simulated vision routine discussed in §5.3.3.

Most commercial game engines have a graphical "world editor" for users to design their environments. Crystal Space does not yet have a native world editor, although the Crystal Space project has supplied several programs to convert environment files generated by some commercial world editors to Crystal Space format.

At the time that I designed and constructed the environment, these tools did not make use of many of the Crystal Space features, so the environment files were built using a text editor.

I did use the automated tools in one instance. The very complex shapes of the robots and people in the world (the same shapes used in the human participant experiment; see Figures 5.3, 5.4, 3.6, and 3.7) were generated from public domain and freely available Quake II model files. These meshes are each composed of several hundred triangles. The Crystal Space conversion tools were used to convert the Quake II format to Crystal Space format and then to merge the resulting file into the environment definition file.

I set up two different environments to work with the Crystal Cassie simulation. The first is the small suite of four interconnected rooms used in the robot-counting task in the human-participant experiment (see §3.3.6) whose floor plan can be seen in Figure 3.3. The second environment is the much larger suite of rooms used in the following tasks in the human participant experiment, whose floor plan can be seen in Figure 3.8.

The rooms in the smaller suite contain tables, chairs, drinking glasses, bottles, and sometimes robots. The larger suite is intended to resemble the wing of an academic building. It includes: two classrooms with chairs, tables, and whiteboards; a computer lab with chairs, tables, computers, monitors, and keyboards; a lounge with a stove, table, and soft drink machine; and a second lab with bulletin boards, a table with a large machine on it, and a filing cabinet. The larger suite is connected to a parking garage with a single car in it. Both suites can also have people or robots wandering through them. Figure 5.3 shows a view of the computer lab in the larger suite, containing some chairs, tables, computers and monitors, and two people. Figure 5.4 shows a view of a room in the smaller suite, containing a table with a bottle and several glasses on it, and two robots roaming about.

5.3.3 Simulated Vision.

Processes P1, P2, and P4 of Crystal Cassie all play parts in the simulated vision system. Vision information is only sent by the simulated robot body when it receives a “look” action request. Cassie begins the simulation not looking at anything, and will only consciously look at the world if her reasoning at the KL triggers an action that requires looking. When she does the PML act of looking, the request is sent to the SAL, which asks the robotic body to look at the world and inform it of the objects that are currently visible.

At the level of the simulated robot body, the objects visible to Cassie are those also visible to the user viewing P4's display(see Figures 5.3 & 5.4). This is, again, the same view that the human participants saw during the experiment. One exception to Cassie "seeing" everything on the P4 display is that because of a peculiarity in how Crystal Space marks things as visible, Cassie cannot see through doors (doorways) into adjacent rooms. This limitation is discussed further below. Cassie has a viewing angle of about 60 to 70 degrees in a single direction. The view is similar to what is visible in most commercial immersive 3-D computer games.

The look act, at the level of the simulated robot body, is implemented as a P4 method which is called from the Crystal Space libraries via the C++ callback mechanism. When requested, Crystal Space's rendering algorithm will call this method for every object that Crystal Space marks as visible. This includes both the meshes that are visible, and the current room itself. For each object, the shape and material keys (see §5.3.2) are extracted. The Crystal Cassie simulation uses the key value for material rather than the name of the actual material for two reasons. The first reason is a limitation of the Crystal Space API in the version we are using. The name of the material is not available from the polygon object associated with it. The second reason for using a material key (should the first issue be resolved) is a simplification of the simulated vision. Some objects, such as the glasses, table, and bottles in Figure 5.4, have a single material on all of their polygons. However, other objects, such as the computer monitors in Figure 5.3 have multiple different materials; some polygons have one material, and other polygons have another. In order to simplify the simulated vision algorithm, I chose the material that I felt was most salient as the material key value. This eliminates the problem of recognizing an object from different viewing angles.

In addition to the shape and material keys, the vision callback also retrieves the object's current center point in world coordinates and calculates the radius of the object. These data points form the 4-tuple {material, shape, location, size} for the current object. When the callback has finished with all of the currently visible objects, the 4-tuples of those objects are sent to the SAL in P2. The simulated sensor controls in the SAL currently send the 4-tuples from the vision system unchanged to the PML.

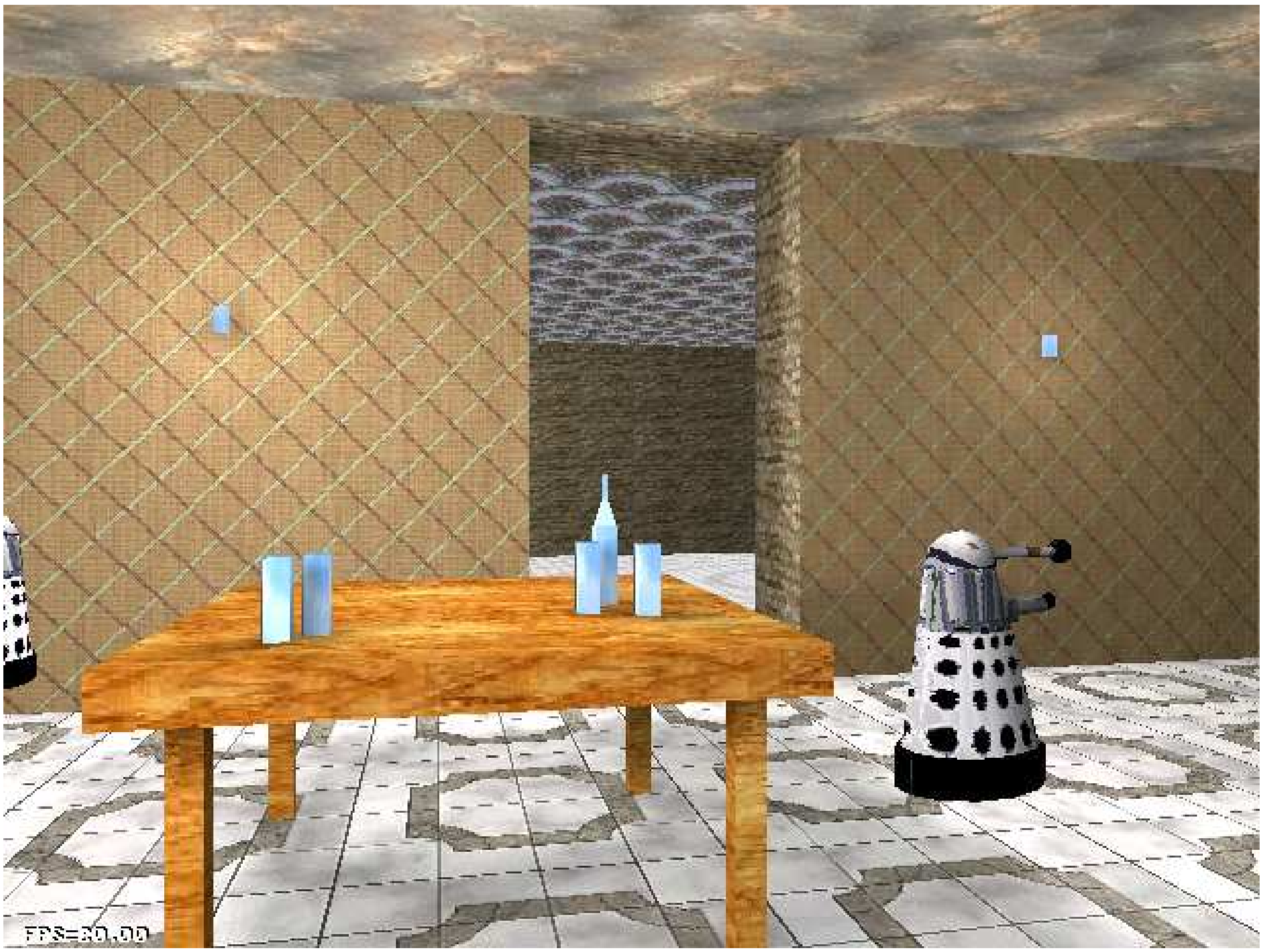

Figure 5.4: A view of the smaller environment from the P4 display, showing a room containing two robots and a table, on which is a bottle and several glasses.

At the PMLa in P1, a vision tuple is "parsed" to see what kind of thing it is. In the PML there is a data structure containing a list of "alignments" between {material, shape} pairs and Cassie's concepts of the categories of things she knows about (for example, the category of wooden tables). Once this kind of basic object recognition is done at the PML, the KL must be invoked to reason about exactly which specific object Cassie is looking at. PML routines start the identification process and invoke KL reasoning in the process of identifying objects. The identification process uses the location information from the 4-tuple to reason about which object Cassie is looking at. It is impossible to distinguish between many of the objects in the environment using only their shape and material; they are perceptually indistinguishable to Crystal Cassie. The glasses in Figure 5.4 cannot be distinguished with only shape and material information, nor can the robots in the same figure. However, Cassie can reason about the identity of the glass she is looking at using the PIO identification model and reasoning about its location (since glasses are stationary objects).

The Crystal Space team recently released a new version of the Crystal Space graphics tools. The original callback method that we used for doing simulated vision was removed in this new version. The original version used to return a list of objects visible in the current room only. The replacement mechanism that I am using now, will return all those objects marked as "probably visible" by the system. Objects marked probably visible are all those with at least part of their bounding box within the view frustum (a term in the graphics field for the mathematical calculations of the visible area) of the P4 and either in the current room or in a nearby room. Unfortunately, this can include objects that are completely occluded by the walls separating adjacent rooms. As a simple way of correcting this, we cull out all meshes that are not in the current room. However doors and the adjacent rooms themselves are still visible. In order to see objects in an adjacent room, Cassie will have to find a doorway and go through it.

In previous GLAIR embodied agents, simulated vision was limited to viewing a single object at a time. It was necessary to eliminate this restriction in the Crystal Cassie simulation in order to support the "simultaneous perceptions" base case (see §4.3.2). Crystal Cassie now "sees" all of the objects that are within her visual field when she does a look act. She identifies all of the objects of a particular type all at once. Through the interaction of the PML and the KL, each object that Crystal Cassie sees is identified independently using the PIO *recognize* routine described beginning on page 101. Cassie identifies objects either as something that she has already seen, in which case she reuses the existing KL concept, or as a new object, in which case she conceives of a new object of the correct type in the perceived location. After identifying all of the objects that she is looking at, Cassie then forms the belief that she is looking at the set of all these objects.

