Social Robot Navigation

Rachel Kirby

CMU-RI-TR-10-13

Submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

May 2010

Thesis Committee:
Reid Simmons, Co-Chair
Jodi Forlizzi, Co-Chair
Illah Nourbakhsh
Henrik Christensen (Georgia Institute of Technology)

ABSTRACT

Mobile robots that encounter people on a regular basis must react to them in some way. While traditional robot control algorithms treat all unexpected sensor readings as objects to be avoided, we argue that robots that operate around people should react *socially* to those people, following the same social conventions that people use around each other.

This thesis presents our COMPANION framework: a Constraint-Optimizing Method for Person–Acceptable NavigatION. COMPANION is a generalized framework for representing social conventions as components of a constraint optimization problem, which is used for path planning and navigation. Social conventions, such as personal space and tending to the right, are described as mathematical cost functions that can be used by an optimal path planner. These social conventions are combined with more traditional constraints, such as minimizing distance, in a flexible way, so that additional constraints can be added easily.

We present a set of constraints that specify the social task of traveling around people. We explore the implementation of this task first in simulation, where we demonstrate a robot's behavior in a wide variety of scenarios. We also detail how a robot's behavior can be changed by using different relative weights between the constraints or by using constraints representing different sociocultural conventions. We then explore the specific case of passing a person in a hallway, using the robot Grace. Through a user study, we show that people interpret the robot's behavior according to human social norms, and also that people ascribe different personalities to the robot depending on its level of social behavior.

In addition, we present an extension of the COMPANION framework that is able to represent joint tasks between the robot and a person. We identify the constraints necessary to represent the task of having a robot escort a person while traveling side-by-side. In simulation, we show the capability of this representation to produce behaviors such as speeding up or slowing down to travel together around corners, as well as complex maneuvers to travel through narrow chokepoints and return to a side-by-side formation.

Finally, we present a newly designed robot, Companion, that is intended as a platform for general social human–robot research. Companion is a holonomic robot, able to move sideways without turning first, which we believe is an important social capability. We detail the design and capabilities of this new platform.

As a whole, this thesis demonstrates both a need for, and an implementation and evaluation of, robots that navigate around people according to social norms.

ACKNOWLEDGEMENTS

First and foremost, I'd like to thank my wonderful husband, Brian Kirby. He keeps me sane (at least mostly). Without him, I doubt I would have finished this thesis. He even built me a robot (see Chapter 7)!

I would also like to thank my advisors, Reid Simmons and Jodi Forlizzi. They have always supported my research, even when I had convinced myself it was all wrong (again), and they both have always kept a box of tissues around for those occasions. Reid found more bugs in my code for me than I care to admit, and Jodi taught me how to run a proper user study.

Thanks to my parents, who supported me through the whole process, and to my sister Beth for editing this document.

Thanks to all of the administrative staff who helped make graduate life much easier: Suzanne Lyons-Muth, Jean Harpley, Karen Widmaier, Kristen Schrauder, David Casillas.

Thanks to the myriad of robograds who have helped me along the way, especially those who already finished their degrees and thus proved that it's possible: Frank Broz, Jonathan Hurst, Sanjeev Koppal, Tom Lauwers, Marek Michalowski, Maayan Roth, Brennan Sellner, Kristen Stubbs, and everyone else. Thanks to other friends who've helped keep me sane: Brina Goyette, Emily Hamner, Krissie Lauwers.

Thank you!

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

Mobile robots that encounter people on a regular basis must react to them in some way. Traditional robot control algorithms for path planning and obstacle avoidance treat all unexpected sensor readings identically: as objects that must be avoided. For a mobile robot that operates near and with people, however, these traditional methods may not follow human social norms. Even a simple convention, such as passing oncoming people in a hallway by moving to the right side, might not be honored by a naïve obstacle avoidance algorithm. However, people generally perceive robots—particularly assistive robots, which must move around people—as human-like, even when the robots are non-anthropomorphic (e.g., Siino and Hinds, 2004). When such robots behave counter to what is socially expected, breakdowns in human–robot interaction occur (e.g., Mutlu and Forlizzi, 2008). While some algorithms have been developed to produce various particular social behaviors around people, they typically do so in a local, reactive way, which may not result in socially correct behavior overall. Such algorithms also are not generally extensible to other situations or additional social conventions.

To address these issues, we have developed a navigational framework for human–robot physical social tasks, such as navigating through crowds or waiting in line. We call our framework COMPANION: a Constraint-Optimizing Method for Person–Acceptable NavigatION. COMPANION is a generalized framework for representing social conventions as components of a constraint optimization problem. Social conventions, such as personal space and tending to the right, are described as mathematical cost functions. These costs are then used in path planning along with more typical task-related metrics, such as the shortest distance.

A key aspect of the COMPANION framework is that social conventions are addressed as part of a global path-planning problem. We argue that people do not apply social conventions in rigid ways, as would be achieved by treating conven-

tions as reactive behaviors. Rather, people consider *global optimality*, trading off different conventions according to the particular situation and their personal preferences. By modeling social conventions as part of a global optimization problem, the COMPANION framework can produce very human-like behaviors.

Our approach is unique in that it is capable of expressing an arbitrary number of social conventions, it explicitly accounts for these conventions in the planning phase, and it is intended to produce socially acceptable, human-like paths.

1.1 Motivation

In recent years, several commercial robots have been designed for deployment in hospitals and office buildings, typically to act as couriers for medications, paperwork, and the like. As more of these robots being used daily, researchers are able to study how people interact with the robots—as well as how people *expect* the robots to behave. An ethnographic study of the introduction of the Pyxis HelpMate robot in a California hospital by Siino and Hinds (2004) found that people most often thought of the robot as human-like, even before the physical robot arrived at the hospital. Many of the hospital workers maintained that viewpoint after the robot was in operation, despite its non-human-like appearance and behavior (Siino and Hinds, 2005). A similar study by Mutlu and Forlizzi (2008) investigated the Aethon TUG robot's use in a Pennsylvania hospital. One of their key findings was that many people felt "disrespected" by the robot because it failed to follow human social norms. Some of the robot's more egregious errors, according to the study's authors, included:

- failing to yield to oncoming people;
- stopping in the middle of the hallway for minutes at a time, while calculating alternate routes; and even
- colliding with people.

One of the main goals of our research is to improve the functionality of such courier-type robots by designing robot behaviors that mimic people's expectations. Doing so allows people to feel a sense of *common ground* (Clark, 1996) with a robot; that is, people will be able to draw on their knowledge of how other people behave when they are interacting with a robot. In particular, we believe that if such robots were able to navigate according to human social norms—such as yielding the right of way and respecting personal space—then they will be better accepted by the people around them. This will allow both the robots and the people to accomplish their jobs more smoothly and efficiently. We hope that our research

will encourage development of more robots with primarily social purposes, such as escorting people through hallways. Since such tasks are defined by social conventions, a robot that does not adhere to social norms may perform poorly or fail to complete its task.

1.2 General approach

In general, our research approach is to study human–human interaction through both literature and direct observations, look for similarities in behaviors across different people and interaction tasks, and use design principles to apply these behaviors to human–robot interaction. We then analyze the robot's behavior in human studies and use the results to further inform our design.

Throughout our work, we argue that robots should behave according to human social principles. Such social conventions are effortlessly used by people every day to interact with each other. We believe that, if a robot follows the same conventions, people will be able to have similarly understandable interactions with the robots. However, we also acknowledge that people may have very different expectations of how robots should behave. We address this issue by not only developing methods of having robots behave according to social norms, but also by studying how people interpret and react to such behaviors.

In particular, our interest is in *spatial* social interactions. We have studied various spatial social conventions as discussed in the literature (Chapter 2), and we have performed our own observational studies where the literature was found to be lacking (Chapter 3). We developed mathematical models of the human behavior, in such a way as to allow a robot to follow similar conventions when navigating through hallways (Chapter 4). We analyzed the behaviors both in simulation and in a controlled user study (Chapter 5), referring to psychological methods for analysis. Finally, we drew on those works to extend the model to the specific task of escorting a person side-by-side (Chapter 6). In addition, we used these results to inform the design of a new robot intended for these types of social tasks (Chapter 7).

1.3 Thesis statement

Thesis: *Human social conventions for movement can be represented as a set of mathematical cost functions. Robots that navigate according to these cost functions are interpreted by people as being socially correct.*

The first part of this statement states that we can model human behavior for various social tasks according to mathematical functions. We support this with

our navigational framework, COMPANION, and its use in both general hallway navigation and in side-by-side escorting. The second part of this thesis statement argues that people will interpret a robot's behavior as social if it navigates according to these cost functions. This both verifies that the cost functions do model human social conventions and also establishes that people interpret the behaviors of a robot in a way similar to how they interpret the behaviors of other people.

1.4 Contributions

This thesis provides four main contributions. First and foremost is our navigational framework, COMPANION, which is designed to produce robot behaviors that adhere to human social conventions. In addition, we provide results, both in simulation and in user studies with a physical robot, from our implementation of the framework in the particular situation of hallway navigation. Furthermore, we provide an extension to the framework that allows the robot to plan joint paths for escorting a person while traveling side-by-side. Finally, we introduce the Companion robot, a holonomic robot specifically designed for social human–robot interaction.

1.4.1 The COMPANION framework (Chapter 4)

We have designed a framework for social robot navigation, which we call COMPANION: a Constraint-Optimizing Method for Person–Acceptable NavigatION. The framework is composed of a set of mathematical constraints and objective functions that model human social conventions, such as avoiding people's personal space and tending to the right side of hallways, as well as task-based constraints, such as minimizing distance. The various functions are combined under a single heuristic path planner. By using a global, optimal path planner, the framework is able to produce results that accurately model human behavior.

Chapter 4 introduces the framework and defines a set of constraints that we believe are sufficient for producing robot behavior that models human social norms. In particular, we define cost functions for each of the following conventions:

- Minimizing the distance traveled to a goal, to conserve energy;
- Avoiding obstacles;
- Keeping a safety buffer around obstacles;
- Avoiding people, including keeping out of their personal space;
- Protecting the robot's own "personal" space;

- Tending to the right when passing people;
- Keeping a default velocity, so as not to expend extra energy;
- Facing the direction of travel, but allowing for sidestepping obstacles as people do; and
- Maintaining forward inertia, rather than zig-zagging repeatedly, which is both inefficient and socially awkward.

We argue that these constraints will produce social robot behavior in tasks that require passive social interaction, such as traveling through hallways, where the robot will encounter people and must react to—but not directly interact with—them. For tasks that require additional levels of interaction, additional constraints must be added, as discussed in Chapter 6.

Furthermore, we argue that the COMPANION framework can be used to create a *range* of socially acceptable behavior using different relative weights between the constraints. That is, though different sets of weights will produce different behaviors, we argue that socially acceptable behaviors will result from any number of such sets.

Finally, Chapter 4 discusses details of the framework's implementation within the context of a complete navigational system.

1.4.2 Hallway navigation results (Chapter 5)

Chapter 5 presents an analysis of the COMPANION framework for simple hallway navigation scenarios. The analysis is composed of two main sections: results from simulation (Section 5.1) and results from a user study on a physical robot (Section 5.2).

In simulation, we demonstrate many different scenarios of the robot navigating to various goals in the presence of people. We discuss how the resulting behaviors follow human social norms, and we further describe how the behaviors can be altered to produce different social "personalities," such as extremely deferential (always moving to the right out of a person's way) or more aggressive (continuing to face a person while passing).

We describe a user study designed to understand the behavior of the COMPANION framework on a physical robot, in a controlled hallway navigation task. Twenty-seven participants walked past a robot while it either attempted to observe social norms or merely avoided hitting the person. We show that people rated the robot as having more socially appropriate movement when it attempted to observe social norms, including tending to the right of the hallway and respecting their

personal space. However, we also note that people felt that the robot was overly deferential in its method of avoiding them, and we discuss ways to change the robot's behavior to make it be more (or less) social.

1.4.3 Side-by-side results (Chapter 6)

In Chapter 6, we extend the COMPANION framework for *joint* human–robot path-planning. We focus on the task of escorting someone while traveling side-by-side. We describe the necessary modifications to the COMPANION framework to plan joint paths in general as well as the specific constraints needed to perform the escorting task.

The extension to the COMPANION framework is based on our argument that generating a joint plan for both the robot and the person will produce robot behavior that smoothly adheres to the social conventions of tasks such as side-by-side escorting. To extend the framework to such joint activities, we define the concepts of *joint goals*, *joint actions*, and *joint constraints*, all of which we argue are necessary for planning paths for both a robot and a person.

For the specific case of side-by-side escorting, we define two additional constraints: remaining near a person and keeping a preferred angle to that person. Finally, we present the results from several simulated scenarios that demonstrate the resulting escorting behavior, including behaviors around corners and through chokepoints. We argue that these joint plans produce socially appropriate escorting behavior.

1.4.4 The Companion robot (Chapter 7)

The final contribution of this thesis is a new platform for social robot research: the Companion robot (see Figure 1.1). Chapter 7 discusses the rationale for developing a new robot as well as the details of the robot's design.

Our research on social navigation and the COMPANION framework has indicated the importance of sideways maneuvers, such as the human behavior of sidestepping around obstacles. In contrast to most robots used in human–robot interaction research, the Companion robot is based on a holonomic platform, which allows it to move sideways without having to turn first. Furthermore, we detail the design of the robot's body, which is intended to better support human–robot social interaction by giving the robot a more "friendly" appearance. The robot's head displays a graphical face to provide a focus for face-to-face interactions. We expect Companion to be a versatile platform for future social robotics research.

Figure 1.1: The Companion robot.

This chapter also discusses the interdisciplinary nature of the design of Companion. The author's contribution to the robot is that of team leader, both driving the design effort and making key design decisions.

1.5 Summary

This thesis presents a case for mobile robots that behave according to human social norms. We detail the framework we developed, COMPANION, which provides a method for representing human social norms as constraints on a robot's path planning and navigation. We present an evaluation of the system for the task of hallway navigation, with results demonstrated both in simulation and in a user study. Furthermore, we extend the COMPANION framework to the joint task of side-by-side escorting, and we present results from simulations. Finally, we present a new robotic platform for use in similar social human–robot interaction research, the Companion robot. As a whole, this thesis demonstrates both a need for, and an

implementation and evaluation of, robots that navigate around people according to social norms.

Chapter 2

Related Work

The overall goal of this research is to create robots that interact with people in socially acceptable ways. As such, this thesis draws on work from many fields, including human and social psychology, robot navigation, and human–robot collaboration. Here we present some of the most relevant research.

2.1 Human social navigation

As we are interested in *social* human–robot interaction, one key aspect is how people behave. We therefore draw on research from the fields of psychology and sociology, which we can then extend to robot behavior.

When two people walk together, they coordinate their movements with each other while observing many social conventions, such as what distance to keep from each other and how to indicate when to turn or stop. Despite the complexity of such interpersonal coordination, very little research has been done to determine exactly what people do and what social conventions they follow (Ducourant et al., 2005; Marsh et al., 2006).

One aspect of social conventions for spatial interaction that has been widely studied is the idea of personal space, or *proxemics* (Hall, 1966, 1974; Mishra, 1983; Aiello, 1987; Burgoon et al., 1989). According to Hall (1966), people maintain different culturally defined interpersonal distances from each other, depending on the type of interaction and the relationship between the people. Specifically, Hall differentiated between four different "zones" as follows:

- *Intimate*: from close physical contact to about 0.5 m apart
- *Personal*: friendly interaction at "arm's length," 0.5–1 m

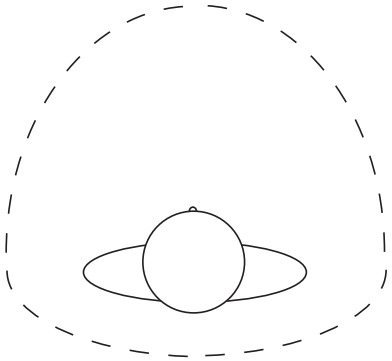


Figure 2.1: The approximate shape of a person's personal space. Frontal distance is the greatest, while rear distance is smallest.

- *Social*: business interaction, 1–4 m
- *Public*: speaking to a crowd, more than 4 m away

For this thesis, we are particularly interested in the zone of *personal space*, as it is a culturally defined zone of "spatial insulation" that people maintain around themselves and others (Burgoon et al., 1989). Research has indicated that the shape of personal space is asymmetric for both approach distances (Ashton and Shaw, 1980) and standing in line (Nakauchi and Simmons, 2000); the approximate shape of personal space is shown in Figure 2.1. The exact size of personal space is not constant and differs across cultures and familiarity groups (Baxter, 1970; Burgess, 1983). Furthermore, the size and shape of personal space changes based on walking speed, foreknowledge of obstacles' movements, and other mental tasks being performed while walking (Gérin-Lajoie et al., 2005). In addition, the size of personal space tends to be smaller between people performing a cooperative task than a competitive one (Burgoon et al., 1989). The violation of personal space leads to discomfort and misunderstandings (Watson, 1970). Because personal space is such an important aspect of how people interact with each other, we take it to be one of the primary social conventions that a robot should respect when interacting with people.

When people interact with each other, the interaction may take the form of an intentional, focused interaction, such as a conversation, or it may be non-intimate, or even adversarial. Examples of non-intimate social interactions include how crowds gather (McPhail and Wohlstein, 1986) and what social conventions are used when pedestrians pass each other, such as moving to one side, smiling, or ignoring the passing person (Wolfinger, 1995; Patterson et al., 2002). In America, as well as

many other Western cultures, the typical convention employed when passing others is to walk on the right side of a hallway or sidewalk (Whyte, 1988; Bitgood and Dukes, 2006). As with personal space, we have identified this "pass on the right" tendency as a key social convention that a robot should also respect. Most of these studies investigated these interactions from an individual's standpoint, rather than attempting to understand joint actions between two or more people. In contrast, Ducourant et al. (2005) studied how people attempt to break or maintain interpersonal distance when facing each other in an adversarial situation. This work verified that people mutually influence each other in such a situation, which we believe also applies to non-adversarial interactions.

Most studies of human interaction, however, have examined focused interactions, when two or more people intentionally engage in face-to-face conversations. For example, Kendon and Ferber (1990) studied the individual actions people perform when greeting one another, such as waving or nodding one's head. Studies of people involved in conversation have shown that conversational partners unintentionally become entrained and mimic each others' posture (Shockley et al., 2003; Richardson et al., 2005). A large body of research has examined how people form and maintain *common ground*, the shared knowledge and suppositions between conversational partners (Clark and Brennan, 1991; Clark, 1996). Clark (1996) divides common ground into three components:

- *Initial common ground.* Initial common ground includes all of the background facts and assumptions made by the participants. Knowledge of societal conventions fall into this category.
- *Current state of the joint activity.* Each participant has a mental representation of the state of the conversation, including what each believes the other knows.
- *Public events so far.* This category includes both historical events that each partner may be expected to know as well as the history of their joint activity up to the current state.

While the idea of common ground has traditionally been applied only to conversational interactions, recent research has begun to extend the concept to any type of joint activity. In particular, successful joint activity requires not only common ground, but also the ability to predict and to direct the other's actions (Klein et al., 2005; Sebanz et al., 2006). Additional research has provided insight into how people automatically predict many aspects of what others are going to do (Frith and Frith, 2006). We extend common ground theory to human–robot interactions; in particular, we argue that a robot's physical behavior can be used to build common ground with people by demonstrating a shared knowledge of social conventions.

A final related area of psychology research is on efficiency, in terms of both economy of movement and collaborative effort. People are remarkably efficient at minimizing energy expenditure in their physical actions (Sparrow and Newell, 1998). For example, studies conducted in malls indicate that people determine which side of the corridor they walk on and which direction they turn at intersections based on minimizing the required number of steps they must take (Bitgood and Dukes, 2006). Even in conversation people attempt to minimize effort, as described by the Principle of Least Collaborative Effort:

> *The principle of least collaborative effort*: In conversation, the participants try to minimize their collaborative effort—the work that both do from the initiation of each contribution to its mutual acceptance. (Clark and Brennan, 1991, p.226)

Klein et al. (2005) extends this concept by arguing that any time two people begin a collaborative process they not only attempt to minimize their joint effort but also enter a "basic compact," a tacit agreement that they will do so. Each person can thus assume that the other will put forth the required effort to collaborate. We apply this idea to people who are walking together: once two people begin to walk with the intention of walking together, each can assume that the other will continually take the necessary actions to maintain their partnership. Gilbert (1990) terms the state of such people a "plural subject" to indicate their mutual obligations to each other.

2.2 Social robot planning and navigation

A second area of research related to this thesis is the topic of robot navigation, both for local and global obstacle avoidance, and for specific tasks involving human interaction.

2.2.1 Local obstacle avoidance

A mobile robot must be able to avoid obstacles in its environment, and many different algorithms for obstacle avoidance have been developed. Many times, unexpected obstacles (e.g., obstacles not appearing in a map) are handled only in a locally reactive manner. Traditional algorithms for local obstacle avoidance include the Artificial Potential Field method (Khatib, 1986) and its extension, the Vector Histogram approach (Borenstein and Koren, 1989). In both of these methods, obstacles exert a virtual force on the robot, which allows the robot to avoid collisions. However, these methods treat all unexpected obstacles as static (non-moving), and

they do not account for vehicle dynamics. Obstacle avoidance algorithms that do account for vehicle dynamics, such as how quickly the robot can accelerate or decelerate, include the Dynamic Window Approach (Fox et al., 1997), the Curvature Velocity Method (Simmons, 1996), the Lane-Curvature Method (Ko and Simmons, 1998), and LaValle and Kuffner's randomized kinodynamic planning using Rapidly-exploring Random Trees (RRTs) (1999). Algorithms that account for obstacles that may be moving over time include the Velocity Obstacle approach (Fiorini and Shiller, 1998); Partial Motion Planning (Laugier et al., 2005), which uses RRTs to find the best partial path to a goal within some time period; and Reflective Navigation (Kluge, 2003). Finally, several approaches consider both vehicle dynamics and dynamic obstacles, including Castro et al.'s use of Velocity Obstacles within the robot's Dynamic Window (2002), Foka and Trahanias's predictive navigation (2003), and Owen and Montano's planning in velocity space (2005). A summary of these methods is shown in Table 2.1. While any of these algorithms can be used to produce varying degrees of safe and effective obstacle avoidance, none of them explicitly account for the pre-established social conventions that people use when moving around each other. Furthermore, such local avoidance behaviors do not account for global goals, and thus often produce globally sub-optimal behavior.

2.2.2 Global planning

Global path planners are used to determine possible paths through a known environment, and generally operate independently of local obstacle avoidance. Two main types of planners are currently used: heuristic search algorithms and randomized planners. Heuristic search algorithms, most notably A* (Hart et al., 1968), can find optimal paths, but typically do not run fast enough to replan in real time, as the robot receives new sensory data. Many variations on A* exist in order to improve replanning time, typically by saving and reusing portions of the search tree. Lifelong Planning A* (LPA*) (Koenig et al., 2004) can rapidly replan when the environment changes, but only when planning from the same start state, and thus cannot be used for a moving robot. The replanning algorithms D* (Stentz, 1994) and D* Lite (Koenig and Likhachev, 2002) allow the start state to change, but do so by planning in reverse—from the goal state to the robot's current position. This works in many cases, but not with dynamic obstacles—the robot has no way of knowing where the dynamic obstacles will be at the time it reaches the goal, so the state of the world when the robot reaches the goal is unknown.

In contrast, Real-Time Adaptive A* (RTAA*) (Koenig and Likhachev, 2006) and Generalized Adaptive A* (GAA*) (Sun et al., 2008) both plan forward, from start to goal, and allow for changing action costs. RTAA* handles only increasing

Algorithm	Reference	Vehicle Dynamics	Dynamic Obstacles	Social Conventions
Artificial Potential Field	Khatib (1986)	No	No	None
Vector Histograms	Borenstein and Koren (1989)	No	No	None
Dynamic Window (DW)	Fox et al. (1997)	Yes	No	None
Curvature Velocity Method (CVM)	Simmons (1996)	Yes	No	None
Lane-Curvature Method (LCM)	Ko and Simmons (1998)	Yes	No	None
Randomized Kinodynamic Planning	LaValle and Kuffner (1999)	Yes	No	None
Velocity Obstacle (VO)	Fiorini and Shiller (1998)	No	Yes	None
Partial Motion Planning	Laugier et al. (2005)	Yes	Yes	None
Reflective Navigation	Kluge (2003)	No	Yes	None
Combined DW and VO	Castro et al. (2002)	Yes	Yes	None
Predictive Navigation	Foka and Trahanias (2003)	Yes	Yes	None
Velocity Space Planning	Owen and Montano (2005)	Yes	Yes	None
Modified LCM	Olivera and Simmons (2002)	Yes	No	Passing on the right
Person Passage	Pacchierotti et al. (2005a,b)	No	Yes	Passing on the right
Human-Aware Navigation	Sisbot et al. (2006)	No	No	Visibility to people
Line-standing	Nakauchi and Simmons (2000)	No	No	Standing in line
Dynamical systems	Althaus et al. (2004)	No	No	Entering a group

Table 2.1: Summary of some relevant robot navigational algorithms, including whether they explicitly account for vehicle dynamics or dynamic obstacles as well as what social conventions they implement.

costs, such as an obstacle perceived where the previous search assumed free space, and thus it is unable to handle dynamic obstacles. GAA*, in contrast, allows for action costs to increase or decrease; however, it typically performs worse than A* when a large number of costs change between searches (Sun et al., 2008).

A different approach to real-time replanning uses randomization, rather than exhaustive search. One common approach is to use Rapidly-exploring Random Trees (RRTs) (LaValle, 1998; LaValle and Kuffner, 1999), which are designed to explore the environment quickly. RRTs typically find *some* path to the goal, but not necessarily an *optimal* path. Methods exist to bias RRTs heuristically to find the goal state more rapidly and partially account for path cost (e.g., Urmson and Simmons, 2003). However, despite biasing, RRTs do not find smooth or optimal paths. While the generated paths can be post-processed to yield smoother paths, doing so may eliminate legitimate avoidance maneuvers around moving obstacles. Because we believe that people take optimal paths whenever possible, we reject such probabilistic planners in favor of A*.

An alternate method of improving search speed and results involves modifying the search space, such as done in Quadtree and Framed-Quadtree planners (Yahja et al., 1998), as well as other planners that use quadtree-like hierarchical decompositions of space (Fujimura and Samet, 1989). Quadtrees are irregularly-sized grids formed by recursively subdividing regions into four quadrants until each region is either free of obstacles or is the smallest allowed resolution. In sparse maps, quadtrees reduce the memory requirements (and thus search time) over regular grids. However, paths found with quadtrees are usually sub-optimal as compared to regular grids, particularly in large areas of free space. Framed Quadtrees create more optimal paths by modifying the quadtree data structure, but at the expense of greater memory requirements. In particular, framed quadtrees perform poorly when the world is generally known in advance. However, in our work, we typically assume that the robot has access to a map of the environment. To improve search speed, we use a variable grid that does not rely on the environmental structure; this is described in Chapter 4.

2.2.3 Social navigation

A number of methods have been developed to allow robots to navigate around people in specific, typically non-generalizable, tasks. Some of these tasks include tending toward the right side of a hallway, particularly when passing people (Olivera and Simmons, 2002; Pacchierotti et al., 2005a,b), standing in line (Nakauchi and Simmons, 2000), and approaching people to join conversational groups (Althaus et al., 2004). Museum tour guide robots are often given the capability to detect and attempt to handle people who are blocking their paths (Burgard et al.,

1999; Thrun et al., 1999). Algorithms developed for the robot Grace allowed it to navigate a conference hall, ride an elevator, and stand in line to register for a conference (Simmons et al., 2003). Prassler et al. (2002) demonstrated a robotic wheelchair that can follow next to a person, but their method does not account for social cues the human might use nor allow for any social interaction. Sviestins et al. (2007) have begun investigating how a robot might adapt its speed when traveling next to a person, but they have obtained mixed results even in a controlled laboratory setting. In contrast, this thesis presents a generalized framework for integrating multiple social conventions into a robot's behavior, thus producing more natural and understandable robot movement.

Several groups have begun to address questions relating to planning complete paths around people, rather than relying on solely reactive behaviors. Shi et al. (2008) discusses a method for a robot to change its velocity near people. While this method begins to address ideas of planning around people, it does not directly consider social conventions. In contrast, the Human–Aware Motion Planner (HAMP) (Sisbot et al., 2007) considers the safety and reliability of the robot's movement as well as "human comfort," which attempts to keep the robot in front of people and visible at all times. However, the paths that the planner generates may be very unnatural due to its attempts to stay visible to people. In contrast, we are proposing a more general framework for representing spatial social tasks, which we believe will allow our work to address a wider range of social situations. Furthermore, we focus on behaviors that are not only aware of people but also socially acceptable to people.

2.2.4 People tracking

Extending any navigational algorithm to account for people requires the ability to identify and track which sensor readings correspond to people. Many different ways of identifying and tracking people have been proposed, including using vision for color-blob tracking (Schlegel et al., 1998), using vision to track faces (Sidenbladh et al., 1999), and various laser-based methods (Castro et al., 2004; Kluge et al., 2001b; Schulz et al., 2003; Topp and Christensen, 2005; Cui et al., 2006), including our own particle-filter-based technique (Gockley et al., 2007). All of these methods have various benefits and shortcomings. All camera-based methods suffer when exposed to variable lighting conditions, and face-tracking methods work only when the person is facing the robot, which is not necessarily the case in social human–robot interaction. Methods that use a laser rangefinder typically cannot accurately differentiate between people and other objects. Perhaps more promising are multi-sensor methods that combine information from both a camera and a laser rangefinder (Kleinehagenbrock et al., 2002; Kobilarov et al., 2006; Michalowski

and Simmons, 2006). Other researchers have investigated the use of radio tags to track people (Bianco et al., 2003; Kanda et al., 2003), but these methods do not provide very accurate position information and also require instrumenting the person with sensors. However, the focus of this thesis is not on the sensing problem as such, and so we will rely as much as possible on these existing identification and tracking methods.

An additional aspect of understanding how a robot should navigate around people involves learning people's behaviors. Several approaches have been proposed, though the typical method is to use off-line learning techniques to build a map of "common" destination points for people in the environment, and then use this map to augment both on-line person-tracking and navigation (e.g., Bennewitz et al., 2003, 2005; Bruce and Gordon, 2004; Foka, 2005; Kanda et al., 2009; Ziebart et al., 2009). While we do not currently implement any of these methods, a complete robotic system may greatly benefit from the better person-prediction these methods afford.

2.3 Social human–robot collaboration

Since we are interested in socially collaborative tasks, such as walking together side-by-side, a final related area of research is the field of human–robot collaboration. However, unlike our work, human–robot collaboration research has typically focused on conversational-type interaction with a stationary robot (e.g., Trafton et al., 2005; Hoffman and Breazeal, 2004; Sidner and Dzikovska, 2002). Ikeura et al. (1994) investigated cooperative object manipulation between a person and a robotic arm, and found that for that type of physical collaboration people preferred for the robot to behave in a human-like manner.