5.4 Experiences using Crystal Space

When I decided to use an existing 3-D graphics/gaming engine for the next simulated robot to be built on the GLAIR architecture, I needed a tool with a published API. I also wanted, if possible, an engine with publicly available source code, so I could build the engine for different operating systems. There were two possibilities: using a commercial game engine that had released the source code for its game under a public license, or using a publicly available engine built by a group of hobbyists for their own enjoyment.

At the time, id Software[2] had released the software for their original Quake game engine; they have since released the software for some of their newer Quake engines as well. The benefit of using this type of engine was that the API would be well defined and fixed. The drawback was that the company was no longer supporting the product, so any bugs that were found or any new features needed would have to be taken care of by the end user.

The 3-D engines produced by hobbyists had a complementary set of benefits and drawbacks. Bugs are always being fixed, and new features are being added. However, this can make the API quite unstable. On the other hand, the original designers are available to answer questions and provide support.

Given these considerations, and the tools available when this project started, I chose the Crystal Space engine, which is a hobbyist project. Because Crystal Space development is ongoing, I have had to adjust the code for the Crystal Cassie with every release to reflect API changes. The Crystal Space developers include a list of most of the necessary changes with each release. Some releases have required many fewer changes, but most require a large number of changes to adjust for an updated API. The community of hobbyists that design and use the Crystal Space engine supports projects built using Crystal Space tools by quickly answering most

[2] http://www.idsoftware.com/

questions. The chances are fairly good that someone will know which of the hundreds of C++ classes that compose Crystal Space will be the best to use for a particular need.

I have used Crystal Space without modification as a set of shared libraries, on both Linux and Solaris platforms as an off-the-shelf 3-D engine solution to meet robot simulation needs. It provides enough flexibility in building environments to allow a reasonably complex environment for our agent to explore.

Crystal Space has performed adequately as a library for developing a simulated cognitive agent. Crystal Space still has a few holes in its API, and the Crystal Space designers are working to fix them, and are committed to doing so. Overall I have been able to use Crystal Space successfully, though each Crystal Space "stable release" sometimes requires a significant modification of the Crystal Cassie simulation.

5.5 The PIO identification routines in Crystal Cassie

5.5.1 Implementation.

Currently, Crystal Cassie has a small repertoire of primitive acts which she can use to build more complex behavior by reasoning and planning at the KL. At the most basic, Cassie can be told to turn left or right, or to go forward or backward. In this way she can be tele-operated through natural language. Cassie also has a primitive act for looking, which is used to invoke the simulated vision system (see §5.3.3).

Finding an object that is in the current room is also a primitive act. Cassie will look, turn, and then look again, repeating this sequence until she either finds the object or completes a 360-degree rotation. If she cannot find the object in this room, she must reason at the KL about how to look for it in another room.

Counting objects is a primitive act. Cassie will look around for any objects with the correct properties within the current room, identify them, and then think about all of the objects that she has ever seen and report the total count. Cassie is currently restricted to counting within a single room at a time because she has no concept of a map, and therefore wouldn't know when to stop moving from one room to the next if she tried to search through all of the rooms.

Cassie's next primitive act is to go to an object. In order to go to an object she has to be looking at it, which means that she has found it. When going to an object, Cassie uses the location and size values in the vision 4-tuple (see §5.3.3.]) of the PML description of the object. The location is the object's center of gravity. The size is the object's largest radius in the <x,z> plane. When Cassie goes to an object she plots a straight-line course from her current position to a position close to the object. This new position is calculated using the location value for the object and the sum of the size value and a small constant. Cassie will then move along this straight line until she reaches the new point.

Cassie's next primitive act is to go through a room or doorway. When she goes through the room or doorway, she focuses on the center and then moves from where she is through the center point and stops beyond the center at a point four times her own radius through the center point.

Like previous versions of Cassie, following a moving object is a primitive act in Crystal Cassie. However, in Crystal Cassie, in order to follow a moving object, Cassie must perform a series of finding, recognizing and going-to acts. Cassie searches her visual field for objects of the correct type, recognizes the correct object, and then goes near where she sees the object. Finally she repeats the whole process again until told to stop, or until she loses the followed object.

The five functions that describe the computational model of identifying PIOs described in §4.6.3 have been implemented in the Crystal Cassie simulation. These PIO identification functions call on the helper functions described in §4.6.2. Many of these helper functions are trivially implemented as calls to built-in SNePS functions; some others were also relatively easy to implement. However, some of the helper functions are more involved. In particular, those helper functions that are designed to create beliefs from sensor data are not well defined. Some helpers, like *MotivationsOf,* rely on beliefs created from the analysis of sensor data. The processing of sensor data to understand the motivation of another agent are the focus of significant ongoing open research in the areas of plan recognition (Shapiro et al., 1989; Han and Veloso, 2000; Kaminka and Avrahami, 2004) and emotion recognition (Nakatsu et al., 1999; Pardas et al., 2002; McGlaun et al., 2002). A complete model of forming beliefs from sensor data is beyond the scope of this dissertation. However, each such helper function is implemented so that it initiates KL reasoning to determine if the appropriate beliefs have already been formed. If the beliefs exist, the PIO identification routines will use those beliefs as described in chapter 4.

The robot simulation has a working implementation of the PIO identification routines for the first three base cases (simultaneous perceptions, objects with a unique appearance, and immobile objects) and for the first three intermediate cases (rapid perceptions, locally unique appearance, and stationary objects). The base and intermediate cases of continuously viewed and continually perceived objects require a computational implementation of target tracking (Papanikolopoulos et al., 1993) or of FINSTS (Pylyshyn, 1989). The general case of identifying PIOs is also implemented in Crystal Cassie.

Cassie, in this simulation, can correctly identify objects using these base cases in all tests done in the simple four-room suite seen in figure 3.3. Human participants were also very good at identifying objects when they were able to use a base or intermediate case correctly. Cassie has all of the background knowledge (at the KL) that she needs in order to apply the base and intermediate cases in the correct situations, and to not apply those cases in the wrong situations.

Crystal Cassie's natural language interaction is handled using a GATN grammar. The GATN used is the Fevahr Cassie grammar (Shapiro and Ismail, 2003) overlayed with modifications to handle time as described by Ismail (2001). The modifications to handle time add additional network structure to every belief about an action. Though a personal sense of time is clearly needed, the resulting generation grammar is somewhat brittle and occasionally produces a semi gibberish line as Cassie searches for how to say something and then vocalizes all her thoughts. For example in figure 5.5, Cassie does this when she says that she is counting. She refers to herself as 'I', 'Cassie' and the robot, and reports that she is counting in each case. She also refers to counting both as 'counting', the lexeme that she has for counting, and as 'm11', her intensional concept of counting. Even though the natural language generation is somewhat brittle, the natural language input is much more complete, accepting the full input set from both the FEVAHR grammar and the time grammar.

5.5.2 Crystal Cassie in action.

This section of the chapter shows Crystal Cassie performing tasks similar to those performed by the human subjects, showing that the PIO identification algorithm works in a variety of situations. Crystal Cassie is a completely graphical simulation, unlike some earlier simulations. The best way to show Cassie in action, would be to show it as a movie. Unfortunately, at the time of this writing I do not have access to the hardware necessary to capture the video for a movie. I will instead insert several still images showing Crystal Cassie performing some typical actions.

The first action to be demonstrated is counting. Cassie counts the objects in her suite of rooms just like the people in the experiment did. She begins without knowledge of any of the object (with the exception of herself as a robot, see below), and then, each time she encounters an object that she believes is new, she conceives of a new entity. When she believes that she has finished surveying the room, Cassie determines how many objects there are by thinking about all of the objects that she knows about (and disregarding extra coreferential entities). She then reports the count of all of the objects that she has identified.

Like the human participants who were sure of the objects they had identified and the counts of those objects, Cassie will not reset her count to zero until she is placed in a new world to count from scratch again. If she makes an initial count, and then finds new objects, she will add these new objects to the count, but not start again from scratch since she is in the same suite of rooms.

Figure 5.5 shows the natural language interactions with Cassie as the user asks her to count the glasses. She counts the glasses in the room. Figure 5.6 and Figure 5.7 show two of Cassie's views of the world during this counting task. Cassie sees the same table with the same two glasses and bottle on it from two different angles as she turns around counting. However, she correctly identifies the pairs of glasses as two sightings of the same objects. Later, the user tells Cassie to turn several times and then to advance. This steers her into an adjacent room. (Cassie is not using her vision system at this time, since turning and moving forward do not require it.) Once in the new room, the user again asks Cassie to count the glasses. Cassie again looks around the room for glasses, sees the glass on the chair (see Figure 5.8) and identifies it correctly. Now she has identified three distinct glasses, so she replies that she counted three of them.

Counting glasses requires Cassie to identify those glasses. The identification is an instance of the *stationary objects* base case. Recall that the base cases always give the correct answer so long as the knowledge that the agent uses to invoke the base case is correct. Human participants who counted glasses incorrectly

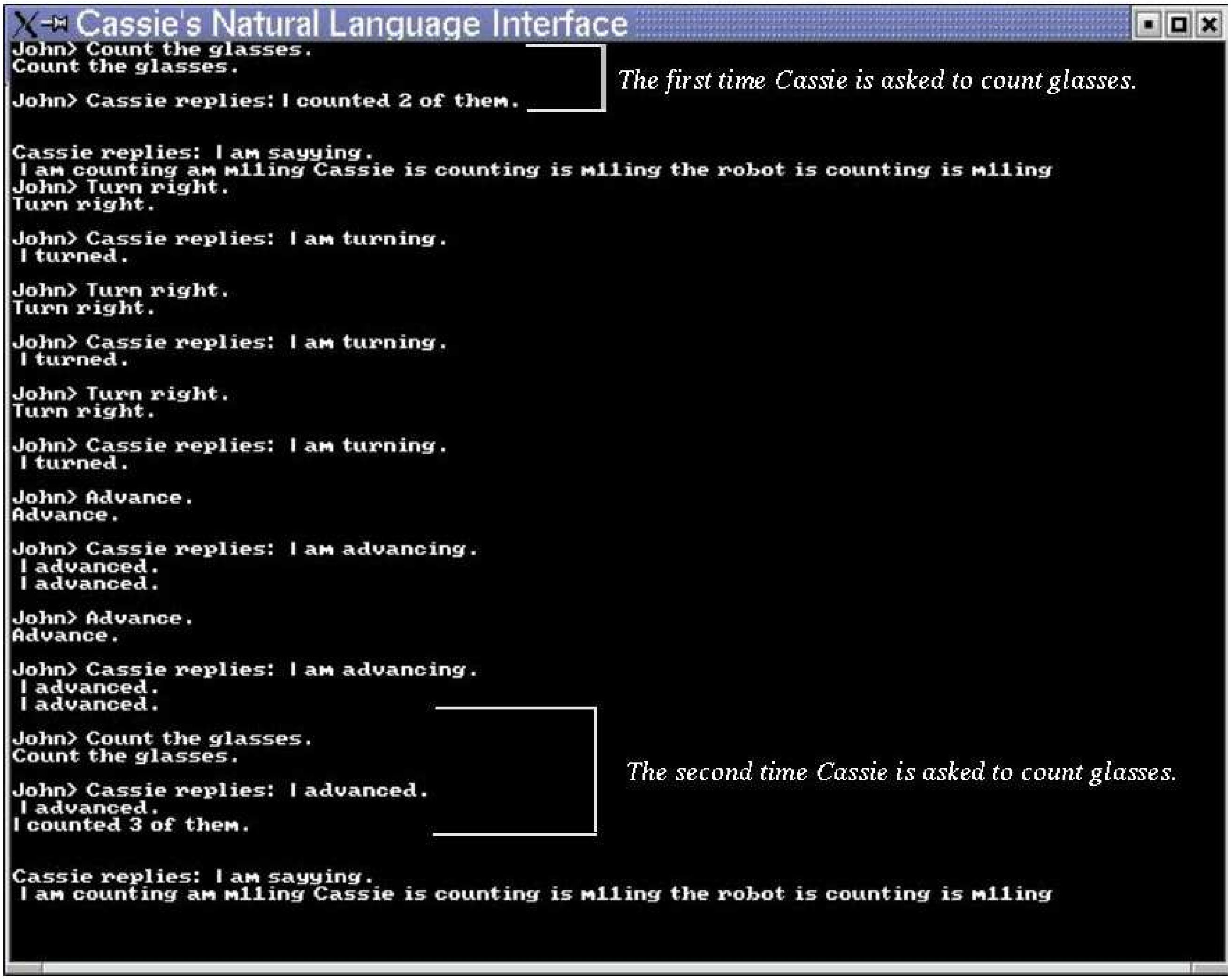

Figure 5.5: The NLI showing the results of asking Cassie to count glasses; steering her into another room using natural language and then asking her to count again. Note that there is only one glass in the second room (and two in the first room).

Figure 5.6: Cassie beginning to count glasses in the first room.

Figure 5.7: The second time Cassie looks at the same glasses originally seen in Figure 5.6 during the counting of glasses.

Figure 5.8: Cassie looking at a glass sitting on a chair while counting glasses in the second room.

Reason	Number of Participants
Mis-adding correctly identified objects	1
Carelessness in observing objects	2
Perceptual aliasing	2

Table 5.1: Reasons for the five human participant failures in the counting glasses task.

were either careless and didn't look at all of the glasses, or suffered from perceptual aliasing. See Table 5.1 for the numbers of each type of failure. Cassie is very methodical and tries to view the entire room. (There is a small amount of uncertainty built into the simulation to give a greater realism, so she doesn't always see all 360 degrees.) Cassie's location system is built on grid coordinates, so she is not vulnerable to perceptual aliasing. Given these conditions, Cassie should count glasses correctly every time unless she simply doesn't see one due to the simulated robot slippage error. Cassie does count glasses correctly each time.

Counting mobile objects is more difficult than counting stationary objects. Identifying stationary objects, where there is no other agent as a mover is an intermediate case and is quite easy for agents able to avoid perceptual aliasing. Counting mobile objects often means identifying those objects using the general case and considerable reasoning. Figure 5.9 shows the NLI during one successful run of Cassie counting robots. She is asked, at the beginning of the run, to enter a new room where there were two robots. See Figure 5.10

Reason	Number of Participants.
Robots moved and were never seen by participant. (undercount)	1
Participant perceptually aliased two rooms and mis-identified the robots in those rooms. (undercount)	3
Misidentification in the general case through a compression error. (undercount)	4
Misidentification in the general case through a dispersion error. (overcount)	1
Unclear from Transcript	2

Table 5.2: Reasons for the eleven human participant failures in counting robots. Undercount indicates that the participant(s) thought that there were fewer robots than there really were while overcount means that the participant thought there were more robots than there really were.

for a snapshot of a view Cassie had of this room. After she scans the room she reports a count of three robots (she counts herself as one of the robots.) She is then directed into the next room and asked to count robots again. There is a single robot in this room for this trial. After scanning the room and identifying that robot, she reports that she has counted four robots. The four robots are the one from the current room, the two in the previous room, and Cassie, herself.

Aggregating the data from both of the mobile object(robot)-counting tasks (the one with all silver robots and the one with both silver and gold robots), participants in the experiment achieved the correct count 68.5% of the time. Cassie, when asked to count robots in two different rooms, achieved the correct count 63.6% of the time with 11 total trials. This difference was not at all significant with a t-test (p=.77).

Human participants failed to correctly count robots for several reasons: a robot was missed entirely and never seen since the robots moved; some participants perceptually aliased two of the rooms and therefore identified some robots as being the same as others; several participants misidentified robots in the general case, both in under counting (seeing two robots and identifying them as a single robot) or in over counting (seeing one robot at different times and identifying it as two different robots). When participants misidentified objects in the general case, their *IsSame* functionality returned the wrong value (or it returned 'UNKNOWN correctly and then the participant happened to guess wrong) See Table 5.2 for the numbers of each failure type.

Cassie's failures fall into the categories of *Misidentification in the general case (undercount)*, *Robots moved and were never seen by participant (under count),* and *Misidentification in the general case (over-*

count). Cassie had a single failure wherein she simply never saw one of the robots. She also had three failures in which she misidentified robots in the general case because of the time between sightings (two overcounts and one undercount). The KL rapidly slows down as the knowledge base grows large. This means that each recognition stage takes a bit longer until recognitions take several seconds. Sometimes this gives the robots a chance to move to unexpected places while Cassie is busy thinking. The three of Cassie's failures in the general case can all be attributed to the long time in identifying objects. Some of the human participants dithered during their tasks and took a long time to look from one robot to another, but participants didn't take quite as long as Cassie. Some did, though, make the same kinds of mistakes due to the long time between sightings that she did.

At the time of this writing, Cassie can only perform the following task for the first segment. She will find the robot to be followed, go to where it is, then look for it again, and go there again. She does this until the first time the robot that she is following enters another room. Then her inability to see through doors becomes a big liability. She always loses the robot that she is following at the first door.

5.5.3 Discussion of the implementation and the ramifications of the algorithm.

The PIO identification routines discussed in this dissertation and the implementation described in the previous section have all been designed to be cognitively plausible and cognitively based. Crystal Cassie, equipped with an implemented PIO identification system, performs much like a person does in the same situations. Though Cassie is vulnerable to some difficulties unique to her status as a cognitive robot, such as slowing down as her knowledge base grows large, her mistakes still fall into the same categories as the mistakes of human participants. That is, she makes the same sorts of mistakes that a person might make in the same situation.

When Cassie performs the counting tasks, she does so at a competency level near that of people. She is not subject to perceptual aliasing because she uses grid based coordinates, and so she does not fall subject to errors in identifying objects caused by perceptual aliasing. She is very methodical so she does not fall victim to the errors that human carelessness introduced into the experiments. The combination of immunity