Forming common ground between a person and a robot can also be viewed as a collaborative activity. As with the human psychological literature, much of the work on forming common ground in human–robot interaction focuses on conversational dialog (e.g., Powers et al., 2005; Li et al., 2006). Our own prior work shows that some degree of common ground can be formed through the robot's use of emotional expressions (Gockley et al., 2006). Stubbs et al. (2006) describes how problems in grounding between people and robots can hinder human–robot collaboration, reinforcing the need for successful grounding processes. Finally, Klein et al. (2005) discusses some of the steps necessary to extend the idea of common ground theory to joint activity between agents, such as creating agents that act predictably and signal their intentions.

Other research on socially collaborative robots includes areas such as shared attention based on gaze tracking (Kozima et al., 2003; Yamato et al., 2004) and

perspective-taking (Trafton et al., 2005). While either of these aspects of interaction may be necessary for a fully competent social system, they are not a focus of this thesis. Somewhat more relevant is research into behavior recognition, particularly regarding typical behaviors in public environments (Kluge et al., 2001a). For example, a robot that is traveling side-by-side with a person may need to differentiate between behaviors such as the person stopping to talk with a friend versus stopping because he is feeling ill. However, such recognition is beyond the scope of this research.

2.4 Summary

Many research areas are relevant to social robot navigation. Essential to our approach is research on how people interact with each other, particularly when walking. We use the descriptions of human social conventions as a basis for our implementations of social robot behaviors. Furthermore, we use the theory of common ground to provide rationale for making robots behave in human-like ways. Work on robot navigation demonstrates the lack of research in having robots react to people as social entities, rather than inanimate obstacles. While the HAMP architecture begins to address this idea, it assumes that people will be wary of the robot, and thus requires the robot to remain "visible" when navigating around people. In contrast, we argue that the robot should behave in a *social* manner, which will allow people to understand it implicitly. Finally, work in social human–robot collaboration reinforces the idea that common ground can be formed through a robot's behaviors, and that grounding is necessary for collaboration—including traveling with people—to occur smoothly.

Chapter 3

Background and Preparatory Work

This thesis will present the COMPANION framework for person–acceptable robot navigation. However, several studies we performed prior to developing the COMPANION framework served as a foundation for the research. In particular, we designed a laser-based person-tracking system, analyzed two person-following behaviors for a mobile robot, and studied how people walk in pairs. These studies are described below.

3.1 Person tracking

For a robot to behave socially around people, it must be able to track people who may be moving (or stationary) in unknown, potentially dynamic, indoor environments. While our tracker is similar to that of Topp and Christensen (2005), we present the details of our particular implementation here. Briefly, each scan from the laser is segmented into person-sized blobs, which are tracked using individual particle filters (Arulampalam et al., 2002) for each blob. The basic algorithm we use is as follows:

1. Since the robot—and hence the laser—may be moving, the particles being tracked are first transformed into the robot's current frame of reference. Updating the old information into the new frame is preferable to working in absolute coordinates, as odometry errors are not compounded over time.

2. The laser scan is next divided into segments. Adjacent points in the scan are considered part of the same segment if they are less than 10 cm apart.

3. Segments that contain any points further away than some threshold for tracking (we use 3.5 m) are discarded.

4. Segments with a width (straight-line distance between the two endpoints) that is greater than 60 cm or less than 5 cm are discarded, as such measurements are unlikely to correspond to people.

5. Remaining segments that are greater than 20 cm are classified as a potential person. Smaller segments may be individual legs, and so we perform rudimentary clustering of these potential legs. If two such "leg" segments are separated by less than 40 cm, they are classified as a single person. If no second leg is close enough to some segment, that segment is considered a potential person by itself.

6. All potential persons are tracked with a standard particle filter algorithm, using one filter for each person and 100 particles per filter. We use a Brownian (random) model of movement to predict where each segment might travel, as we found that any more sophisticated motion model could not account as well for a person's sudden stops or turns. Each filter is assigned to the closest potential person within 40 cm of the filter's center, and a new filter is created for any potential person that is more than 40 cm away from any unassigned filter.

7. Filters may be unassigned for up to 5 cycles of the tracker, after which they are removed. Allowing filters to remain unassigned helps to account for short occlusions, such as a person walking quickly past the person or object being tracked.

This tracking method, unlike most vision-based trackers (which typically track faces; see Section 2.2.4), is relatively robust to the person's orientation; people can be tracked walking toward, away from, or past the robot. As such, this method can be used to track people in front of the robot, for following behind them (as discussed below), or to track people next to the robot, for side-by-side accompaniment or escorting.

An example laser scan with identified objects marked is shown in Figure 3.1. Note that this method of tracking identifies any "person-sized" objects as people, including objects such as chairs and garbage cans. However, without the use of additional sensors, such as vision, differentiating between a stationary person and similarly shaped inanimate objects is nearly impossible. Since we wish to track even people who are not moving, we chose to allow the tracker to identify other objects as people. Truly social interaction with people will require a more robust method that can distinguish people from inanimate objects; however, we currently

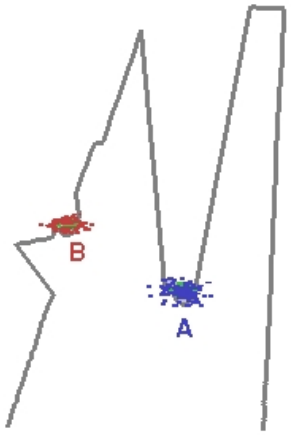


Figure 3.1: A sample scan from the laser range-finder, taken in a hallway, and overlaid with samples from the person tracker. The center-most samples (labeled `A`) correspond to the person being tracked and followed; the leftmost samples (labeled `B`) correspond to clutter in a doorway.

favor false positives (identifying something inanimate as a person) over false negatives (failing to track an actual person).

3.2 Person-following

As a first step toward developing robots that can accompany people in socially acceptable ways, we investigated social perceptions of a robot's movement as it followed behind a person (Figure 3.2), as a social assistant robot might when passing through doorways or navigating around obstacles. We designed and tested two modes of person-following to determine which is more natural and socially acceptable. Participants in a pilot study agreed that the robot's behavior was more human-like when the robot always drove toward the person (i.e., in the direction of the person's current location), rather than when it followed the person's exact path. Furthermore, this "direction-following" method was rated as better matching people's expectations for the robot's behavior. This finding demonstrates that people may expect robots to behave according to human social conventions, such as minimizing travel distance. This study is described below and can also be found in Gockley et al. (2007).

Figure 3.2: The robot, Grace, following a person down a hallway.

3.2.1 Design approach

We designed this study to investigate social behaviors for robots that allow people to feel comfortable in the robot's presence and understand the robot's intentions. The factors we considered in designing person-following behaviors for our robot included:

- **Personal space**: People determine how close they should be to one another according to societal conventions regarding personal space (Hall, 1966). The robot should always remain at a socially appropriate distance.
- **Reliability**: The robot and its sensors must be capable of tracking a person with a high degree of reliability in order to remain useful and not frustrate the person.
- **Safety**: The robot must ensure the person's safety at all times; in particular, the robot must maintain enough space between itself and the person so as to avoid collisions.

Finally, we considered the robot's **human-likeness**. In particular, we asked: to what extent should the robot's behavior match that of a human in the same situation? While it is clearly desirable for the robot to behave according to people's expectations, people may not expect a machine-like robot to act according to social

Figure 3.3: The LCD screen with graphical face used on the robot, shown here with a speech bubble that echos what the robot says ("Keep going!").

conventions. For this study, we designed the robot's behaviors to test this "human-likeness" factor.

3.2.2 Hardware

Our research platform for this work was Grace (Simmons et al., 2003), an RWI B21 base with an LCD "head" mounted on top, as shown in Figure 3.2. With the head, the robot is roughly human-height. The robot uses one primary sensor, a SICK LMS200 scanning laser range-finder, mounted approximately 40 cm above the ground. The robot can move at speeds of up to 90 cm/s, but we tend to limit the speed to no more than 70 cm/s due to safety concerns.

The robot's LCD screen is used to display an expressive, graphical face (Figure 3.3), which has been shown to encourage human interaction with the robot (Bruce et al., 2002). The robot is capable of speech via a synthesized voice and a text-to-speech system. The robot's face automatically lip-syncs with the speech.

In order for Grace to track people, we used the tracking method described above (Section 3.1).

3.2.3 Different person-following behaviors

To test different levels of "human-likeness" in person-following, we designed and evaluated two robot behaviors. The simplest method is to have the robot always attempt to drive directly toward the person's location. From general observations, we suspect that this is how people most often follow other people. This method often results in the follower cutting corners and generally not following in the exact footsteps of the leader. The second method, then, is to have the robot attempt to follow the exact path that the person took. While this method may not be the most human-like method, we hypothesized that it may better match people's ex-

pectations for a machine-like robot. For example, if a person is leading a robot somewhere, any step in the person's path may be taken for reasons that the robot does not know (such as avoiding obstacles the robot is unable to sense), and thus following the person's exact path may be the more appropriate behavior. Using the person-tracker described above, we have implemented both of these methods.

In both methods, the robot begins to follow a person as soon as someone is detected within 125 cm of the robot, in a cone of ± 0.5 radians, as measured from the average location of a particle filter's samples. The robot then attempts to remain a constant distance (120 cm, ± 10 cm) from the tracked person. This is achieved through a simple feedback control loop based on two factors. First, a proportional controller works to minimize the error between the robot's current distance from the person and its desired position. Secondly, the change in range error over time is used to reduce oscillations. Specifically, if the robot begins to fall further behind the person (i.e., the range error is increasing), then the robot's velocity is increased based on the error; if the robot is too close and getting closer, then the velocity is similarly decreased. The robot stops if the distance to the person drops below 90 cm. The robot's maximum velocity is capped at 70 cm/s, due to safety concerns.

For this work, the distance at which the robot tried to follow is held constant, and is designed to keep the robot just outside of one's personal space. As several studies have found, the appropriate distance may vary according to an individual's personality traits (Walters et al., 2005; Gockley and Matarić, 2006). We chose the value of 120 cm as a comfortable distance for the experimenter; we have not tested the person-following with different distances at this time.

The two person-following methods differ in how they select the robot's direction of travel. These differences, as well as social aspects of the robot's behavior, are discussed in the following sections.

Direction-following

In this method of following a person, the robot simply attempts to drive in the direction of the tracked person's current position. This is combined with the underlying obstacle avoidance control system by setting these goal directions using the Curvature-Velocity Method (CVM) (Simmons, 1996). With this method, the robot is able to follow the person through doorways and around corners without collisions.

It is interesting to note that person-following and most obstacle avoidance methods are fundamentally at odds, since following a person requires the robot to drive straight toward something that would normally be interpreted as an obstacle. To convince the Curvature-Velocity method to follow a person, we weight the CVM parameters to strongly favor the goal direction over the preferred dis-

tance from obstacles and the preferred maximum speed. That is, the robot will favor going slowly close to obstacles (such as the person) as long as its heading is correct. However, this trade-off is not ideal, and we address the need for integrating obstacle avoidance and social conventions with our COMPANION framework (Chapter 4).

Path-following

In this more sophisticated approach, the robot attempts to follow the path that the person took as closely as possible, such as switching to the opposite side of the hallway at a certain location and driving around corners with the same curvature as the person's travel. Path-following is achieved in much the same way as direction-following, except that the robot's goal direction is chosen according to the Pure Pursuit path-following algorithm (Coulter, 1992). At each tracker cycle, the person's location is stored, building a history of the person's path. The robot's goal point is selected as the point at which the person was at the desired distance from their current position (that is, 120 cm behind the person). As with direction-following, the CVM method is used to integrate obstacle avoidance with the person-following behavior.

In addition, the robot's goal direction is constrained such that the robot will never intentionally turn to a point at which it can no longer track the person. This is necessary because the robot does not have a full 360-degree sensor coverage, but means that the robot may not always follow the person's exact path, particularly if the person walks in a tight circle around the robot.

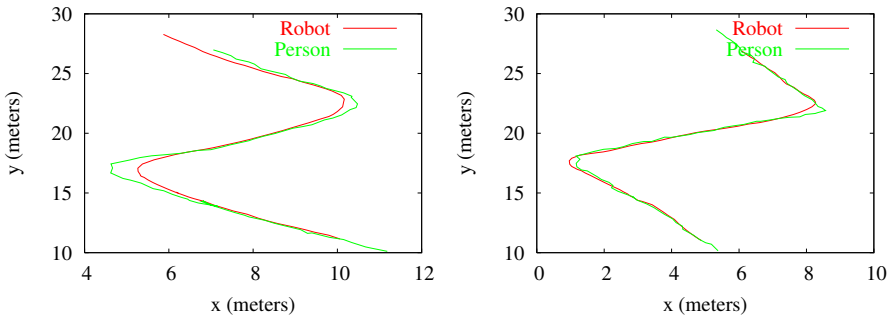
3.2.4 Performance

The two person-following algorithms differ most noticeably when guiding the robot around corners: the direction-following approach results in the robot rounding corners much more so than the person does, whereas the robot explicitly attempts to follow the same curvature as the person when using the path-following approach. This distinction can be seen in Figure 3.4.

We present here the results from several trial runs with each person-following algorithm. Trials were performed both at the robot's maximum speed and at slower speeds.

Procedure

All trials took place in office building hallways, with varying amounts of clutter. While other people occasionally passed by the robot, no occlusions were permitted



(a) Direction-following (b) Path-following

Figure 3.4: Paths of the person and the robot around corners, for each of the two approaches. The robot drastically cuts corners when not following the person's exact path. Note that each path shown is roughly 15 m in length.

between the robot and the person it was following. In order to analyze the technical performance of the robot, a single person led the robot for all trials. Care was taken to make the person's behavior as consistent as possible across trial runs, though obviously no two runs with either algorithm were identical. The person had prior knowledge of the robot's person-following behavior and did not attempt to "trick" the robot with sudden changes in movement patterns.

Results

Four trials with each approach were run at relatively high speeds. Each approach was run for a total of about 30 minutes (5-10 minutes per trial) and covered a total traversal of over 1 kilometer, with average speeds of close to the maximum allowed speed of the robot, 70 cm/s. On average, the robot was able to track the person over a distance of about 30 meters (1 minute) before an error in tracking occurred. Both approaches had a furthest distance between tracking errors of over 160 meters (over 3.5 minutes). Using an analysis of variance (ANOVA), no significant difference was found between the two approaches in terms of the distance or time between tracking errors (distance $F[1, 65] = 0.79, p = 0.3$; time $F[1, 65] = 0.27, p = 0.6$). In all cases of tracker failure, the robot was able to re-acquire the person within moments of the person re-approaching the robot.

The tracking performed considerably better when the robot moves more slowly. When the person traveled at speeds of about 45 cm/s, the robot was able to follow for over 320 meters (over 10 minutes) before tracker failure, using each of the methods. During these slower traversals, the robot was able to remain approxi-

mately 130 cm (SD 25 cm) from the person (recall that the goal distance was 120 cm). Again, both person-following behaviors performed equivalently from a technical perspective.

3.2.5 User acceptance

While the above results demonstrate the technical performance of the robot, one of our main interests is in how people *perceive* the different robot behaviors. We performed a pilot study to explore whether people preferred one person-following method over another.

Procedure

This study was performed during an informal gathering and had 10 adult participants (8 males, 2 females), including students, staff, and faculty members. Though all participants were experienced with robotics, few had advance knowledge of the nature of the study. Participants were asked to observe the robot's behavior as it followed the experimenter around the lab for several minutes. They then answered a short questionnaire on the robot's behavior, including whether the behavior met their expectations, how natural the behavior was, and how appropriate the robot's following and stopping distances were. This process was done once for each of the two person-following algorithms. To ensure that all participants viewed identical behaviors of the robot, all participants viewed both conditions as a single group; as such, conditions were not counter-balanced.