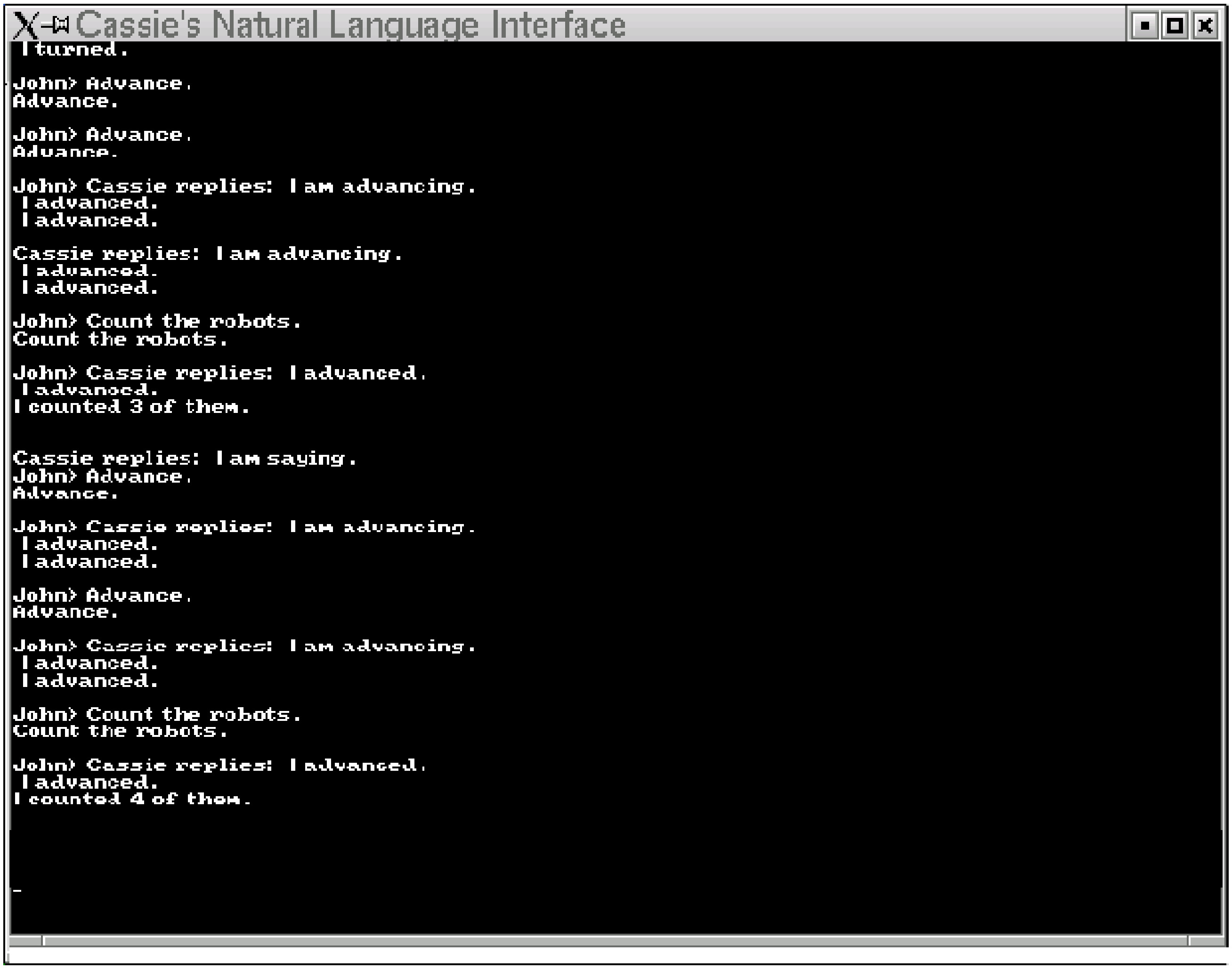

Figure 5.9: The NLI during a run of Cassie counting robots. Cassie's responses are printed by the terminal program where ever the cursor is; this sometimes leads to the response being printed right at the prompt, rather than below it as one might expect. The prompt is reprinted whenever the user types a key again.

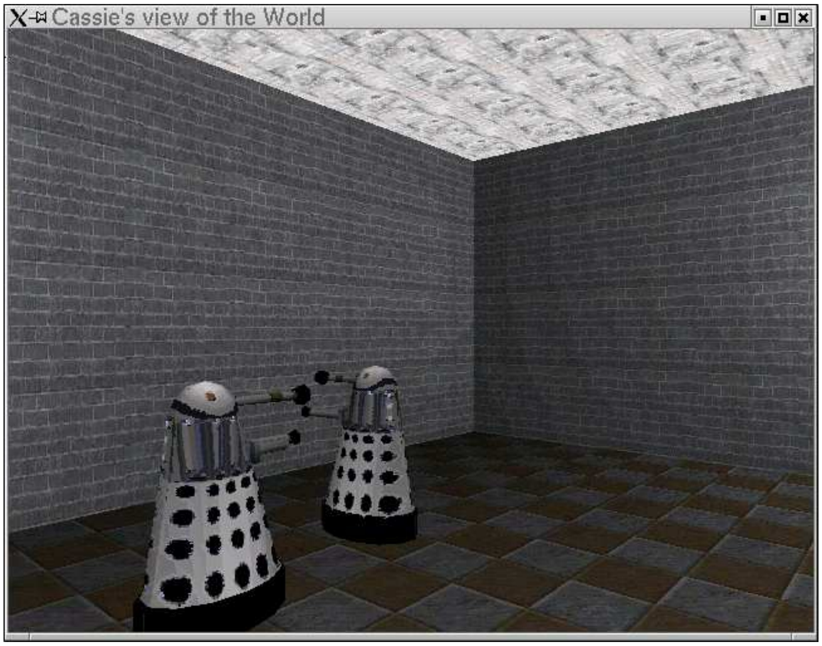

Figure 5.10: A view of the second room showing Cassie viewing both robots at once. She is able to use the base case of simultaneous perceptions in this particular run. She was not always able to use this case; in some runs the random motions of the robots took the two robots to opposite sides of the room.

to aliasing and careful method makes Cassie perform even better than people at identifying stationary PIOs.

Cassie is, however, vulnerable to some of the common problems that caused human participants to over or under count when counting moving objects. If one of the robots that she is trying to count moves in such a way that she never sees it, Cassie will of course give an undercount of the objects. Cassie is also vulnerable to the leading cause of counting failure by the human participants in my experiment: if Cassie sees two different robots at different times, but at similar points in space, she is likely to assume that they are the same robot. This is a mistake in applying the general case of PIO identification using time, location and possible travel distance. This is the mistake made by the largest set of participants who made an error in counting. See Table 5.2 for the counts of human failures in the robot counting task. The fact that Cassie is vulnerable to this mistake as well indicates that the PIO identification routines are cognitively plausible since she makes similar mistakes to those people make.

When people form beliefs and use those beliefs while acting in their interaction with the world, they do not reconsider those beliefs unless they notice some failure or discrepancy. Consider the experiment done by Simons and Levin (1998). Their experiment began with a conversation between one experimenter and an unsuspecting pedestrian. That conversation was interrupted by collaborators walking between the first experimenter and the pedestrian carrying a door. During the interruption, the first experimenter would leave with the door, and a second experimenter would remain behind and continue the conversation. In one version of this experiment, more than 65% of the pedestrians failed to notice the switch and continued the conversation with the second experimenter that they had begun with the first experimenter. Because they didn't notice a problem, they continued with the same reasoning and background knowledge and acted accordingly. Participant P42 in the human participant experiment fell victim to the same fate. She believed that there was a single robot, and didn't see anything to conflict with that belief until long after she began following the wrong robot. Cassie, equipped to identify PIOs in a manner similar to people, also fell victim to similar problems. Moreover, like people, Cassie will never reconsider her identification of an object unless she conceives that there is a problem.

Finally, with the above in mind, let me consider a problem observed in some recent robot exhibitions[3]. In the exhibition, the robot was trying to follow a focus person. It did so by tracking the person's shirt color and following the appropriate color blob. It was possible for someone with a similarly colored shirt (a person perceptually indistinguishable to the robot) to come by and "steal" the focus of the robot, whereupon the robot would follow the new person. Would such a robot equipped with the computational model described in this dissertation fall victim to the same error? The answer depends on how the distractor person entered the visual field. If the distractor person entered from the side of the visual field, such that the robot was able to sense two independent color blobs, then the robot equipped with this PIO identification algorithm would be able to notice the potential problem and identify and follow the focus person correctly. This assumes that the focus person and the distractor person didn't do something odd like reversing directions while one occluded the other. On the other hand, if the distractor entered the visual field so close to the focus person that their perceptually indistinguishable shirts overlapped immediately in the robot's visual field, then such a robot would probably not make the right decision. The robot's vision is based on color blob tracking, and if

[3]Personal communication from Stuart C. Shapiro, reporting a demonstration that he observed.

the two shirts overlap in its visual field the robot would see only a single blob of the correct color. Because the robot would see only a single object, one that has been under "continuous observation" as far as the robot is concerned, a robot equipped with the PIO identification model described in this dissertation may still lose the focus person. The robot will likely not even realize that there has ever been a second person, especially if the focus person leaves in the direction that the distractor person arrived from. Because the Cassie simulation does not use color blob vision, I cannot test this empirically, and so the above discussion has been speculative.

Chapter 6

Conclusion.

6.1 Summary.

6.1.1 The nutshell summary.

This dissertation has been a report on an investigation into how Perceptually Indistinguishable Objects (PIOs) can be identified. The empirical experiment with human participants evidence that people can identify PIOs in many cases. It also showed the strategies that people use to identify PIOs while counting and following. It showed that some strategies were more preferred and more effective in the identification task. A new cognitively plausible computational model of identifying PIOs was developed from the results of the human subjects experiment; the strategies that people preferred in the empirical experiment are the base and intermediate cases of the computational theory. Finally, a new simulated cognitive robot with an implementation of the computational model showed that the model works for the counting tasks. The robot simulation performed the counting tasks from the empirical experiment at the same level that people performed. This indicated that the computational theory has merit as a cognitively plausible theory of PIO identification. Each of these contributions is discussed in more detail in its own section below.

6.1.2 Human identification of PIOs.

After accounting for recording failures, the experiment yielded 133 transcripts for analysis. The transcripts were analyzed both quantitatively and qualitatively through an examination of the transcripts, and the codes assigned to the transcript segments.

The qualitative analysis showed that the human participants had a marked preference for certain situations that they believed helped them identify the PIOs more readily. Participants would try, as much as possible, to keep themselves in these situations. Participants liked to try to see as many PIOs as they could at once. If they could see multiple objects at once, then the participants knew that there were at least that many objects with that appearance in the world. Participants liked to continuously observe an object, particularly in the following tasks when it was possible to avoid identifying some of the PIOs in the task so long as the focus object was identified with absolute certainty. Participants made very strong assumptions about objects that they believed had a unique appearance (and, therefore, were not PIOs). Whenever the participants saw something that they believed to have a unique appearance, they instantly believed they were seeing that very object. Lastly, participants had a very easy time identifying immobile objects in the experiment, relying on location information nearly to the exclusion of any other information. Participants who were able to place themselves into these situations and had correct beliefs were almost always able to identify objects correctly.

Many participants in the experiments also showed a similar tendency to favor situations that were slightly less reliable, but still easy to consider and usually accurate. Many participants who saw an object in front of them that was perceptually indistinguishable from one that they had recently seen would quickly look behind them to make sure that the previously seen object was still there; in this way, the participant could be assured that there were at least two such objects in the world. Participants also sometimes believed that only one object could be in a particular room or other location at the time the participant was about to enter; when the participant entered and found only one, the participant believed that the object seen was the one that they thought would be there and identified the object as that one. When there was no other agent able to move a stationary object, participants often treated them like immobile objects and used the object's location as the primary means to identifying the object. There were also several participants who claimed to be continuously viewing an object, when in fact they had lost sight of the object several times for short periods. However, the participants appeared to keep a sort of mental marker on the object and returned to viewing it in the spot that the object had moved to, rather than the last spot that the participants had seen it.

If participants were not able to put themselves into a position to take advantage of any of the situations described above, they could still sometimes identify objects correctly. However they were much more often not successful than if they could use the strategies implied by the use of the above situations. Participants in this situation would use a variety of cues to identify PIOs. Those cues included knowledge about the motivations of the object to be identified and knowledge about what an object with those motivations ought to do. The rate of speed that an object moved was important as well. Another factor that participants used to identify PIOs was the amount of time since the participant had last seen the object that he believed might be the one he was currently encountering. Participants often used a combination of these factors to identify

PIOs.

The quantitative results of the experiment provided empirical data that shows that people can indeed identify PIOs. The quantitative analysis also showed that people identify perceptually *distinguishable* objects more readily than PIOs by a significant margin. There was no significant difference between identifying PIOs in the different tasks. However, there were significant differences in participant success based on the participant's prior experience with first-person video games. More experienced participants did significantly better than medium-experienced participants, who themselves did better than unexperienced participants. This greater success is explained by the fact that the more experienced the participant was with the control of this sort of game, the more the participant was able to use the easier methods of identifying PIOs described above.

6.1.3 A computational model of identifying PIOs.

A new cognitively plausible computational model of identifying PIOs, developed from the results of the human participant experiments is presented. The model uses an intensional representation so that the terms in the modeling language represented mental concepts rather than objects in the world.

The cues and strategies that participants in the experiment went out of their way to use are treated separately. When participants were able to use these preferred strategies, they reused their existing mental entity for the object identified whenever that object was one that the participant had already identified. Some participant-preferred strategies always led to the correct answer/identification when the participant's background beliefs were correct. These strategies are generally perceptually based. They were called base cases. The four base cases are:

- simultaneous perceptions,
- objects with a unique appearance,
- immobile objects
- continuous viewing.

The simultaneous-perceptions case is used to identify two perceptions as being different objects. The other three cases — objects with a unique appearance, immobile objects, and continuous viewing — are used to identify a current perception as representing the same object as a previous perception.

The other set of strategies that the participants tried to use were not quite as perceptually based and are called intermediate cases. These strategies, when correctly used, lead to the right answer a great deal of the time, but are slightly more prone to failure than the cases summarized above. The intermediate cases in the identification of PIOs are:

- rapid perceptions
- locally unique objects
- stationary objects
- continually perceived objects.

These are analogous to the base cases in that the rapid perceptions case is used to identify two perceptions as being different objects, while the remaining three — locally unique objects, stationary objects, and continually perceived objects — are used to identify two perceptions as representing the same object.

The remaining strategies used by human participants were not preferred and collectively make up the general case of identifying PIOs. When a participant used the general case to identify PIOs, the participant created a new mental entity and then wondered if the currently considered object was the same as the one the participant had previously conceived of. The computational model therefore creates a new intensional entity.

The five functions of the algorithm further formalize the amalgamated behavior of all of the experiment participants into a computational model of identifying PIOs. In the algorithm, the base and intermediate cases are considered first in an interleaved manner. A base case is considered just before its related intermediate case. Thus, immobile objects are considered just before stationary objects and then continuously viewed objects are considered just before continually perceived objects etc. If a base or intermediate case can identify all of the PIOs in the current visual field, the algorithm ends at that point. If not, then for any remaining perceptions, the general case of identifying PIOs is used. The agent using the algorithm creates a new mental entity and then reasons whether the currently perceived object is the same as one of the PIOs it has perceived before.

6.1.4 A PIO-aware simulated robot.

Crystal Cassie is a simulated cognitive agent, built on GLAIR, an existing architecture for representing the control of cognitive agents, SNePS, an existing knowledge representation and reasoning system, and Crystal Space, an off-the-shelf 3-D engine produced by a community of hobbyists. Crystal Cassie can understand and generate sentences in a fragment of English using a GATN grammar, and has a small suite of primitive acts which allow her to perform several small tasks including finding and counting objects which require the identification of PIOs. GLAIR is a three-layer architecture(KL, PML, SAL), the middle layer of which is separated into three sublayers. These five layers are implemented in two processes, P1 and P2, written in different computer languages, that communicate over four socket connections representing the modalities of speech, hearing, vision, and action. The lowest GLAIR layer communicates with two additional processes, implementing Cassie's sensor and effector organs. These communications take place over two two-way, and one one-way socket connections, for natural language interaction (two-way), action (two-way), and vision (one-way). The process that contains Cassie's vision organ and action effectors also contains a program that operates the simulated world that she occupies. A human user/interlocutor/observer can talk with Cassie using a keyboard and display connected to her natural-language-interaction process, and can observe the simulated world "through Cassie's eyes" via a display attached to the world simulator.

Crystal Cassie was designed and implemented in order to develop a computational theory of identifying perceptually indistinguishable objects. With the computational model of identifying PIOs that was described in chapter 4 implemented at the PML level of the Crystal Cassie simulation, Cassie can count both moving and stationary objects about as well as people do. Her success rate was not significantly different from the success rate of people at the same tasks. Cassie's mistakes fall into a subset of the mistakes that the human participants made in the experiment. In particular, Cassie was vulnerable to compression and dispersion errors and to missing robots that wandered in such a way that she never saw them when she scanned the room.

For those who would like to use Crystal Cassie themselves, Appendix B describes how to get, build and use the simulation at the University at Buffalo.

6.2 Future work and directions.

6.2.1 Ease of extensions and new directions.

The work described in this dissertation lends itself naturally to several refinements and new lines of research. Some of these refinements and possible directions are discussed in this section.

6.2.2 Following task in the simulation

The following task requires a bit more work in order to work in the simulation. The implementation is mostly in place, but Cassie needs to find and go through the closest door when she loses her robot. Once through the door she will need to quickly find the robot and identify it. Currently, every identify takes a little longer than the one before. Some additional optimization at the KL level would be needed to complement the implementation of finding the nearest door at the PML and SAL/body levels.

6.2.3 A hardware robot

One possible future direction would be to move the the test platform from a simulated robot to a real robot. A simulated robot was appropriate for this research because it is the same environment that the human participants used. However, it has been asserted (Brooks, 1992) that simulated robots, no matter how realistically programmed, are never completely accurate and lead to a certain amount of "cheating." A simulation is by necessity a model of the world; and in models one must choose the relevant parts of an object to model. Because the builder chooses the constraints on the model, that builder may inherently bias the model in some way. In addition, even if there is no real bias built into the simulated robot or its simulated world, the full breadth of a real world experience cannot be fully simulated. There are innumerable permutations and unexpected occurrences that can happen in the real world that will better enable a thorough test of the computational model described in this dissertation. This can be as simple as an unexpected student walking between the robot and the object it is trying to identify. In a simulation, this sort of activity would have to be rigorously planned and implemented. The test of how this would affect the identification could be done in a few seconds (planned or unplanned) in the real world. Smith (1991) agrees, arguing that complex computer simulations are impossible to prove completely correct without thoroughly testing the programs in the real world. A simulated test of a computer system is never a complete test that should make a user completely convinced that the program will work in the world.

6.2.4 Additional testing of human participants

The human-participants experiment was designed to elicit the maximum amount of information possible about a topic that had not been directly addressed. The tasks were designed to force people to identify PIOs and to elicit how people identify those PIOs. The tasks focused on two basic types of object: glasses that could neither move nor be moved in the task, and robots/people who move and can easily be anthropomorphized as motivated, thinking beings. This leaves at least one immediate hole: a task in which stationary but movable objects are to be identified. In this task, a second agent in the world would have the opportunity to move these objects around. The participants in such an experiment would have to be aware of both the stationary nature of the objects themselves and the other agents' capabilities to consistently identify all of the objects. The other agent might be relatively random. It might have a pattern to its movements (and thus perhaps have a motivation ascribed to it by some participants). Or the other agent might be designed to make the task more difficult by moving objects mostly from rooms that the participant had recently been in. I would expect that in such an experiment, participants would likely identify these objects as stationary and not moving unless they saw the other agent or noticed that one of the stationary objects was in a significantly different position. If a participant did notice the actions of the other agent, I hypothesize that the participant would be forced to treat these stationary objects much more like mobile objects because there was another agent that could and did act as a mover. Therefore the assumption that stationary objects will be in the same location would no longer hold.

Another refinement that could be made to the human participant experiment would be to exploit the natural heuristics that humans use. For example, a robot-following task could be developed in which the focus robot went behind an occluding object and then stopped there briefly while a second (distractor) robot left the occluder at about the same time that the first robot would have, had it continued at its original pace. It would be interesting to investigate what effect this variation would have on the success rates of the participants. It would also be interesting to investigate whether the inherent difficulty of identifying PIOs would make participants more likely to set aside the "continually perceived object" heuristic in this situation. I would expect in this case that human performance would deteriorate considerably if these human heuristics were exploited in an attempt to fool participants. I expect that some of the participants would notice the problem, but I expect that most would not.

One last extension of the human participant experiment that is immediately evident is to do experiments similar to the one described in this dissertation, but design these new experiments as more traditional psychology experiments. That is, make them experiments in which a trial of one to two dozen participants are given a task, and then a similar number does a nearly identical task that has a single change. This painstaking process of testing a single variable a time could take the general results developed here in this dissertation and refine them. Such a series of experiment could for example, determine which of the strategies enumerated in this dissertation help the most to identify PIOs for most people. Such a series of experiments could also determine more precisely in what sorts of situations people find the identification of PIOs harder or, conversely, in what situations people can easily identify PIOs. I expect that the strategies that I have called base cases will be the most useful to people in identifying PIOs. Those identified as intermediate cases will, I expect, be the next most useful. Finally, I expect that the various strategies used in the general case will be the least successful, though they are necessary when the base and intermediate cases are not useful.