The robot followed the experimenter for about 50 m with each behavior. On average, the robot remained approximately 1.5 m from the experimenter during each trial. The experimenter stopped and started several times during each person-following method, so that participants could observe the robot's behavior both while moving and while stopped.

Results

Due to the within-subjects nature of this study, we analyzed the survey responses using paired t-tests across trials. Average responses and t-values for each question are given in Table 3.1.

Although the two behaviors were very similar, participants noticed their differences. Participants were asked to rate the robot's behavior according to whether it met their expectations, on a scale of 1 ("not at all") to 7 ("very much"), and how natural the behavior was, on a scale of 1 ("not at all" to 7 ("human-like"). As shown in Table 3.1, participants rated the robot's behavior as significantly more natural

Table 3.1: Average responses to the survey questions, with standard deviations given in parentheses. All questions were asked on scales of 1–7. $N = 10$.

	Following Behavior		Paired
Question	Direction	Path	t
Met expectations (not at all—very much)	5.0 (0.94)	3.7 (0.95)	-4.33*
Natural (not at all—human-like)	4.0 (1.15)	2.9 (0.88)	-3.97*
Following distance (too close—too far)	3.0 (0.94)	2.9 (1.29)	-0.43
Stopping distance (too close—too far)	4.9 (1.20)	5.4 (1.71)	1.86

* significant at $p < 0.01$

and human-like in the direction-following condition. In addition, participants felt that the direction-following robot behaved more according to their expectations. Notably, none of the participants rated the path-following behavior as better than the direction-following behavior on either of these questions. Furthermore, the answers to these two questions were highly correlated ($r = 0.80$, $p < 0.0001$), indicating that participants expected the robot's behavior to be human-like, despite the robot's non-anthropomorphic shape.

Participants were also asked whether the robot followed and stopped at appropriate distances from the experimenter. They rated the distances on a scale of 1 ("too far away") to 7 ("too close"). Overall, participants felt that the robot stayed a little too far away from the experimenter while moving (overall mean 2.95, SD 1.10), but stopped at an appropriate distance (overall mean 5.15, SD 1.46). There were no significant differences in participants answers across the two person-following behaviors.

3.2.6 Discussion

Quantitatively, the two methods of person-following were equivalent: the behaviors did not differ in laser tracking performance, and both allowed the robot to follow smoothly behind a person. The primary difference between the two behaviors occurred at corners, when the direction-following behavior caused the robot to curve much more gently than the person, given sufficient space to do so. If the following were to occur in narrow corridors, the differences between the behaviors would lessen, as obstacle avoidance would constrain the robot's movement.

Qualitatively, however, people indicated that the direction-following behavior was significantly more human-like and more closely matched their expectations than when the robot follows the person's path. Several participants commented that, when performing path-following, the robot did not appear to react "quickly enough" to the person's turns (since the robot turned at the location where the person turned, rather than at the same time as the person), which may help explain this finding.

To date, we have performed only the small pilot study as described above. Obviously, many caveats apply to this sort of study, as the participants were all familiar with robots and most likely had very different expectations of robotic behavior than non-roboticists. However, while the participant population may have influenced the exact values of the survey questions, we expect that the relative differences between the different robot behaviors would be similar across other populations, as well. A further shortcoming of this study is that it analyzed only people's third-person observations of the robot's behaviors, and thus the results may not capture the full spectrum of people's preferences. However, while people's in situ experiences of a robot's behavior are clearly valuable, such testing is difficult to perform and evaluate when the robot's behavior occurs strictly behind—and thus out of sight of—the person.

3.2.7 Summary

This study focused on one particular aspect of human–robot social interaction: a robot that follows behind a person. By implementing two different methods of person-following, we were able to analyze the robot's behavior in both a quantitative and a qualitative way. We found that when the robot followed the person more loosely, cutting corners when possible, observers of the robot felt that it behaved more human-like and, importantly, better matched their expectations. From this, we form the more general hypothesis that robots should follow human social norms for navigation, both when following behind a person (for which the norm seems to be minimizing the distance to the person, rather than keeping to the person's fixed path) and for other, more general, human–robot encounters.

3.3 Social aspects of walking together

From our research on person-following, we believe that robots should observe human social norms for navigation. However, as discussed in Section 2.1, many of these norms are poorly understood. For example, when two people walk together, they coordinate their movements with each other while observing many social con-

ventions, such as what distance to keep from each other and how to indicate when to turn or stop. Furthermore, if either partner fails to use or respond to such conventions, the interaction becomes difficult and awkward. Despite the complexity of such interpersonal coordination, extremely little research has been done to determine exactly what people do and what social conventions they follow (Ducourant et al., 2005). To provide a basis for our research, we performed an observational study of how older adults in a local retirement community walk together, using ethnographic methodologies borrowed from social anthropology. These results can also be found in Gockley (2007).

3.3.1 Procedure

Observations took place at a local retirement community. Investigators used an ethnographic approach that involved making observations as unobtrusively as possible while seated, standing, or walking within 20 feet of participants (investigators were somewhat conspicuous by virtue of their younger age in comparison to the community residents). Genders of the participants and observation locations were documented. Observations were made of pairs of people regarding:

1. what route they took, including stops;

2. the relative ages of the walking companions (e.g., two older adults, one older adult and one staff person, etc.);

3. how the companions positioned themselves relative to each other;

4. whether either person had any obvious disabilities, including walker use;

5. what each person was holding or carrying;

6. whether one person was leading or escorting the other (if so, who was in which role); and

7. the amount of social interaction, both between the two walkers and any interactions with people outside of the pair.

Residents and staff received prior notice of the study through the community's weekly newsletter, and the experimenter willingly explained the nature of the study when requested during observations. To protect residents' privacy, no personally identifying information was collected and no photographs were taken.

Table 3.2: Number of walking pairs observed, separated by situation (escorting or social), with typical interpersonal distances for each situation. 54 pairs observed.

Escorting				
Leader	*Follower*	*Count*	*Typical interpersonal distances* (m)	
			Side-to-side	*Front-to-back*
Resident	Resident	4	0.3–0.5	≤ 0.3 *
Resident	Non-resident	1	≤ 0.3	1.0 *
Non-resident	Resident	14	≤ 0.3	≤ 0.3 *
Non-resident	Non-resident	5	0.3–0.5	varied †

Social			
Pairing	*Count*	*Typical interpersonal distances* (m)	
		Side-to-side	*Front-to-back*
Both residents	15	≤ 0.3	0’
Both non-residents	7	0.3–0.5	varied †
Resident with non-resident	8	0–0.5’	varied †

* Directly side-by-side or leader in front.
†Partners within each pair did not maintain a consistent distance.

3.3.2 Results

Observations were performed in three-hour blocks on 4 days, for a total of 12 hours. Data was collected on 54 pairs of people. The situational breakdown of people can be seen in Table 3.2.

Escorting behaviors

We observed several behaviors specific to escorting situations, including gestures, physical contact, and body movements to indicate direction.

- *Gestures and physical contact.* In escorting situations, the leader often used gestures or physical contact to indicate the intended direction. We observed five instances of the leader pointing toward a destination, and four instances of the leader using physical contact—a hand on the follower’s arm, shoulder, or back—to direct the follower.

- *Body movements.* Intuitively, we suspect that leaders use movement into or out of their partner’s personal space in order to indicate turns along the path. Unfortunately, these movements are subtle and difficult to detect in this sort

of observational study. We observed several instances, typically involving a non-resident leading a resident, where the leader appeared to speed up on outside turns and slow down on inside turns, allowing the follower to maintain a constant speed. In addition, we observed one instance where such body movements failed to properly convey a turn; the leader began to turn a corner by moving away from the follower, but the follower did not immediately correct her movement. Rather, once the pair had separated to about 1.5 m between them, the leader turned to the other, gestured, and said, "This way." Here, a failure in leading via body movements was corrected with a gesture and spoken command.

Interpersonal distances

Table 3.2 lists the typical distances maintained between walking partners. These distances were estimated by the observers from a distance (typically from across the room), and thus should not be considered absolute measurements. All distances (both side-to-side and front-to-back) between companions were highly variable—not just across different pairs of people, but also within individual pairs as they walked. However, we can note that pairs consisting of two residents walking socially or of a non-resident and a resident in an escorting situation tended to maintain much closer side-to-side distance (0.5 m or less) than most other types of pairs. In addition, either partner's use of a walker or cane did not appear to have an impact on their interpersonal distance.

Obstacles and bottlenecks

We observed four main behaviors when pairs encountered obstacles (e.g., another person or object in the way) or chokepoints (narrowing of the passageway):

1. *Simultaneous movement.* Both partners simultaneously move to the side. This behavior was observed only once; both partners were able-bodied and had sufficient space in the hallway to avoid the obstacle.

2. *Speed increase.* One partner speeds up to pass the other and proceeds first (observed 8 times). This behavior occurred primarily in social accompaniment situations, and in particular occurred when one partner was able-bodied but the other was not, in which case the able-bodied partner proceeded first.

3. *Speed decrease.* One partner slows down and allows the other to proceed ahead. This behavior was observed 12 times, in the following situations:

- When both partners were able-bodied, the partner closest to the obstacle fell behind while the other partner proceeded straight ahead.
- When a more able-bodied person was leading a less able-bodied follower, the able-bodied leader slowed down, allowing the other to pass, and often used physical contact (such as a hand on the other's shoulder) to continue guiding the other from behind.
- This behavior was also observed in social accompaniment situations between an able-bodied and a less able-bodied person, in which case the able-bodied person slowed down to let the other pass first.

4. *Separation.* The partners pass on opposite sides of the obstacle. The retirement community's common room has a central lounge area with multiple chairs and couches surrounded by several structural columns. In several cases, when partners were walking together through this area, they would separate and pass on opposite sides of a column or chair before returning to side-by-side travel. This behavior was observed three times and only occurred in this common area around inanimate objects; no partners were seen separating within a restricted hallway or around a person.

Unexpected stops

We observed five instances of one partner stopping suddenly—to speak to a passerby or to search through a bag—without the other partner's prior knowledge. In each of these cases, the other partner continued on for 0.5–2.5 m before stopping, then turned to face the stopped partner. Generally, the other person did not reverse direction, but rather waited in place for the first to resume walking.

Social interaction

In general, partners who were conversing with each other tended to look forward, with occasional glances toward the other partner. However, more detailed observations (such as video coding) may be necessary to fully understand the use of gaze in such situations.

Gender differences

We are not currently able to report on gender differences due to a heavy bias toward women in both the residents and the staff. Of the 54 pairs observed, only 15 contained at least one male partner, and only three of those pairs were both male.

3.3.3 Summary

We performed an observational study of how older adults walk together in pairs. From this study, we can derive some specific conventions that people obey when walking in pairs, such as how far apart they walk from each other and how they signal where they are going. We argue that all of these rules can be defined as mathematical constraints on the partners' movements.

An obvious question regarding this research is whether it generalizes to locations and populations other than this particular retirement community. Obviously, we cannot state conclusively that it does, without further research. However, from our own casual observations of people in day-to-day life, we anticipate that the conventions we listed above do generalize (at least to American populations). In our COMPANION framework (Chapter 4), we implement such conventions as flexible mathematical formulae, which can easily be adjusted to different situations. While conventions such as interpersonal distances and speeds are likely influenced by the relative ages, social status, relationship, and so on, between the two partners, these can all be modeled by additional societal constraints within the COMPANION framework (see Chapter 8).

3.4 Summary

In this chapter, we presented several aspects of research that we performed prior to developing the COMPANION framework (which is described in detail in the next chapter).

We developed a person-tracking system that relies only on the robot's laser range-finder for detecting and tracking people. The system uses particle filters to track "person-sized" segments in a laser scan, and is able to handle short occlusions.

By studying people's reactions to a robot that follows behind a person, we found that people prefer the robot to follow as if it were human. In particular, the behavior rated more human-like ("direction-following" rather than "path-following") was also rated as better matching people's expectations for the robot's behavior. Since people are better able to understand and react to a robot's movements if it behaves according to their expectations, this study provides preliminary evidence that people expect robots to move in a human-like manner. This finding aligns with other recent studies on human–robot social interaction (e.g., Mutlu and Forlizzi, 2008).

In order to better understand some of the social norms people use, we performed an observational study of people walking together. We collected data on people escorting one another, as well as walking together socially. From this, we

were able to enumerate many of the conventions used in each situation. In particular, we quantified interpersonal distances (which were shown to be highly variable) and behaviors around obstacles and chokepoints. These conventions were utilized in the development of methods for a robot to escort a person side-by-side (see Chapter 6) and in the design of a new robot for social human–robot interaction (see Chapter 7).

Chapter 4

Approach

Our approach to social robot navigation is the COMPANION framework: a Constraint-Optimizing Method for Person-Acceptable NavigatION. In this chapter, we argue two main points: that appropriate social behavior requires optimal global planning for obstacle avoidance, rather than locally reactive behaviors, and that social behaviors can be represented as a relatively small number of mathematical cost functions, which can be used for planning. This chapter discusses these points as well as details of our implementation of the framework. The COMPANION framework was first introduced in Kirby et al. (2009a), but is expanded on here.

4.1 Optimal global planning

As people walk around each other, they account for social conventions, such as avoiding people's personal space, while also trying to optimize their task requirements, such as traveling the least possible distance to their goals. While other researchers have implemented social conventions as reactive behaviors (see discussion in Section 2.2.3), we believe that these trade-offs between task goals and social conventions occur at a global level. Consider the following example:

Scenario 1. *Consider walking down an office hallway and encountering someone walking toward you. In the United States, social convention dictates that you should move to the right side of the hallway; the other person will do similarly, thus allowing you to pass each other without incident. However, suppose instead that your goal is an office down an intersecting hallway to your left. You may now choose to walk across the hallway in front of the oncoming person, effectively passing them on the left of the corridor.*

Neither of the behaviors described in the above scenario are *anti*-social, and both behaviors allowed the person to reach his or her goal. Instead, this scenario presented a personal trade-off between social conventions and what we might call "task conventions," such as the desire to reach a goal in as little time as possible. While it may seem that we could enumerate all possible scenarios in order to define a set of reactive behaviors, this quickly becomes infeasible, given the myriad ways that different conventions may interact with each other. Rather, we argue that, for a robot to navigate in a human-like manner, it must account for human social conventions not just at a reactive level, but at a global planning level. That is, social human behavior cannot be fully represented by a hybrid approach in which reactive avoidance maneuvers are performed without consideration of the overall goal, but simply perturb a given path. To further demonstrate this point, consider this second scenario:

Scenario 2. *Consider again walking down an office hallway, but this time noticing a large crowd ahead. If you were in a great rush to reach your goal, you might simply maneuver straight through the crowd. However, you may also choose to take a longer path, along a side-corridor, and thus avoid interrupting the group despite the larger distance you must now travel.*

If social conventions were purely reactive behaviors, then this complex behavior—choosing an entirely different path to the goal—would never occur. By considering social and task conventions together at a global level, then, we can better model human behavior. At a global scope, the robot can consider each convention as a constraint that must be optimized. Furthermore, since the robot may not know exactly what people in the environment will do at each instant, the robot must continually re-plan its path, so that it will properly react to people's movements in a global manner.

Thus, to produce human-like navigation in a mobile robot, the robot must use a fast global planner that is capable of optimizing among multiple constraints (that is, multiple social and task conventions). As discussed in Section 2.2.2, most global path planning algorithms fall into two categories: either heuristic search algorithms or randomized planners. Because we require the ability to optimize a cost function, we have chosen to use the heuristic planner A* with a cost function that accounts for both task and social conventions, expressed as mathematical costs.