6.2.5 Cleaning up the robot simulation

Two of the acts in the Crystal Cassie robot simulation, *follow* and *count,* were defined as primitive acts in the PML when they should really be complex acts at the KL. The *follow* act should be implemented as an iterated series of looking-for, recognizing, and going-to acts. Crystal Cassie should look for the object she is following, recognize the correct object as the one she is following, go to it, and repeat until she is told to stop. The *count* act is similar, Crystal Cassie should look around her current room, see and identify all of the objects of the correct type in the room, then look for a door that she hasn't taken yet, and then move to the next room and repeat. Both of these acts really need a new SNeRE primitive act to be easily implemented. They need a primitive act that combines the functionality of *sniterate* and *cascade.* The *sniterate* act repeatedly tries to execute its guarded acts until none of the guards holds true. When the guarded act is selected, the SNeRE system sends an acting request and then assumes that the act has been immediately completed, which in the *count* or *follow* acts is usually not the case, since often reasoning can take less time than moving between rooms. The *cascade* act on the other hand, takes a series of steps and waits until the goal of the previous step holds before beginning the next step. It will not however repeat its actions. When the final act in the *cascade* is done, so is the *cascade* itself. A combination of the functionality of these two would allow me to implement my follow and count actions as the KL complex acts that they really ought to be. The primitive act that I propose here could have the following operational semantics

> embodied-sniterate(object1) where object1 is a guarded sequence of acts. The sequence will only be executed if the guard is true. Each act in the sequence will only be executed after receiving feedback from the body that the previous act in the sequence is completed.

The current implementation should have no adverse impact on the results of the experiments, but counting and following seems more like knowledge level acts rather than primitive acts. Therefore they should be implemented as such.

6.2.6 Mapping in Cassie.

Cassie doesn't have any concepts of maps or relative locations. In order to make the model of PIO identification presented in this dissertation even more cognitively plausible, Cassie needs a good model of map making and understanding of locations in relation to nearby landmarks. Currently Cassie uses her concepts of locations to identify an object's location at the KL. However at the PML these concepts are aligned with a coordinate grid location. It would be better if Cassie had a more cognitive notion of location, like a location based on nearby landmarks. Cassie also needs a concept of a map. Many human participants, when counting, stated that they needed to get a good idea of the nature (the map) of the suite of rooms that they were in in order to count properly. For example P6 counting robots says: "I think I have an idea of the... of the layout of the rooms probably I'm just trying to make sure." If Cassie had a concept of a map, and the ability to map out a suite or rooms in her own mind the way people do, she could do the complete counting task rather than just counting in one room at a time.

6.2.7 Forming complex beliefs from sensor data.

The computational model proposed in this dissertation relies on an agent having accurate beliefs about the state of the world. As people, we continuously use our sensors (our senses) to gather information about the world and form beliefs about the world. Using those beliefs, we identify objects and plan our actions. While it is understandable that we sometimes form the wrong beliefs about the world, which can lead to mistaken conclusions later, people are generally quite capable of forming beliefs that are reasonable approximations of what is occurring in the world. These beliefs can be quite complex, including beliefs about the motivations of another agent or the purpose of a device that is not currently in use.

However, there has not been much work done on forming complex beliefs from sensor data for cognitive robots and other embodied AI-based reasoning systems. Many systems can form some pretty straightforward beliefs about objects in the world from the data acquired by their sensors. For example, target tracking systems can often form beliefs about the objects they are tracking. The WITAS system (Doherty et al., 2000) discussed in chapter 2 is an agent that can track a car as it drives along the road and can form beliefs about the car's position and its heading and speed. However, the WITAS system does not have any way of forming beliefs about the driver's motivation or plans from the sensor information it takes in, nor does it have a way of predicting where the car will move next beyond reliance in the limitations imposed by the road.

Other systems (like the one described by Kuipers and Beeson (2002)) can map out the area that the agent operates in, and form beliefs about which room is associated with which sensor readings. However, most cannot identify the purpose of the individual rooms from their contents or their position in relation to other rooms. Participants in the experiment did both, and doing so was often important to their identification of objects in the task.

The model described in this dissertation is a knowledge representation and reasoning model. As such, it can and does rely on some of the beliefs of the agent that ought to be formed directly from sensor data. Many of the beliefs of Crystal Cassie are formed directly from sensor data. However, some of the necessary beliefs, such as beliefs about the motivations/purpose of an object, or beliefs about the rate than an object is moving or the beliefs about which objects are continuously observed, are not currently formed from sensor data. A model of forming these beliefs that the PIO identification algorithm needs from the sensor data would allow this model to be much more useful. Forming beliefs about things like the motivations and plans of other agents from sensor data is the most significant missing piece that is needed to make the model described in this dissertation useful to a real agent.

Bibliography

Black, M. (1952). The identity of indiscernibles. *Mind*, 61:153–164.

Brooks, R. A. (1992). Artificial life and real robots. In *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life.*, pages 3–10, Cambridge, MA. MIT Press/Bradford Books.

Chachich, A., Pau, A., Barber, A., Kennedy, K., Olejniczak, E., Hackney, J., Sun, Q., and Mireles, E. (1997). Traffic sensor using a color vision method. In *Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS SPIE Proc. Vol 2902*, pages 156–165, Bellingham, WA. SPIE–The International Society for Optical Engineering.

Chalupsky, H. and Shapiro, S. C. (1996). Reasoning about incomplete agents. In *Proceedings of the Fifth International Conference on User Modeling (UM-96)*, pages 169–177. User Modeling, Inc.

Coradeschi, S. and Saffioti, A. (2003). An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2-3):85–96.

Coradeschi, S. and Saffiotti, A. (2001a). Forward. In Coradeschi, S. and Saffiotti, A., editors, *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems: Papers from the 2001 AAAI Fall Symposium, Technical Report FS-01-01*, page viii, Menlo Park CA. AAAI Press.

Coradeschi, S. and Saffiotti, A. (2001b). Perceptual anchoring of symbols for action. In *Proceedings of* *the 17th international joint conference on artificial intelligence (IJCAI-01)*, pages 407–412, San Francisco CA. Morgan Kaufman.

Doherty, P. (2004). Advanced reaserch with autonomous unmanned aerial vehicles. In Dubois, D., Welty, C. A., and Williams, M.-A., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004)*, pages 731–732, Menlo Park, CA. AAAI Press.

Doherty, P., Granlund, G., Kuchcinski, K., Sandewall, E., Nordberg, K., Skarman, E., and Wiklund, J. (2000). The WITAS Unmanned Aerial Vehicle Project. In Horn, W., editor, *ECAI 2000. Proceedings of the 14th European Conference on Artificial Intelligence*, pages 747–755, Berlin.

Ericsson, K. A. and Simon, H. A. (1984). *Protocal Analysis*. MIT Press, Cambridge, MA.

Fodor, J. A., editor (1994). *The Elm and the Expert*. MIT Press, Cambridge MA.

Forrest, P. (2002). The identity of indiscernibles. Stanford Encyclopedia of Philosophy: http://plato.stanford.edu/entries/identity-indiscernible.

Frege, G. (1892a). *On Sense and Nominatum*, pages 85–102. Reprinted in **?**.

Frege, G. (1892b). *On Sense and Reference*, pages 56–79. Reprinted in **?**.

Gopnik, A. and Sobel, D. (2000). Detecting blickets: How young children use information bout novel causal powers in categorization and induction. *Child Development*, 71(5):1205–1222.

Gough, M. (1996). The aesthetic, authenticity and the artistically valuable. *The Philosopher*, 84(1):13–14.

Han, K. and Veloso, M. (2000). Automated robot behavior recognition applied to robotic soccer. In Hollerbach, J. and Koditschek, D., editors, *Robotics Research: the Ninth International Symposium*, pages 199–204, London. Springer-Verlag.

Heintz, F. and Doherty, P. (2004). Managing dynamic object structures using hypothesis generation and validation. In Coradeschi, S. and Saffioti, A., editors, *Anchoring Symbols to Sensor Data, Papers from the AAAI Workshop, Technical Report WS-04-03*, pages 54–62, Menlo Park, CA. AAAI Press.

Hexmoor, H., Lammens, J., and Shapiro, S. C. (1993). Embodiment in GLAIR: a grounded layered architecture with integrated reasoning for autonomous agents. In Dankel, II, D. D. and Stewman, J., editors, *Proceedings of The Sixth Florida AI Research Symposium (FLAIRS 93)*, pages 325–329. The Florida AI Research Society.

Hexmoor, H. and Shapiro, S. C. (1997). Integrating skill and knowledge in expert agents. In P. J. Feltovic, K. M. Ford, R. R. H., editor, *Expertise in Context*, pages 383–404. AAAI/MIT Press, Cambridge, MA.

Ismail, H. O. (2001). *Reasoning and Acting in Time*. PhD thesis, University at Buffalo, The State University of New York, Buffalo, NY.

Ismail, H. O. and Shapiro, S. C. (2000). Two problems with reasoning and acting in time. In A. G. Cohn, F. G. and Selman, B., editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the Seventh International Conference*, pages 355–365, San Francisco, CA. AAAI, Morgan Kaufmann.

Jain, R., Kasturi, R., and Schunck, B. (1995). *Machine Vision*. McGraw-Hill, New York.

Jenkins, J. (1995). Where beauty lies - fakes and forgeries. *The Philosopher*, 83(2):6–11.

Johnson, S. P. (1998). *Object perception and object knowledge in young infants: a view from studies of visual development*, pages 211–239. Published in **?**.

Kaminka, G. A. and Avrahami, D. (2004). Symbolic behavior-recognition. In *Proceedings of the AAMAS 2004 Workshop on Modeling Other Agents from Observations (MOO 2004)*.

Kaplan, A. N. and Schubert, L. K. (2000). A computational model of belief. *Artificial Intelligence*, 120(1):119–160.

Kuipers, B. and Beeson, P. (2002). Bootstrap learning for place recognition. In *Proceedings of the Eighteenth National Conference on on Artificial Intelligence (AAAI-02)*, pages 174–180, San Francisco, CA. Morgan Kaufmann.

Kuipers, B. and Byun, Y.-T. (1991). A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:47–63,.

Leibniz, G. W., Leroy E. Loemker, E., and Translator (1969). *Philosophical Papers and Letters*. Dordrecht: D. Reidel Publishing Company, second edition.