The basic A* algorithm to find a path from start state s_{start} to end state s_{goal} is presented in Algorithm 4.1. The algorithm relies on a cost function between two states ($cost(s_1, s_2)$) and a heuristic function ($h(s)$) to estimate the expected remaining cost to goal. A* is guaranteed to find the optimal path (given the cost function) as long as the heuristic is *admissible*, meaning that it never over-estimates the cost to the goal. The time complexity of A* depends on the quality of the

Algorithm 4.1 Basic A* algorithm to find an optimal path from start state s_{start} to end state s_{goal}, given cost function $cost(s_i, s_j)$ and heuristic function $h(s_i)$.

```
1:  g(s_start) ← 0;
2:  OPEN ← priority queue containing s_start;
3:  CLOSED ← empty set;
4:  while OPEN ≠ ∅ do
5:     s ← state from OPEN with lowest f(s);
6:     if s == s_goal then
7:        Reconstruct path in reverse from s_goal to s_start with parent links;
8:        return Path;
9:     end if
10:    add s to CLOSED;
11:    for all s_n ∈ neighbors of s do
12:       c ← g(s) + cost(s, s_n);
13:       if s_n ∈ OPEN and g(s_n) > c then
14:          Remove old s_n from OPEN;
15:       end if
16:       if s_n ∉ OPEN and s_n ∉ CLOSED then
17:          g(s_n) ← c;
18:          f(s_n) ← g(s_n) + h(s_n);
19:          Add s_n to OPEN;
20:          parent(s_n) ← s;
21:       end if
22:    end for
23: end while
24: return No path.
```

heuristic, but in general the number of nodes expanded is at least polynomial in the length of the solution and the size of the state space; with a sub-optimal heuristic, the growth is exponential (Russell and Norvig, 2003).

For typical shortest-distance path planning, the cost function used by A* is the distance between two states, and the heuristic function is a rough estimate of remaining distance to the goal. In order to account for multiple costs, we use a weighted linear combination of individual costs, i.e.:

$$cost(s_1, s_2) = \sum_i w_i \cdot c_i(s_1, s_2) \tag{4.1}$$

where each individual cost c_i has an associated weight w_i. Each cost may also have an associated heuristic function, which may be weighted similarly[1]:

$$h(s) = \sum_i w_i \cdot h_i(s) \tag{4.2}$$

The exact cost functions and weights used are addressed in Section 4.2 and Section 4.3, respectively. Computing the costs for each constraint adds an additional overhead to the time complexity of A*.

Note that we use the phrase "global planning" to underscore the goal-directedness of the resulting robot movement, as distinct from purely reactive, behavior-based systems. Our method allows a robot to respond to obstacles in an intentional way, given some goal. That goal, however, is understood to be relatively short-term—such as between two offices on the same floor of a building, or from an office to an elevator. In this document, we demonstrate paths planned to goals on the order of 10–20 meters away. We assume that, if longer paths are needed, a higher level planner would be used to supply waypoints to our planning system. This assumption is necessary due to the computational difficulties of path planning in real-time. The robot must react to changes in people's behaviors, and it can only do so in a global way if the path planner executes in a matter of milliseconds. As we will discuss in Section 4.4 below, such rates can be difficult to achieve with this system, particularly over long distances.

A final note must be made regarding the definition of "optimal." In typical path-planning, optimal paths are those which minimize the travel distance. However, optimality can easily be extended to minimizing (or even maximizing) any arbitrary cost function. In this thesis, we consider paths optimal if they minimize the given cost function, even though these paths are typically not shortest-distance paths (see, for example, Figure 4.1). Furthermore, note that we implement the search on an 8-connected grid with discretized actions (see Section 4.4), which may not produce paths that are optimal if measured over a continuous space. However, discretization is necessary for tractability.

4.2 Constraints

Constraints and objective functions are related mathematical concepts. *Constraints* limit the allowable range of a variable (e.g., "x is constrained to be less than 100"). Constraints may be *hard* or *soft*: hard constraints provide an absolute limit, whereas soft constraints allow a variable to pass a given limit, but at an associated

[1] In fact, as long as each weight in the heuristic function is less than or equal to the corresponding weight in the cost function, the heuristic remains admissible.

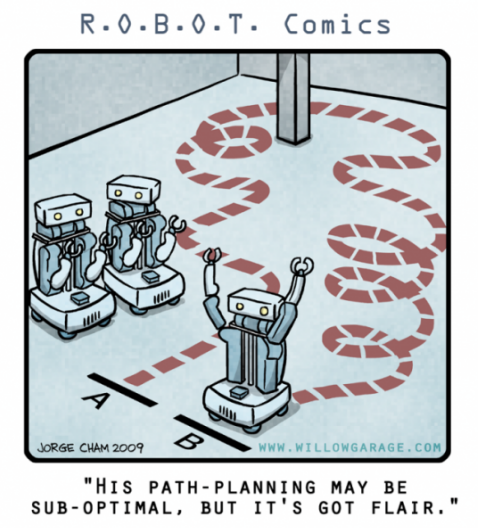


Figure 4.1: While this robot's path may be sub-optimal with regard to distance, perhaps its optimality may be measured by a "flair" function. Comic is distributed under the Creative Commons License; image courtesy Willow Garage.

cost. A cost or *objective function* is a mathematical function that can be *optimized*—that is, maximized or minimized. Soft constraints and objective functions can be mathematically transformed into each other (Williams, 1999); in the following, we will use the terms interchangeably.

We have identified a small set of constraints as particularly important for social behavior in hallway situations, as shown in Table 4.1. The first three, minimizing distance and two aspects of obstacle avoidance, relate to the *task* of traveling to a goal, whereas the person avoidance, personal space for both people and the robot, and tending to the right all relate to the *social* aspects of traveling around people. Three further constraints—default velocity, facing the direction of travel, and inertia—can be understood as both task and social conventions, because failing to observe them is both inefficient (task-related) and socially awkward (social conventions).

In the following, cost functions for each constraint are expressed in terms of s_1, s_2, and a, where s_1 and s_2 are world states, containing the robot's position $(s_i.x, s_i.y, s_i.\theta)$ and velocity $(s_i.v_x, s_i.v_y, s_i.v_\theta)$ as well as the positions and velocities of people in the environment, and a is the action that moves the robot from s_1 to s_2. Actions consist of the desired velocity triplet $(a.v_x, a.v_y, a.v_\theta)$ as well as an execution time $(a.t)$. Not all constraints rely on all aspects of the world state;

Table 4.1: Relevant constraints for a robot that navigates around people.

Constraint Name	**Type**	**Description**	**Notes**
Minimize distance	Task	Find shortest distance paths to a goal.	Lower weights allow greater deviation from the shortest path due to other costs.
Obstacle avoidance	Task	Hard constraint to avoid hitting obstacles.	Necessary for navigating environments safely.
Obstacle buffer	Task	Soft constraint to keep a safety zone around obstacles.	Higher weights result in a larger buffer zone.
People avoidance	Social	Hard constraint to avoid driving through people.	Turning off will allow physical contact with people, which may be necessary in highly crowded situations.
Personal space	Social	Keep a "bubble" of personal space around people in the environment.	Needs to be sized appropriately for the culture.
Robot space	Social	Keep a "bubble" of space around the robot.	Helps the robot to keep people away from its front.
Pass on right	Social	Tend to the right side when passing oncoming people.	Can be mirrored for a tend-to-the-left version.
Default velocity	Social + Task	Prefer to keep a set pace.	Modeled on people's preference to reduce energy expenditure.
Face travel	Social + Task	Prefer to face the direction of travel, rather than sidestepping.	Keeps the robot from driving sideways down hallways. Weight must be balanced with the "inertia" constraint.
Inertia	Social + Task	Prefer to drive straight, rather than turning.	Relative weight between "inertia" and "face travel" helps determine whether the robot will sidestep or turn in an arc around obstacles.

a summary of influencing factors is shown in Table 4.2. The following sections describe each of the constraints in detail.

4.2.1 Minimize Distance

When walking to some goal, people tend to choose paths that minimize their energy expenditure (Sparrow and Newell, 1998; Bitgood and Dukes, 2006), taking shortcuts when available (Whyte, 1988). At some level, people plan to take the shortest possible path to their destination. Thus, one part of the robot's objective function should be to minimize the overall path length to the goal. This cost is computed by finding the distance traveled between two states, as follows:

$$c_{distance}(s_1, s_2, a) = \sqrt{(s_2.x - s_1.x)^2 + (s_2.y - s_1.y)^2} \tag{4.3}$$

Distance is also used as a heuristic function for the A* planner; that is:

$$h_{distance}(s) = \sqrt{(s_{goal}.x - s.x)^2 + (s_{goal}.y - s.y)^2} \tag{4.4}$$

This heuristic can be improved by initializing the minimum distance to the goal from any point via a wavefront propagation algorithm, which is a simple method for pre-computing the shortest distance to a goal from any point on a map.

4.2.2 Obstacle Avoidance

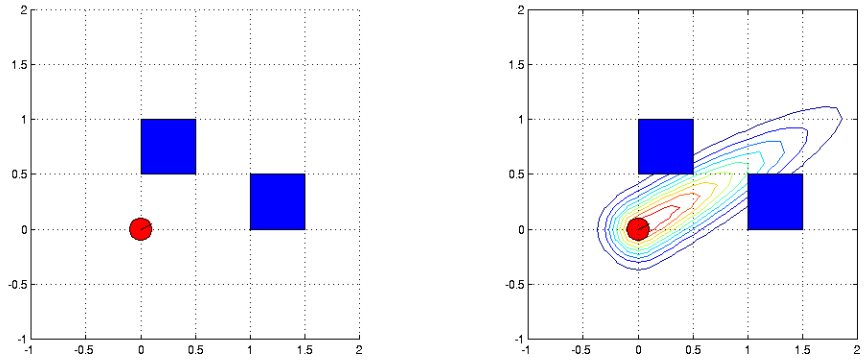
When navigating through an environment, a robot must avoid obstacles in some way. In particular, the robot has a hard constraint against collisions with static obstacles in the world. This is a standard constraint in robot path-planning algorithms. To quickly compute collisions on the path between two states, we employ the Bresenham Line Algorithm (Bresenham, 1965) on the robot's map of its environment. Actions that would produce a collision are discarded. All static obstacles are assumed to be represented in the map. In our experiments, the map of the environment is learned in advance, but it could be continually updated as the robot moves and detects new (non-human) obstacles.

4.2.3 Obstacle Buffer Space

In addition to hard obstacle avoidance, the robot also attempts to keep from traveling too close to static obstacles. It does this by incurring a cost when it approaches obstacles. While this cost is primarily for the robot's safety, it also mimics human behavior. The cost varies according to the robot's speed and direction. The cost for each point on the map is computed by considering a two-dimensional *Asymmetric*

Table 4.2: Influencing factors for each constraint given in Table 4.1.

Constraint Name	**Robot State**					**People's States**			**Obstacles**
	(x, y)	θ	v_x	v_y	v_θ	(x, y)	θ	v	(x, y)
Minimize distance	x								
Obstacle avoidance	x								x
Obstacle buffer	x		x	x	x				x
People avoidance	x		x	x	x	x		x	
Personal space	x					x	x	x	
Robot space	x	x	x	x		x			
Pass on right	x					x	x		
Default velocity			x						
Face travel				x					
Inertia					x				



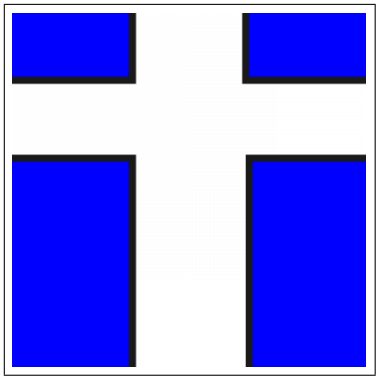
(a) Robot (red circle) approaching some obstacles (blue squares).

(b) Robot approaching obstacles, with Gaussian cost function centered over the robot.

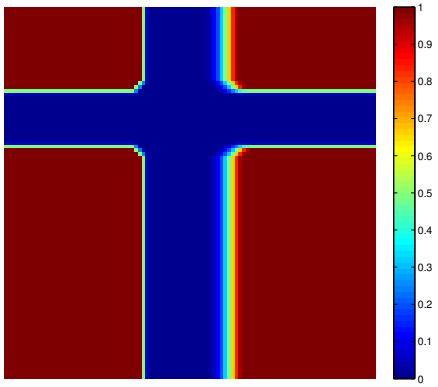
Figure 4.2: Computing the obstacle buffer cost, for a robot driving at 1.0 m/s at a 30° angle. The cost for the robot to be in this state is the maximum value of the Gaussian function intersecting any obstacle. If there are no obstacles, the cost for the state is 0.

Gaussian function centered over the robot. The Asymmetric Gaussian function, as we define in Appendix A, is composed of two halves of 2D Gaussian functions: an elliptical function in one direction, and a different ellipse in the opposite direction. For the obstacle buffer space, we define $\sigma = v$ in the direction of the robot heading (where v is the robot's velocity), and $\sigma = v/6$ to the side (and behind). The cost is the highest value of this Gaussian intersecting any obstacle in the environment. Additionally, since the Gaussian function rapidly approaches zero, obstacles that are far away from the robot can be ignored completely. This function gives a high cost for driving quickly directly toward an obstacle, and lower costs for driving slowly, particularly along the side of obstacles, such as along hallways. These particular values of σ were chosen so that the robot would begin incurring costs within a 2-second time window of obstacles to the front. See Figure 4.2 for a visual depiction of how this cost is computed.

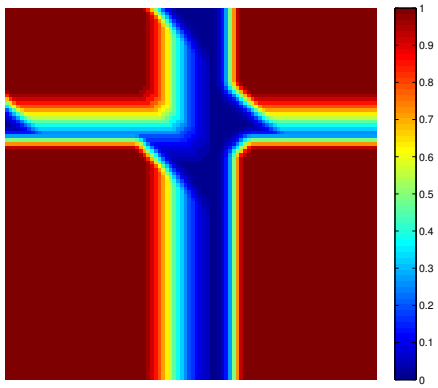
The obstacle buffer cost is computed in advance, according to obstacles known on the map, and is computed for a discrete set of possible angles and velocities. Figure 4.3 shows the cost regions and various speeds and directions of the robot, for a simple hallway environment (which is used in Chapter 5). To compute the cost of traveling between states s_1 and s_2, we again use the Bresenham Line Algorithm in order to sum the costs for each discrete cell on the map that such a transition causes the robot to pass through.



(a) Map of a simple environment, composed of intersecting hallways.



(b) Obstacle buffer costs for the simple map shown in (a), for the robot traveling at a velocity of 0.3 m/s at $\theta = 0$ (i.e., to the right).



(c) Obstacle buffer cost region for the simple map shown in (a), for the robot traveling at a velocity of 1.0 m/s at $\theta = 3\pi/4$ (i.e., toward the upper left-hand corner).

Figure 4.3: Obstacle buffer cost regions for two robot velocities and directions, where the shading corresponds to the cost of encountering that spot on the map. For a faster speed (c), the cost regions cover a larger portion of the map. Furthermore, the robot's direction of travel influences the width of the cost region, so that the robot incurs a higher cost when driving directly toward an obstacle rather than along side one.

4.2.4 Person Avoidance

While certain circumstances may arise in which a robot may be allowed to contact a person, the robot must never plan a path that would attempt to drive *through* a person at any point in time. In planning, this can be achieved by rejecting robot actions that would cause the robot's path and the person's path to intersect.

To implement this constraint, we consider the rectangular swaths that a person and the robot would each cover during one timestep. If the swaths collide, then the constraint is violated. The robot's width is assumed to be known; people are assumed[2] to have a width of 30 cm. While fairly narrow, this width will prevent the robot from planning a path that directly intersects with someone while not discounting paths that approach a person very closely. This computation is done for each person in the environment.