Lessing, A. (1965). What is wrong with a forgery? *Journal of Aesthetics and Art*, 23(4):461–471.

Lozano-Perez, T. and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral objects. *Communications of the ACM*, 22(10):560–570.

Maida, A. (1993). Propositionally representing incomplete knowledge about existence. *Journal of Theoretical and Experimental Artificial Intelligence*, 5:185–197.

Maida, A. S. and Shapiro, S. C. (1982). Intensional concepts in propositional semantic networks. *Cognitive Science*, 6(4):291–330. Reprinted in **?**.

Mandler, J. M. (1991). Prelinguistic primitives. In *Proceedings of the Seventeenth Annual Meeting of the Berkeley Linguistics Society*, pages 414–425, Ann Arbor. Braun-Brumfield Inc.

McGlaun, G., Althoff, F., Lang, F., and Rigoll, G. (2002). Towards multimodal detection and classification of emotional patterns in human-machine-interaction – results of a baseline study. In Callaos, N., Porter, J., and Rishe, N., editors, *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics Information Systems Development II*, pages 136–141.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97.

Nakatsu, R., Nicholson, J., and Tosa, N. (1999). Emotion recognition and its application to computer agents with spontaneous interactive capabilities. In *Proceedings of the Seventh ACM International Conference on Multimedia (Part 1)*, pages 343 –351, New York. ACM Press.

Newell, A. and Simon, H. A. (1972). *Human Problem Solving*. Prentice-Hall, Englewood Cliffs NJ.

Papanikolopoulos, N., Khosla, P., and Kanade, T. (1993). Visual tracking of a moving target by a camera mounted on a robot: A combination of control and vision. *IEEE Transactions on Robotics and Automation*, 9(1):14–35.

Pardas, M., Bonafonte, A., and Landabaso, J. L. (2002). Emotion recognition based on MPEG4 facial animation parameters. In *IEEE International Conference on Acoustics, Speech and Signal Processing 2002*, pages 3624–3627, New York. Wiley- IEEE press.

Piaget, J. (1948). *The Child's Conception of Movement and Speed*, pages 520–546. Reprinted in **?**.

Piaget, J. (1952). *The Origins of Intelligence in Children*. W. W. Norton, New York. Translated by Margaret Cook.

Piaget, J. (1954). *The Construction of reality in the child*. Basic Books, New York. Translated by Margaret Cook.

Pollock, J. (1974). *Knowledge and Justification*. Princeton University Press, Princeton.

Pylyshyn, Z. (1989). The role of location indexes in spatial perception: A sketch of the FINST spatial-index model. *Cognition*, 32(1):65–97.

Pylyshyn, Z. and Storm, R. (1988). Tracking multiple independent targets: Evidence for a parallel tracking mechanism. *Spacial Vision*, 3(3):179–197.

Rowe, A., Rosenberg, C., and Nourbakhsh, I. (2002). A low cost embedded color vision system. In *Proceedings of IROS 2002*.

Russell, B. (1905). On denoting. *Mind*, 14(56):479–493. Reprinted in **?**.

Sandewall, E. (2002). Use of cognitive robotics logic in a double helix architecture for autonomous systems. In Beetz, M., Hertzberg, J., Ghallab, M., and Pollack, M. E., editors, *Advances in Plan-Based Control of Robotic Agents*, Lecture Notes in Computer Science, pages 226–248, New York. Springer Verlag.

Sanka, M., Hlavac, V., and Boyle, R. (1999). *Image Processing, Analysis and Machine Vision: Second Edition*. Brooks/Cole, Pacific Grove, CA.

Santore, J. F. and Shapiro, S. C. (2003). Crystal Cassie: Use of a 3-D gaming environment for a cognitive agent. In Sun, R., editor, *Papers of the IJCAI 2003 Workshop on Cognitive Modeling of Agents and Multi-Agent Interactions*, pages 84–91, Menlo Park, CA. AAAI Press.

Santos, P. E. (2003). *Spatial Reasoning and Abductive Interpretation of Sensor Data Obtained by a Mobile Robot in Dynamic Environment.* PhD thesis, Imperial College, London.

Schmidt, W. C., Fisher, B. D., and Pylyshyn, Z. W. (1998). Multiple-location access in vision: Evidence from illusory line motion. *Journal of Experimental Psychology*, 24(2):505–525.

Shapiro, S. C. (1982). Generalized augmented transition network grammars for generation from semantic networks. *The American Journal of Computational Linguistics*, 8(1):12–25.

Shapiro, S. C. (1989). The CASSIE projects: An approach to natural language competence. In Martins, J. P. and Morgado, E. M., editors, *EPIA 89: 4th Portugese Conference on Artificial Intelligence 390*, pages 362–380. Springer-Verlag.

Shapiro, S. C. (1993). Belief spaces as sets of propositions. *Journal of Experimental and Theoretical Artificial Intelligence (JETAI)*, 5(2 & 3):225–235.

Shapiro, S. C. (1998). Embodied Cassie. In *Cognitive Robotics: Papers from the 1998 AAAI Fall Symposium, Technical Report FS-98-02*, pages 136–143. AAAI Press, Menlo Park, CA.

Shapiro, S. C. (2000). SNePS: A logic for natural language understanding and commonsense reasoning. In Łucja Iwańska and Stuart C. Shapiro, editor, *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*, pages 175–195. AAAI Press/MIT Press, Menlo Park CA.

Shapiro, S. C. and Ismail, H. O. (2001). Symbol-anchoring in Cassie. In Coradeschi, S. and Saffioti, A., editors, *Anchoring Symbols to Sensor Data in Single and Multiple Robot Systems: Papers from the 2001 AAAI Fall Symposium, Technical Report FS-01-01*, pages 2–8, Menlo Park, CA. AAAI Press.

Shapiro, S. C. and Ismail, H. O. (2003). Symbol anchoring in a grounded layered architecture with integrated reasoning. *Robotics and Autonomous Systems*, 2–3:97–108.

Shapiro, S. C., Ismail, H. O., and Santore, J. F. (2000). Our dinner with Cassie. In *Working Notes for the AAAI 2000 Spring Symposium on Natural Dialogues with Practical Robotic Devices*, pages 57–61, Menlo Park, CA. AAAI.

Shapiro, S. C., Kumar, D., and Ali, S. (1989). A propositional network approach to plans and plan recognition. In *Proceedings of the AAAI 1988 Workshop on Plan Recognition*, page 21, Los Altos, CA. Morgan Kaufmann.

Shapiro, S. C. and Rapaport, W. J. (1987). SNePS considered as a fully intensional propositional semantic network. In Cercone, N. and McCalla, G., editors, *The Knowlege Frontier*, pages 263–315. Springer-Verlag, New York.

Shapiro, S. C. and Rapaport, W. J. (1992). The SNePS family. *Computers and Mathematics with Applications*, 23(2-5):243–275. Reprinted in (**?**, pp. 243–275).

Shapiro, S. C. and the SNePS Implementation Group (2004). *SNePS 2.6.1 User's Manual*. Department of Computer Science and Engineering, University at Buffalo, The State University of New York, Buffalo NY.

Simons, D. J. and Levin, D. T. (1998). Failure to detect changes to people during a real-world interaction. *Psychonomic Bulletin & Review*, 5(4):644–649.

Singh, M. P. (1998). The intentions of teams: Team structure, endodeixis, and exodeixis. In Prade, H., editor, *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI-98)*, pages 303–307. John Wiley & Sons.

Smith, B. C. (1991). "limits of correctness in computers. In Dunlop, C. and Kling, R., editors, *Computerization and Controversy*, pages 632–646. Academic Press, San Diego, CA.

Sowa, J. F. (2000). *Knowledge Representation: Logical, Philosophical, and Computational Foundations*, pages 120–121. Brooks/Cole, Pacific Grove, CA.

Spelke, E. S., Breinlinger, K., Macomber, J., and Jacobson, K. (1992). Origins of knowledge. *Psychological Review*, 99(4):605–632.

Strawson, P. (1964). *Individuals*, pages 15–36. Methuen & Co Ltd, London.

Trafton, J. G., Shultz, A. C., Perzanowski, D., Bugajska, M. D., Adams, W., Cassimatis, N. L., and Brock, D. P. (2004). Children and robots learning to play hide and seek. *Cognitive Systems Journal*.

Tyberghein, J., Zabolotny, A., Sunshine, E., and et. al. (2002). *Crystal Space: Open Source 3D Engine Documentation*. http://crystal.sourceforge.net, .94 release edition. ftp://crystal.sourceforge.net/pub/crystal/docs/download/.

von Hofsten, C., Feng, Q., and Spelke, E. S. (2000). Object representation and predictive action in infancy. *Developmental Science*, 3(2):193–205.

Appendix A

Codes used in participant transcripts.

This appendix contains the list of codes used to encode the participant transcripts along with the semantics of those codes.

A.1 The Semantics of the functions used for protocol coding

Begin-task Space/time filling sound as the participant began to work on the task

Finish-Task Phrase representing that the participant believes the task is finished.

Rationale:X Participant gives X as the rationale for recent or current actions. Note that X is not a code in section A.2, but an English phrase. The rationales were often strategies and like the arguments to the strategy function listed below, were varied and are the focus of this chapter.

See(X[:y]) Participant reports noticing X, Optional :y indicates that participant has reported noticing y instances of X

Search(X) Participant is actively looking (and possibly moving in support of this looking) within the (usually room) X for the focus object(s)

Move(X) Participant moves in(to) room X (where 'in' can be read as within when X is 'current room')

Report(X) Summarize previous actions and verbalize the current state as X

InterResult→X class Participant reports an intermediate result (in tasks wherein the participant must count the number of a type of object) of having seen X members of the class of objects she was asked to count.

Identity(X) The identity of X, where X is some object being perceived at the time of vocalization by the participant.

Infer(X) Participant deduces X and believes it to be correct.

Speculate(X) Participant considers X but explicitly doesn't commit to the correctness of the proposition.

Label(X) Participant gives object X a name that participant can use later to refer to a unique object.