Note that this constraint may actually not be desired in some situations, particularly if the robot's prediction of people's paths is overly naïve. For example, our current implementation naïvely assumes that stationary people will remain stationary, and moving people will always continue at the same speed and direction. Thus, if a person is standing in the middle of a hallway and the robot cannot fit on either side, the robot will attempt to find a path through an alternate hallway, or will declare failure, rather than assume the person might shift out of its way. Furthermore, we might want the robot to be able to nudge against people in a highly crowded environment, which again would require that the robot reason about people moving in response to it. As such reasoning about people is beyond the scope of this research, we use the hard person-avoidance constraint to provide a level of safety to the robot's behaviors.

4.2.5 Personal Space

As discussed in Section 2.1, personal space, or more broadly *proxemics*, is the "bubble" of space that people attempt to keep around themselves and others (Hall, 1966; Ashton and Shaw, 1980; Aiello, 1987). The shape of personal space is asymmetric—greatest to the front of a person—but its exact size is not constant and differs across cultures and familiarity groups (Baxter, 1970). Furthermore, the size of personal space can change based on walking speed as well as other factors (Gérin-Lajoie et al., 2005).

Some attempts have been made to measure the robotic equivalent of personal space (e.g., Nakauchi and Simmons, 2000; Walters et al., 2005). In general, these studies have found that people tend to keep a similar space around a robot as if

[2]The actual width of a person could be used if it were sensed reliably.

it were human, so the constraint to our planner should also respect human-like tendencies.

The personal space constraint can be modeled as an Asymmetric Gaussian function (see Appendix A). This function is our own formulation to model the shape of personal space as defined in the literature; we are unaware of any pre-existing mathematical formula for personal space. We align the cost function to the person's heading; that is, $\theta = \theta_p$. In this direction, the variance of the Gaussian function is set to:

$$\sigma_h = \max(2v, 1/2) \tag{4.5}$$

where v is the person's velocity. The variances to the side and rear are given as:

$$\sigma_s = \frac{2}{3}\sigma_h \tag{4.6}$$

$$\sigma_r = \frac{1}{2}\sigma_h \tag{4.7}$$

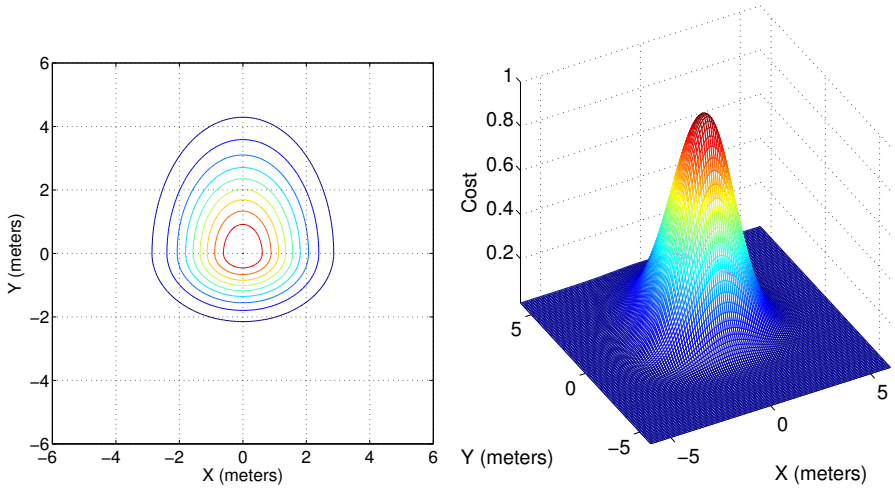
This cost function was designed to roughly match the personal space kept in the United States, as described in Section 2.1. In particular, the cost is greatest in front of a person, and least behind. Since personal space tends to have the same basic shape (if not size) across cultures, modifying this cost to be more appropriate in another culture requires only a scaling of σ_h.

Figure 4.4 shows the cost function for a person moving along the positive Y-axis, with a velocity of 1.0 m/s. A special case is for a stationary person, for which we force the cost to be symmetric, as shown in Figure 4.5. This is because the robot cannot currently detect a stationary person's orientation reliably. With better sensing technology, this special case would not be necessary.

To compute the cost $c_{personal-space}(s_1, s_2, a)$ between two states, we compute an approximate integral of the value of the Gaussian function over time (see Appendix A for details). The cost is summed for each person in the environment.

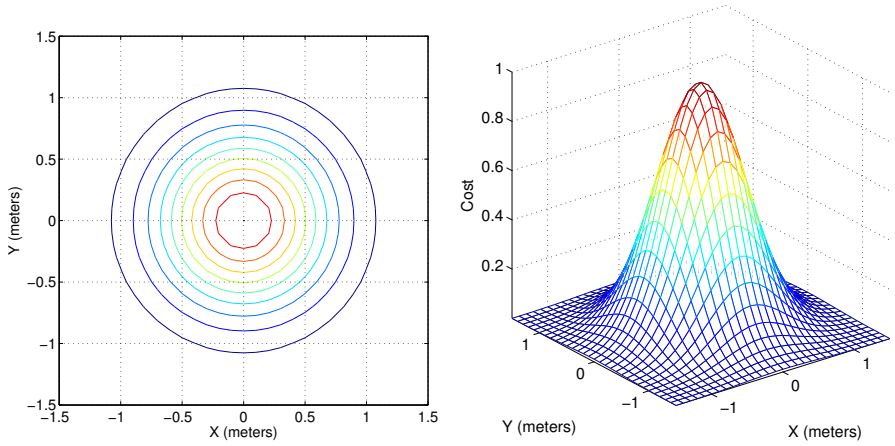
4.2.6 Robot "Personal" Space

As mentioned above, personal space can be described as the space people keep around *themselves* as well as others. That is, people not only avoid entering the personal space of others but also try to keep their own personal space free. Currently, we use the same formula for the robot's "personal" space as we do for human personal space, but with the size and orientation of the function dependent on the robot's velocity and facing rather than the person's. If future research indicates that people prefer a different amount of space between themselves and a robot, the cost function can be grown or shrunk accordingly. The cost is summed for each person



(a) Contour map (b) Surface plot

Figure 4.4: Personal space cost for a person moving at 1.0 m/s along the positive Y-axis (up).



(a) Contour map (b) Surface plot

Figure 4.5: Personal space cost for a stationary person. The cost function is symmetric because the robot cannot reliably detect a stationary person's orientation. Note the difference in scale from Figure 4.4; the personal space of a stationary person is smaller than that of a moving person.

in the environment. Since people far from the robot would incur an infinitesimal cost, in practice the robot can ignore any person beyond some distance threshold when computing the total cost.

4.2.7 Pass on the Right

When approaching a person who is traveling in the opposite direction, people typically avoid collision by moving to one particular side. In the United States, people tend to move to their right (Bitgood and Dukes, 2006). This tendency can be modeled by adding a region of increased cost to the right-hand side of people in the environment. In a head-on encounter, this will cause the robot to prefer to stay to its right (the person's left). Modeling this convention in this way also accounts for the tendency of people to pass a slower-moving person headed in the same direction on the left. As with personal space, the convention to pass on the right can be modeled as an Asymmetric Gaussian function, as shown in Figure 4.6. For this constraint, the parameters of the Asymmetric Gaussian function are given as:

$$\theta = \theta_p - \frac{\pi}{2} \tag{4.8}$$

$$\sigma_h = 2.0 \tag{4.9}$$

$$\sigma_s = \frac{1}{4} \tag{4.10}$$

$$\sigma_r = 0.01 \tag{4.11}$$

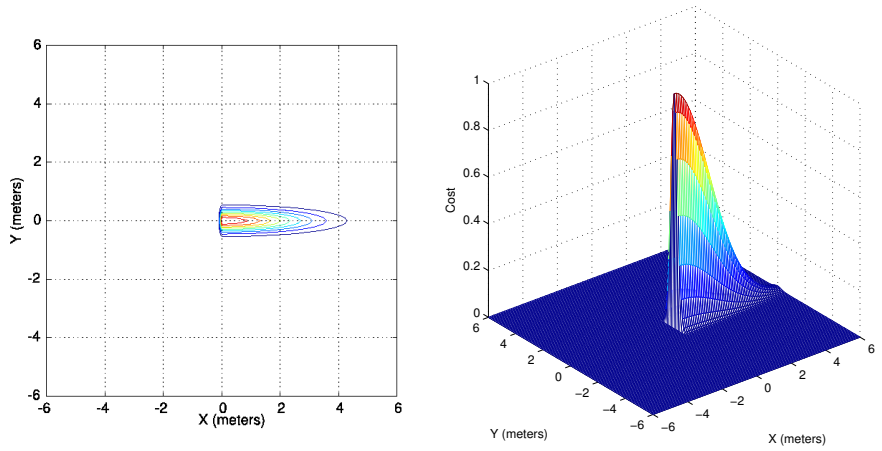
where θ_p is the heading of the person.

The small σ_r allows the function to remain smooth (though smoothness is not strictly necessary). Note that this cost is dependent only on the person's orientation, not on their velocity, and is designed to reach well across most hallway environments. However, no cost is incurred for a stationary person, as our own observations have indicated that people tend to the right only when passing someone who is moving. The cost is summed for each person in the environment.

To modify this constraint for cultures that pass on the left, the cost function merely needs to be rotated by 180°, that is, $\theta = \theta_p + \pi/2$ (as we will show in Section 5.1.3).

4.2.8 Default Velocity

To navigate around moving obstacles, the robot should be able to modify its speed when appropriate—for example, so that the robot can slow down when an obstacle unexpectedly moves into its path, rather than quickly swerving out of the way. However, people tend to keep a set pace, as this minimizes their energy expendi-



(a) Contour map (b) Surface plot

Figure 4.6: Tend-to-the-right cost for a person moving along the positive Y-axis (up). The person is centered at (0,0). The robot can freely pass on the person's left, but incurs a cost for traveling on the person's right.

ture (Sparrow and Newell, 1998). Similarly, the robot should prefer to keep a constant velocity. Changes to the default velocity should result in a cost to the robot, such that the robot would have a cost trade-off between slowing down versus traveling a greater distance around an obstacle or person. We model this objective as proportional to the absolute difference between the chosen velocity and the default velocity; that is, both increases and decreases in speed incur a cost, and greater changes cause greater costs. This cost is computed according to the following equation:

$$c_{velocity}(s_1, s_2, a) = a.t \left| v_x^0 - a.v_x \right| \tag{4.12}$$

where v_x^0 is the desired forward velocity. The cost is scaled by the time to execute the action, so that if actions can have variable execution times (as they do when a variable grid is used; see Section 4.4), the cost will also differ. Because of the time scaling, this cost is typically small, and thus may require a higher weighting in the overall objective function to cause a significant change in the planner's behavior. Since moving quickly over one grid cell requires less time than moving slowly the same distance, this function tends to prefer speed increases over speed decreases. However, this tends to be balanced by other constraints that cost more at faster speeds, such as the "obstacle buffer" and "robot 'personal' space" constraints.

4.2.9 Face Direction of Travel

Humans are able to sidestep around obstacles without changing their facing. While not all robots (such as common differential-drive robots) are capable of sideways movements, those that are should be able to take advantage of this type of holonomic[3] movement. However, just as people do not typically sidestep for extended periods of time—such as down entire hallways—the robot should incur a cost for sideways maneuvers. For people, this behavior results from a kinematic expense to stepping sideways (that is, walking forward requires less energy); even if the same is not true for a robot, sideways movement over long distances is also *socially* awkward.

As with the "default velocity" constraint, this cost can be modeled as proportional to the difference from a default velocity. In this case, we consider only the velocity in the sideways (y) direction, and wish to keep that velocity at 0:

$$c_{facing}(s_1, s_2, a) = a.t\,|a.v_y| \tag{4.13}$$

In addition, this cost is also scaled by the action time, resulting in generally small costs per action.

Unfortunately, most robots that are used in human–robot interaction research are non-holonomic, and thus cannot produce this human-like sidestepping behavior. The identification of this constraint led to our development of a new robot for human–robot interaction studies (see Chapter 7).

4.2.10 Inertia

The inertia constraint is similar to the "default velocity" constraint, except that it applies to rotational velocity. Again, just as people prefer to move in a straight line, the robot also should prefer to keep the same heading, rather than turning.

$$c_{inertia}(s_1, s_2, a) = |\alpha| \tag{4.14}$$

where α is the normalized difference in angle between states s_1 and s_2 (that is, $-\pi < \alpha \leq pi$). On an 8-connected grid with discretized actions, the magnitude of

[3]Technically, the term "holonomic" relates to degrees of freedom versus degrees of control. In this document, we focus on robots that move in a single plane (e.g., a floor), and assume that the robots have three degrees of freedom (x, y, and θ). In this context, a *holonomic* robot can control all three degrees of freedom instantaneously, while a *non-holonomic* robot typically has only two degrees of control—usually forward and turning, but not sideways (y) movements. Thus, in this document, we tend to use the term "holonomic" to imply "sideways-capable," even though this is an over-simplification.

turning is not changed with different action execution times; as a result, this cost is not scaled by time.

4.3 Weighting the constraints

All of the constraints given above must be combined into a single objective function. As discussed in Section 4.1, we use a linear combination in which each constraint has an associated weight. In this section, we will discuss this weighting.

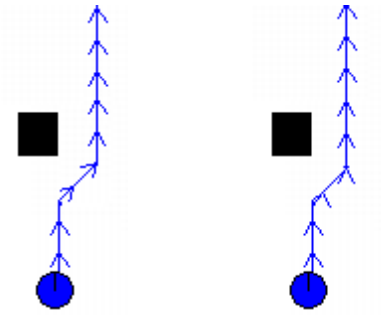
Social conventions are tendencies toward a particular type of behavior; they are not hard rules. Individual people vary widely in their particular behaviors. We argue that the COMPANION framework can be used to create a *range* of socially acceptable behavior using different relative weights between the constraints given in the previous section. That is, though different sets of weights will produce different behaviors, we argue that socially acceptable behaviors will result from any number of such sets. The resulting behavioral differences resulting form different weightings can be interpreted as different "personalities," as we will discuss in Section 5.1.2.

Some weights can be determined analytically, given a desired behavior. For example, consider the "face direction of travel" and "inertia" constraints. The relative weighting of these two constraints will determine whether the robot will sidestep a static obstacle or turn in an arc around it, as shown in Figure 4.7. Driving on an arc around the obstacle requires two 45° turns, given an 8-connected grid. Since the "inertia" cost is equal to the radians turned, such a maneuver will represent a constant cost of $w_{inertia} \cdot \pi/2$. Keeping a constant heading while sidestepping the obstacle incurs a cost relative to the distance moved sideways. If the robot moves s meters to the side, then this cost is $w_{facing} \cdot s$. Suppose we want the robot to move sideways for a maximum of 1 meter, and turn to face travel for longer stretches. We thus want the following inequality to hold:

$$s \cdot w_{facing} < \frac{\pi}{2} w_{inertia} \qquad \text{for } s < 1 \tag{4.15}$$

$$s \cdot w_{facing} > \frac{\pi}{2} w_{inertia} \qquad \text{otherwise} \tag{4.16}$$

Thus, the relative weights of $w_{inertia} = w_{facing} \cdot \pi/2$ will satisfy the above equations. Note, though, that setting the weights in this way will not *force* the robot to move sideways for 1 meter in all cases, particularly when the robot is moving around a person. This is due to additional constraints that rely on the robot's heading, such as the robot's "personal space." In particular, the robot may prefer a turn



(a) Driving on an arc.

(b) Sidestepping.

Figure 4.7: Two ways of navigating around an obstacle: keeping the same heading while sidestepping (b), or always facing the direction of travel while driving in an arc around the obstacle (a). Arrows on the paths indicate the direction the robot is facing and are drawn every 40 cm.

that keeps a person out of its space, even if the cost of moving sideways is less than the cost from turning.

We can compute similar relationships between other constraints to determine particular behaviors. For example, the relative weights between the "shortest distance" and the "obstacle buffer" costs will determine how close the robot will approach corners when turning. However, such a relationship is more complicated than that shown above, due to the more complicated mathematical form of the "obstacle buffer" cost, and also because "obstacle buffer" cost varies according to the robot's speed, adding an additional dependency on the "default velocity" constraint. Defining all such relationships is beyond the scope of this research, as our goal is to produce generally social behavior, rather than specifically model one particular behavior. Section 5.1.2 demonstrates some examples of how different constraint weights can be selected to produce desired behaviors. General relationships between constraints can be seen in Table 4.2; any constraints with overlapping influences will interact with each other. Desired weights could be learned from a training set composed of tele-operated robot data, but that is beyond the scope of this research (see Chapter 8).

Note that the use of a weighted linear combination of constraints precludes a ranked preference ordering. To some extent, preferences can be encoded in the weights; more highly ranked constraints should have greater weights than those ranked lower. However, the COMPANION framework does not support fully dis-

joint constraints, in which only one of several constraints can hold. As we have not identified any such disjoint social conventions, we do not consider this a limitation of the framework.

4.4 Implementation details

We have implemented the COMPANION framework using the Carnegie Mellon Robot Navigation Toolkit (CARMEN[4]). CARMEN provides drivers for many common research robots and sensors, as well as a complete simulation environment. We replaced the built-in path planner and navigator with our own implementation, written primarily in C++.

The remainder of this section discusses specific aspects of our implementation.

4.4.1 Search space

A* searches over a discrete state space, so one key design decision is the implementation of that state space. Clearly, the state space contains the state of the robot, including its x-y position and orientation. Since we allow the robot to travel at different speeds, its velocity (in x, y, and θ) must also be contained in the state space. In addition, the state space must contain some representation of people's locations relative to the robot, and the connectivity between states must be defined. These two design aspects are detailed below.

Representing people

A key tenet of the COMPANION framework is that people cannot be treated as static obstacles: the robot must react to people's dynamic movements through time. A common approach to planning with dynamic obstacles is to add time to the state space, effectively adding another dimension to the search space (e.g., Fraichard, 1999). Unfortunately, this adds a great deal more complexity to the search. Some approaches to managing this complexity include using random planners (e.g., Zucker et al., 2007) or reducing the available action space (e.g., the "canonical trajectories" of Fraichard, 1999). In contrast, our approach is to include the dynamic obstacles (e.g., people), if present, in the state space. This yields several benefits over the state-time representation. To understand why, note that the A* planner, with an admissible heuristic, does not need to examine a state more than once; if it encounters a previously examined state, the new path to that state is guaranteed to be more costly than the one found initially. Adding time to the state

[4]CARMEN is available online via http://carmen.sourceforge.net

space greatly decreases the chances of A* encountering the same state (at the same time) more than once, thus greatly increasing the search time. Adding dynamic obstacles to the state space results in more overlap. In addition, if the planner is allowed to ignore obstacles behind it or moving rapidly away from it (since such obstacles are unlikely to affect the robot's path), the state space can be simplified further. This allows the planner to consider dynamic obstacles with less complexity than a state-time space search entails, while also not limiting the robot's available actions.

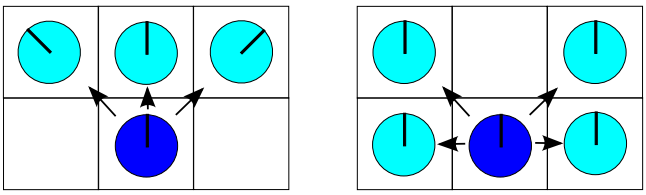
Obstacles are assumed to move in continuous space, even though planning occurs on a grid. To account for this, two world states are considered the same if the robot is in the same state (position, orientation, and velocity) and if all dynamic obstacles are "close enough" to the same positions. We allow "close enough" to vary with the obstacles' distance from the robot. Thus, when obstacles are far from the robot, their positions are considered more coarsely.

Action space

We discretize searching on an 8-connected grid. However, to account for some aspects of vehicle dynamics, not all adjacent states are reachable from any given state. We allow the following actions: straight, forward left turn, forward right turn, stop, sideways left, forward sideways left, sideways right, and forward sideways right. These are shown in Figure 4.8. The first three actions may be executed at any of three speeds: the default speed (for which we typically use 0.5 m/s), a faster speed (0.75 m/s), and a slower speed (0.25 m/s). All actions are not available at multiple speeds to keep the action space tractable. This yields a total of 14 actions at each state. Note, though, that path execution (that is, robot navigation) occurs at a finer granularity (see Section 4.4.4).

4.4.2 Real-time search techniques

Robots that operate in the real world need to respond rapidly to changes in the environment. A plan to the robot's goal, generated at the robot's starting location, quickly becomes invalidated as the environment changes or the robot receives new information. A challenge in mobile robots, then, is replanning paths as quickly as possible. Especially challenging are environments with dynamic obstacles and obstacles with associated costs, such as personal space around people, buffer zones around dangerous vehicles, or rough terrain. Because sensors are imperfect, robots navigating in dynamic environments must replan whenever they receive new sensory data in order to ensure a safe, low-cost path.



(a) Non-holonomic actions. Each of these three actions may be executed at any of three speeds (e.g., default, faster, and slower).

(b) Holonomic actions; none of these actions changes the robot's orientation. Each of these actions may be executed at only the default speed.

Figure 4.8: Non-holonomic (a) and holonomic (b) actions available to the planner.

As discussed in Section 4.1, we use the heuristic planner A* (Hart et al., 1968) to produce optimal paths, according to the given cost function. However, A* alone typically cannot run repeatedly in real-time, which is necessary for a mobile robot operating in the real world. While many variants of A* have been developed to operate in real-time (e.g., D*; Stentz, 1994), none are capable of replanning for a moving robot amongst dynamic obstacles, particularly when those obstacles have associated costs (e.g., personal space). In order to run our planner in real-time, we modify the search space in various ways. Our primary modification involves the use of a variable search grid; others include limiting the action space, ignoring people behind the robot, and searching on a gradient to the goal. Each method is described below.

All of these modifications present trade-offs between search time and optimality; the use of any may result in sub-optimal paths as compared to the un-modified results. However, if the modifications allow the planner to run rapidly enough, only the first action of any single plan will be executed. Thus, the modified planner may still produce optimal behavior from a navigational perspective, even if the individual plans are not globally optimal. Even if the first action of a faster planner differs from the optimal plan, such sub-optimal behavior may be acceptable in order to have the robot react rapidly to incoming sensor data. As computer processor power improves or newer A* approaches are developed, such modifications may become unnecessary.

Variable grid

Heuristic path planners rely on predictive heuristics, such as the remaining distance to the goal, in order to guide the search. Poor heuristics can cause the search to

examine more nodes than necessary. Costs associated with dynamic obstacles (e.g., people) are difficult for heuristic planners because these factors typically do not have useful predictive heuristics. Thus, when a heuristic planner encounters such an obstacle, it must expand a large number of nodes in order to find an optimal path. Unfortunately, heuristic planners such as A* typically have a run-time that is worse than linear in the number of nodes expanded. Reducing the number of nodes the search must expand thus improves the search time.

Our approach (Kirby et al., 2009b) is to modify the search space used by the A* planner. In this way, the planner can be used unchanged. In particular, rather than performing the entire search on a regular grid, as most planning algorithms do, we decrease the resolution of the search further from the robot. That is, only the areas near the robot are searched carefully; areas further from the robot are searched more coarsely. Because this results in many fewer search nodes, planning can occur rapidly. New plans can thus be generated repeatedly as the robot moves, so that the robot will always have a fine-grained path defined for its next action.

This method is related to hierarchical decompositions of space, such as quad-tree-based approaches (as described in Section 2.2.2). Our particular approach uses a variable grid that is composed of regions of regular grids of decreasing resolution, spanning outward from the robot's position, as shown in Figure 4.9. The key idea behind this method is that, if the search can be done quickly enough, then the robot can regenerate plans at each timestep (as it gets new sensor information). Thus, the plan needs to be at a high resolution only near the robot; a rough path is sufficient further from the robot, because the robot will generate a new plan before reaching those areas. Our approach differs from other hierarchical planners in that the grid does not remain static between searches; rather, the grid changes as the robot moves, keeping the finest-resolution cells centered over the robot's position. In addition, by using an implicit representation of the changing grid cells, our approach does not require any additional memory over a typical A* search.

An important aspect of using a variable grid is that action costs may need to scale in relation to the grid size. For example, the "shortest distance" cost must be the actual distance between two grid cells, and thus a larger cost at larger cells. Similarly, the "default velocity" cost is scaled by the travel time, so that larger cells incur larger costs. However, not all costs need to scale; for example, the "inertia" cost relates to the absolute change in heading angle, which does not change at a coarser grid.

Additional design challenges in implementing the variable-grid-cell planner include selecting the grid variations, handling the boundaries between resolutions, and aligning the grid, which we discuss below.

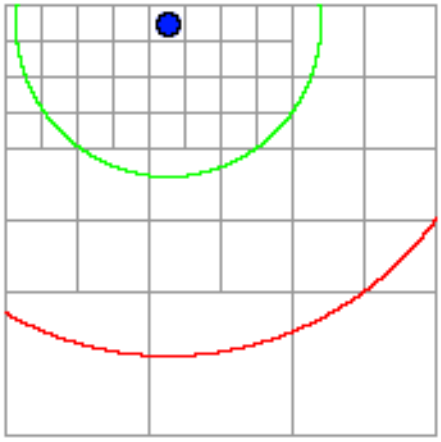


Figure 4.9: A variable grid used for planning. The grid resolution decreases with the distance from the robot (blue circle). Shown are three grid sizes: the finest resolution is close to the robot (within the green circle), next greater is between the green and red circles, and the greatest resolution is furthest away from the robot.

Selecting the grid variations One design consideration with this approach is how to select the grid variations: what resolutions to use, and at what distances to change the resolution. Close to the robot, the planner should use the finest resolution available (e.g., the map resolution). The distance at which the planner can switch to a coarser resolution is dependent primarily on the speed of the robot; the planner should always be able to provide a detailed path for several timesteps. Further away from the robot, a coarser resolution will yield faster path computation, as long as the grid cells are not allowed to be overly large for the environmental conditions.

Furthermore, we must consider obstacles within each grid cell. At the finest resolution available (near the robot), each cell is either occupied or free. However, at coarser resolutions, a grid cell may be only partially free, and partially filled with obstacles. One approach for handling obstacles would be to declare a cell that contains any obstacles to be completely blocked, but this may result in planner failure if even small obstacles appear as large blockades in coarser resolutions. Our approach is to check for obstacle collisions on the straight-line path between grid cell centers, as described in Section 4.2.2.

Dense environments present difficulties for planning on larger grid cell sizes. For example, suppose that the robot will need to navigate through a 1-meter-wide doorway close to the goal. If the grid cell size near the doorway is fairly large

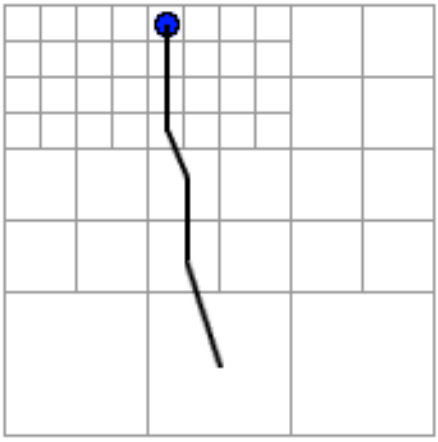


Figure 4.10: A plan generated on the variable grid. Since plans are generated between node centers, a "straight" path may appear to have turns in it.

(say, 0.6 m or 0.8 m), then planning between cell centers may not find a free path through the doorway. This shortcoming can be avoided entirely if one has sufficient *a priori* knowledge of the environment and tailors the grid cell size accordingly. Unfortunately, this may not always be possible. In more complex environments, it may be necessary to perform sub-searches on some of the larger grid cells before declaring them impassable due to obstacles, perhaps using a method similar to the Framed Quadtree approach (Yahja et al., 1998). Since our approach assumes that the robot has a map of the environment (which typically results in poor performance by Framed Quadtrees), one could pre-label obstacle-dense regions on the map that should always be searched at a fine resolution. An alternative approach is to probabilistically estimate the passability of coarser cells, based on their obstacle density.

Resolution boundary challenges An implementation challenge in variable-grid planning is handling the boundaries between resolutions. The primary difficulty involves determining what actions occur at the boundaries, as illustrated by Figure 4.10. However, this challenge can easily be overcome by assuming that the robot will always generate a new plan before it reaches a section of the path that uses a larger grid cell size.

Because actions late in the planned path are assumed never to be executed, the actions across resolution boundaries need only be approximate. Before the robot reaches any of the approximate actions on the plan, it will have generated

a new plan with high-resolution initial actions. This can be guaranteed by simply having the navigation algorithm stop the robot if too much time has elapsed since generating the last plan.

Actions within each section of the grid should move the robot to a neighboring cell of the same resolution. As long as the coarser-resolution sections are integer multiples of the initial grid resolution, computing the within-section actions is trivial. At the boundaries between resolutions, though, actions will not necessarily move the robot to the center of the next grid cell. However, if we assume that the actions are only approximate, as mentioned previously, the path can be aligned to the grid by simply rounding the position to the center of the nearest grid cell. This allows planning to continue on a discrete grid.

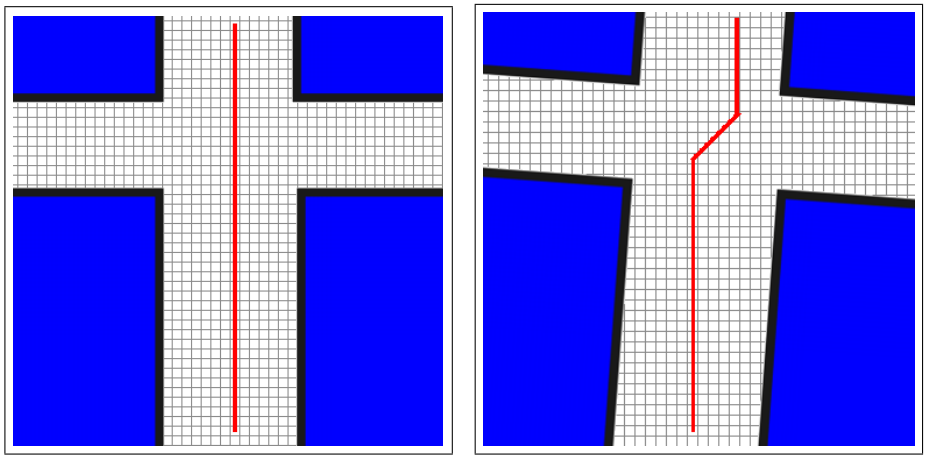
Since we are able to approximate the robot's location to a grid cell of any resolution and we are able to compute each cell's occupancy on the fly, we are able to keep the variable grid implicitly defined over the given fine-grained map. That is, the variable grid incurs no additional memory requirements over the map itself.

Aligning the grid Since the grid is implicitly defined, the robot can overlay its own grid with an arbitrary frame of reference. We consider three notable reference frames: the global map frame, which is fixed to the map definition; the environment frame, which may vary throughout the environment (e.g., if hallways are not all at right angles to each other); and the robot's frame, which changes as the robot moves. For hallway travel, a reference frame that does not align with the hallway results in crooked paths, as shown in Figure 4.11.