Follow(X) Participant reports following object X

Query(X) The participant has expressed question aloud directed to X

Ascribe(X, Y) Participant ascribes X as being a motivation/intention of Y

perceptual-props(X) The perceptual properties of X (usually a room or focus object) that the participant reports in some way.

Fail(X) Participant realizes failure of X and reports it. X can be an action, strategy or plan.

Strategy(X) Participant reports using X as a means of carrying out the task. Note that the X from here is not a code in §A.2, but an English phrase. The strategies were varied and are the focus of this chapter.

Anticipate(X) Participant tries to anticipate where X will move.

Doubt Participant voices doubt about recent inferences

Navigational-hindrance Participant reports trouble navigating in the virtual world (either at the level of participant's virtual avatar or at the level of keyboard navigation.)

Plan(X) Participant reports X as a plan of future actions, Not a strategy intended to solve the task, but a plan of action (usually in service of a strategy)

Both-move(X) Participant reports moving with a focus object to location X (usually in the form of "we are moving...")

Comment Participant says something irrelevant to the task.

Talkto(X) Participant addresses a question/request/command directly to X. X is usually a moving virtual robot or person in the experiment.

Lose(X) Participant was trying to follow X but reports losing track of X

Choose(X) Participant has several courses of action available, and chooses X. This function has two uses:

1. Participant has lost track of the focus object and chooses X as the visible object that the participant believes is the focus object.
2. Participant has two or more rooms/doorways to enter and chooses X as one of them.

Difficulty-Talking The participant has difficulty talking while performing the task.

Memory(X) The participant refers to X, something that occurred earlier in the task.

Correct(X) The participant is correct in choosing or predicting X.

A.2 The Semantics of the Objects in the coding.

[PIO]Distractor A moving object in the suite that is not the object that the participant has been told to focus on. Optional PIO prefix indicates that the distractor is perceptually indistinguishable from the focus object. Used in the following task for non-focus people or robots.

Furniture An immobile object in a room other than one that the participant has been asked to focus on.

FocusObject|*objectname* An object of the type that the participant has been asked to focus on, either by following or counting the object. *objectname* refers to some common type of object such as "glass" or "robot"

CurrentRoom Refers to the room that the participant is currently in at the time of vocalizing.

Room{1...n} Refers to a known room other than the current room referred to by the participant. The room1 is the first that the participant entered, with room2 being the second and so on. Only used if it is clear that the participant knows the identity of the room.

AdjacentRoom Refers to the adjacent room when it is not clear from the transcript that the participant knows the identity of the adjacent room

Door Refers to a doorway leading from the current room to an adjacent room in the virtual environment.

Result In counting tasks, the final number of objects that the participant reports.

Map The layout of the suite of rooms for the given task.

Self Participant reference to the participant.

Experimenter Participant reference to the experimenter.

[Focus|Distractor]Robot Movement The participant's belief about the movement capabilities, or actual movements of the robots in the task. In tasks where the participant had to follow a focus object, "focus" or "distractor" was often prepended if the participant clearly knew which of the two was being referred to.

Task A participant reference to the experimental task as assigned by the experimenter.

Nothing Nothing, no objects.

Appendix B

Crystal Cassie User Manual.

B.1 Appendix contents.

This appendix details how to get, build and run the Crystal Cassie simulation on the UB CSE systems. It assumes that the /projects/jsantore directory exists as it did when this appendix was written.

B.2 Getting the simulation.

You will only need to do the steps in this section once for each machine you want to run the simulation on.

The simulation is kept in a cvs repository. To retrieve a copy for yourself you will need to point your cvs client at the repository. If you are working from a UB cse machine set your cvs root as follows

setenv CVSROOT /projects/jsantore/cvs

This will change your CVSROOT to that used for this project in the current shell only. All other shells on the system will use your original CVSROOT. If you wish you access your projects that use a different CVSROOT in the current shell, you will have to set the CVSROOT environment variable back to its original value.

If you have a UB cse account, but are working from a machine that does not mount the /projects directory directly (like the department linux machines or a personal machine), then use the following for your cvs root.

setenv CVSROOT :ext:<USERNAME>@hadar.cse.buffalo.edu:/projects/jsantore/cvs

Where <USERNAME> should be replaced with your cse.buffalo.edu username. Once your cvs client has its root directory set, check out a copy of the *crystalWork* project for yourself. Other clients may have a graphical way to do this, but this appendix will assume the use of the standard command line cvs client. Checkout the project by moving to the directory in which you want to have the project as a subdirectory and do the following:

cvs checkout crystalWork

At this point you will have a copy of all of the Crystal Cassie-specific source code you need for the simulation. You will need two more things to run the simulation: a compiled version of the Crystal Space tools version

.96, and a working version of SNePS. Note that these need not be on the same machine. The Crystal Space simulation runs well on Linux, but not on Solaris. If you decide to run the simulation across two machines, you will of course need to get two copies of the simulation, one copy on the machine you run the lisp side of the project on, and one copy on the machine that you run the Crystal Space graphics side of the project on. You must run the Crystal Space part of project on the same machine that you display it on. It will not display on another machine.

You must put the Crystal Space tools in a directory that is a sibling of your crystalWork directory. That is, both the CS directory for the crystalspace tools and the crystalWork directory for the simulation should be in the same parent directory. You can retrieve a patched copy of Crystal Space v.96 from /projects/jsantore/CS-diss-patched.tar.gz. I recommend this copy as I've fixed at least one nasty bug, and I have setup all of the world files in that copy. Once you download it, you should be able to compile as suggested in the user manual. See the README.html file in the docs subdiretory and follow the links to chapter 2 of the Crystal Space users manual which is dedicated to the various build options and requirements on different platforms. If your system has the gcc compiler suite, the Z Library, PNG Library, and JPEG Library installed, then you can probably compile the project by doing the following typical steps in your CS directory.

```
./confgure
make -k all
```

You should compile Crystal Space before moving on to the next section.

B.3 Building the simulation.

You should only need to do the steps in this section once for each machine your run the simulation on.

This section assumes that you have retrieved the crystalWork project and that you have sucessfully built the Crystal Space tools. The Crystal Space CS directory is assumed to be a sibling directory to your crystal-Work directory as specified above.

Make the crystalWork directory your working directory. Edit the Makefile to change the following environment variables from their default values to those appropriate for your system. The variables you might need to change are:

BASEDIR=<full path of directory containing your crystalWork and CS directory>

BUILDFLAGS=<your optimizations> *;you can set optimizations for your process here or remove the flag without harm*

LIBS=<the subdirectory of your CS/out directory that actually contains the object files>*; if you cd into the out subdirectory of your CS directory, there will be a single directory tree there. There are three subdirectories before the actual object files, one for the operating system (unix/windows etc), one for the processor type (x86 etc) and one for either* debug *or* optomize. *Tell the simulation exactly where the object files are using this flag.*

GCC=<if you aren't using gcc/g++ put your compiler command here> *;if you find that your version of gcc doesn't compile the simulation, or you have no gcc installed, you can use another compiler, like cc to compile the simulation. In order to do so, change this variable to the name of your compiler.*

Once these changes have been made, compile the tools using the command

make all

When the compiler is finished you should have three new programs: ConsoleIn, Mundus, and SalServer.

B.4 Starting the simulation.

Once you have retrieved and built the tools as described above, you can run the simulation as follows:

1. set the CRYSTAL environment variable to be your crystal space directory e.g.
   setenv CRYSTAL <my/CS-directory>

2. In the crystalWork directory, run the *robotSim* shell script. You may also use the optional *exp1* argument. The argument *exp1*, if used, will begin the simulation in the small, four-room world. If the

optional argument is not present, then the larger academic suite is the default. The *robotSim* script will launch the three programs and pop up a short set of directions to set up the network connections. Follow those directions.

3. Start lisp and load SNePS 2.60.
4. cd into the crystalWork/lisp directory and start SNePS. (this subdirectory was created when you checked out the crystalWork project from cvs.)
5. execute from a SNePS prompt: *(intext "cassie-mentalstate.sneps")*
6. exit sneps
7. :pa PMLb
8. execute *(setup-SAL-onhost "*HOSTNAME*")* where HOSTNAME is the name of whatever machine you ran *robotSim* on above. For example, if I were running the SAL on cicero.cse.buffalo.edu, I would use the following command: *(setup-SAL-onhost "cicero.cse.buffalo.edu")*
9. hit return four time (you will see your prompt come back when you are done).

The simulation should now be running.

B.5 Using the simulation.

All of the user interaction should take place in the NLI (the window with the title "Cassie's Natural Language Interface") and the Mundus (the window with the title "Cassie's view of the World"). Cassie's actions will be visible on the Mundus. Users can type commands in natural language and will see Cassie's response in the NLI.

Cassie will accept the suite of sentances accepted by the FEVAHR version of Cassie with the following additions:

- Advance/Go forward: Cassie will (try to) go forward a fixed distance.

- Turn right/left: Cassie will turn approxamately 60 degrees in the specified direction.
- Count the glasses/robots/etc: Cassie will count the objects of the right type that are in the current room and report how many she know about thus far.
- Follow TourBot: Cassie will follow the robot that starts at the same place as the robot that the human participants followed (use the larger world and press 'F8' from the list of world commands below)

In addition to these new commands, cassie can talk about many of the objects that will be found in these simulations but were not found in the FEVAHR simulations, such as computers, glasses, tables etc.

In order to have moving objects in the simulation, the user must select the window entitled "Cassie's view of the World" (the Mundus process) and press one of the function keys from the top of the keyboard. These keys have the following symantics:

F2	Four gray, randomly moving robots will be created.
F3	Five randomly moving robots, two gold and three gray, will be created.
F4	reserved for debugging
F5	Used for connecting to the SAL - don't use at runtime
F6	Used for connecting to the SAL - don't use at runtime
F7	reserved for debugging
F8	Five robots with predefined paths.
F9	Five people with predefined paths

B.6 Quitting the simulation

To quit the simulation gracefully, select the SalServer process (the Crystal Space window with no title) and press the escape <ESC> key. Doing so will close all three Crystal Space windows and gracefully terminate the connection on the lisp side.