Ensuring that the robot always plans on a properly aligned grid could be accomplished by hand-labeling alignment on a map, or by attempting to compute the hallway alignment automatically, as needed. Another method is simply to align the grid to the robot's position and orientation. This will periodically generate paths such as shown in Figure 4.11(b). However, since the robot is generating new plans continually, if the robot is turning, then each new path will have a slightly different grid alignment. Thus as the robot sweeps through a turn, it will eventually compute the straight-line path down the hallway. In the absence of other obstacles, this straight-line path will have the lowest cost, causing the robot to continue straight down the hallway. Thus, in most cases, the robot will eventually align itself to the hallway, typically as the robot executes the first turn in its path.

Other speed improvements

Additional improvements result from further reducing the search space in various ways. We have designed three simple search space reductions that may be beneficial for current real-time operation:



(a) Map, environment, and grid all aligned. (b) Grid aligned to map, but not environment.

Figure 4.11: Examples of how the grid alignment influences possible paths the robot might take. Aligning the grid to the hallway (a) produces the shortest path. In (b), the robot cannot choose a path straight down the corridor, because the grid is misaligned.

- *Limit action space at coarser resolutions.* As discussed above, we allow for 14 possible actions at each state (Section 4.4.1). That is, for each state examined, as many as 14 additional states are added to the search tree. However, if the search operates rapidly enough, the actions far from the robot can be considered approximate; the robot will always generate a new plan before executing actions that occur late in the path. We can thus greatly reduce the action space—for example, eliminating velocity changes and sideways maneuvers—further from the robot. One way to implement this is to use the full action set only at the smallest grid resolution, near the robot, and to use a minimal action set (e.g., only the three actions of forward, left turn, and right turn) at any larger resolutions.

- *Ignore people behind the robot.* During planning, the robot must predict people's future movements over time. As the robot searches for a plan, it may search states in which the robot has driven past a person. If that person is now behind and moving away from the robot, then the robot can reasonably assume that its future path will not depend on that person in any way, and can thus drop the person from the states searched outwards from that point. Since we treat people, rather than time, as part of the state space (see Section 4.4.1), dropping people in this manner can greatly reduce the search space and,

thus, improving the search speed. This method is particularly useful for robots with a limited sensor range, as people need to be considered only from the point at which they are detected until the point at which the robot passes them. However, this method can occasionally result in the planner computing awkward paths in which the robot drives away from the person, in order to eliminate him from the search space, before continuing on toward the goal.

- *Search on a gradient to the goal.* Finally, the search speed can be improved by imposing a hard constraint on the robot's direction of travel, such that it must always drive along a gradient toward the goal. In particular, at the start of each search, we can use a wavefront propagation algorithm from the goal outward to find the shortest distance path from any point on the map. By limiting how much the robot's path can deviate from the distance gradient, we remove from the search space any states in which the robot is driving away from the goal. Note that if the deviation is limited to zero (that is, the robot must follow the gradient), then the other constraints will have minimal (if any) impact on the path. Furthermore, if the limited deviation is sufficiently high so as to allow travel in the opposite direction of travel, then this constraint will have no effect. Rather, the threshold must be set to a value in between these extremes.

  Note that this method violates the argument presented above for global path planning (see Section 4.1), as it forbids the robot to seek out an alternate, significantly longer path to the goal. Because the gradient constraint is formed from the shortest distance to the goal, this method resembles a hybrid planning approach, in which a path is planned first according to task constraints (shortest distance) and then perturbed by social conventions. As previously discussed, this does not fully model people's behaviors. However, this method is a reasonable temporary measure to improve the speed of the search in lieu of faster processing speeds. In addition, this approach may be useful in environments that are known not to have alternate routes, such as in enclosed hallways.

4.4.3 Laser-based person-tracking

The person-tracking method used in this implementation is similar to that described in Section 3.1. However, we modified the tracker in two key ways: first, to use a map of the environment, and second, to better smooth the tracked velocities.

Algorithm 4.2 Pure Pursuit path-following algorithm, from Coulter (1992).

1: Determine the current location of the vehicle.
2: Find the path point closest to the vehicle.
3: Find the goal (look-ahead) point.
4: Transform the goal point to vehicle coordinates.
5: Calculate the curvature and request the vehicle to set the steering to that curvature.
6: Update the vehicle's position.

Map-based tracking

For this work, we make the simplifying assumption that the robot has an accurate *a priori* map of the environment, which is necessary for global path planning.[5] The robot uses the map to match a given laser scan to its location in the environment. Non-matching segments of scans are segmented into person-sized blobs, which are tracked continuously using particle filters.

Velocity smoothing

Because several of the social constraints in our framework depend on the person's direction of travel, the robot needs to have an accurate estimation of the person's velocity. We do this by performing a linear least-squares regression on the person's tracked position over time. In planning, the robot uses the most current estimation of the person's velocity to predict his or her future location.

4.4.4 Navigation

Plans are generated rapidly, and the robot must be able to navigate along the paths as they change. Due to both odometry slippage and nuances of the localization algorithm, the robot is not guaranteed to remain precisely on the path at any given time—in fact, the robot will almost never be located exactly on the path's discrete grid. To keep the robot following the plan as closely as possible, then, we use the Pure Pursuit path-following algorithm (Coulter, 1992), which guides the robot back onto the path if it strays or if a new path is planned. The basic algorithm is given as Algorithm 4.2.

Line 3 of the Pure Pursuit algorithm computes a look-ahead point some distance ahead of the robot on the path, toward which the robot is then commanded to

[5] This could be generalized by having the robot run a simultaneous localization and mapping (SLAM) algorithm.

steer. In our implementation, we use a constant look-ahead of 0.75 m. However, if the path curves around an obstacle, the look-ahead will cause the robot to clip the corner, potentially causing a collision. We thus consider all of the points between the robot's location and the look-ahead point on the path, in comparison to the straight line connecting those two points. If any point along the path deviates more than 0.20 m from the line, that point is used as the look-ahead. Once the look-ahead point has been found, we compute the necessary curvature to steer the robot toward that point.

The basic Pure Pursuit algorithm computes only the desired steering angle to some look-ahead point, but does not explicitly provide a means of computing a desired velocity. Though the plans we generate include the desired velocity at each state, the given action may not lead the robot correctly along the path (e.g., if the localization places the robot at a point other than one of the path waypoints). Furthermore, for holonomic robots, the navigation method must consider whether the desired action to some state is to move along an arc or to drive sideways. We make this decision based on the desired action associated with the robot's closest point on the path. If that action does not have any sideways movement (i.e., $v_y = 0$), then the robot is commanded to drive along an arc with the action's specified v_x, and the corresponding v_θ, according to the formula:

$$curvature = \frac{v_\theta}{v_x} \tag{4.17}$$

If, instead, the action has a non-zero sideways velocity, the robot is commanded to drive on the straight-line path toward the look-ahead point, with the action's desired velocities scaled to produce the appropriate angle.

Each plan is followed until a new plan is received. Since the planner typically runs as fast as new sensor data is received, the navigator will always have a high-resolution action to follow.

4.5 Summary

In this chapter, we have introduced the COMPANION framework: a Constraint-Optimizing Method for Person–Acceptable NavigatION. The framework is based on two key points:

- Appropriate social behavior requires *global planning* through the environment; and

- Social behaviors can be represented by constraining the path planner to minimize a set of mathematical cost functions (that represent social and task conventions).

Global planning is necessary to model human-like behavior. For example, despite social conventions such as "tend to the right side of a hallway when passing oncoming people," people often move to the left side of a hallway in anticipation of a left-hand turn. We argue for the use of an optimal heuristic planner, such as A*, rather than using locally reactive obstacle avoidance.

We define a set of *constraints* (in the forms of hard limits and of cost functions) that represent the task of traveling through hallways while observing social conventions. These constraints include:

- Minimizing the distance traveled to conserve energy while traveling to a goal;
- Avoiding obstacles;
- Keeping a safety buffer around obstacles;
- Avoiding people, including keeping out of their personal space;
- Protecting the robot's own "personal" space;
- Tending to the right when passing people;
- Keeping a default velocity, so as not to expend extra energy;
- Facing the direction of travel, but allowing for sidestepping obstacles as people do; and
- Maintaining forward inertia, rather than zig-zagging.

These constraints are combined into a single objective function by adding the weighted cost of each constraint. The weights can be determined by a number of methods, and we argue that socially appropriate behavior can be achieved with a wide variety of constraint weights.

Finally, this chapter discussed some of the particular details and compromises to consider when implementing the COMPANION framework for real-time planning and execution. These include the representation of state space for the search, several methods aimed at improving the speed of the planning, and details on other components required to run the framework in a complete robot system (namely person-tracking and navigation). The next chapter details the behavior of the implemented system in both simulated and real robot studies.

Chapter 5

Hallway Interactions

Chapter 4 introduced the COMPANION framework for person–acceptable navigation and discussed details of its implementation. In this chapter, we present several key results showing the COMPANION system operating in hallway environments. We present results both in simulation (Section 5.1) and in user studies with the robot Grace (Section 5.2). These results show that the framework is capable of producing socially acceptable paths.

5.1 Simulations

To understand the behavior of the COMPANION framework in the context of hallway interactions, we ran a large test suite of simulated scenarios. Obviously, there are an infinite number of possible situations; here, we present only a limited number that demonstrate the system's behavior.

Unless specified otherwise below, trials were run using all constraints defined in Section 4.2, with the weights shown in Table 5.1. For these scenarios, all constraints were given integer weights from 1 to 3. The "personal space," "robot 'personal' space," and "pass on the right" constraints were each given a weight greater than the similar buffer around obstacles; that is, we allowed the robot to move closer to static obstacles than to people. The robot's "personal" space was weighted more highly than the standard "personal space" constraint, which can be interpreted as letting the robot be less willing to have people directly in front if it than for it to be directly in front of people; this was done in the interest of safety. Finally, the remaining three costs—"default velocity," "face the direction of travel," and "inertia"—were each given a weight of 2, to produce smoother overall paths. Different sets of constraint weights are addressed in Section 5.1.2 below.

Table 5.1: Constraint weights used in the objective function. In addition, the hard constraints of avoiding obstacles and people were used.

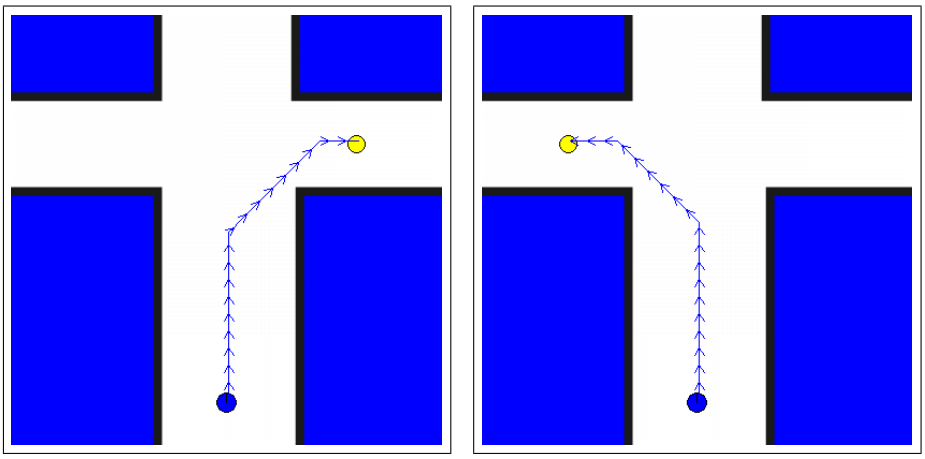
Constraint Name	**Weight (w_c)**
Minimize distance	1
Obstacle buffer	1
Personal space	2
Robot space	3
Pass on right	2
Default velocity	2
Face travel	2
Inertia	2

In general, the various methods discussed in Section 4.4 for improving search speed were not used in the simulation experiments. That is, all searches were performed on a constant grid of 10 cm by 10 cm squares; all actions were available at each search step; people were never eliminated from the state space, regardless of their distance from the robot; and we did not limit the search space with a direction gradient. In most cases, each experiment was performed only once, as the path planner produces deterministic results. Only when probabilistic elements, such as localization and person-tracking, are added, such as in Section 5.1.5, must the path planner be run repeatedly to understand the full system behavior.

The robot being simulated has a circular base, 45 cm in diameter, and is capable of holonomic movement. This matches the specifications of the Companion robot, described in Chapter 7, which was under development at the time of these experiments. The computer used for the simulations contained a dual-core Pentium processor with a 3.4GHz clock speed and 2GB of RAM, and ran the Ubuntu Linux operating system (version 8.04).

5.1.1 Head-on encounters

Here we present a set of scenarios in which the robot must navigate an intersection of two hallways. We use an artificial environment with well-defined, symmetric, straight walls, as this provides a better visual understanding of the robot's behavior. The environment is 10 m by 10 m, with two intersecting hallways; the main hallway is 3 m wide, and the intersecting hallway is 2 m. To establish a base case, we simulated the robot planning a path to each of three goals: a right turn, a left turn,



(a) Goal requiring the robot to turn right

(b) Goal requiring the robot to turn left

(c) Goal straight ahead of the robot

Figure 5.1: Paths planned for the robot to each of three goals in a simple environment with no people present. The robot (blue circle) begins centered in the lower part of the hallway; the goals are shown in yellow. The environment is 10 m by 10 m, and the hallways are 3 m and 2 m wide.

Table 5.2: Search statistics for paths planned for the robot to each of three goals in a simple environment with no people present.

Goal location	Path length	Path cost	Search time	Nodes searched
Right	7.77 m	10.95	0.12 s	4175
Left	7.89 m	11.07	0.13 s	4185
Straight	8.00 m	8.00	0.008 s	81

or straight ahead. These paths can be seen in Figure 5.1. Note that each goal is comprised of both location and orientation; that is, the robot must also face right at the right-turn goal, and so on. Statistics for these searches are shown in Table 5.2. The larger search space and time for the "left" and "right" conditions are due to the need to search around a corner in each case; similarly, the path costs are higher in these conditions due to the turns (thus incurring inertia costs), as well as obstacle buffer costs near the corners. The slight asymmetry between the left- and right-turn goals is due to alignment on the 10 cm grid cells. In these scenarios, the only actions used are forward and turning maneuvers at a constant speed.

To understand the planner's behavior in simple head-on encounters, we introduced a single person into the environment. The person began facing the robot with one of three starting locations:

- The left of the hallway with respect to the robot (i.e., the right with respect to the person);
- Centered in the hallway; or
- The right of the hallway with respect to the robot (i.e., the left with respect to the person).

These locations are shown in Figure 5.2. Furthermore, the person moved at one of three speeds (slower than, faster than, or the same speed as the robot). The person locations were chosen so that the robot could physically pass on either side of the person in all cases. Each person location and speed combination was used with each of the three robot goals, yielding 27 possible scenarios. Pictorial results from all scenarios are presented in Appendix B; here we present statistics and some key aspects.

In all cases, the robot's plan required it to move out of the way of the person, keeping a minimum of 0.41 meters away (average 1.13 m), as measured robot-center to person-center (as both robot and person have width, the actual free space between them would be smaller—on average about 0.75 m side-to-side). In 18 out

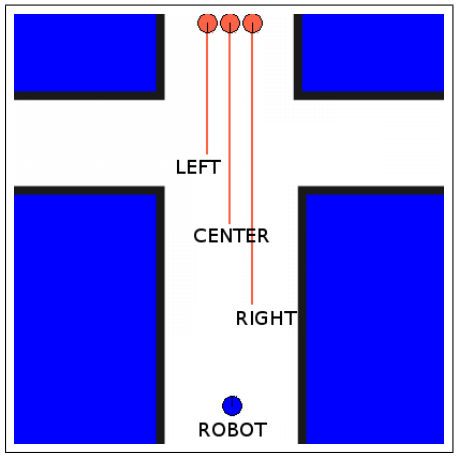


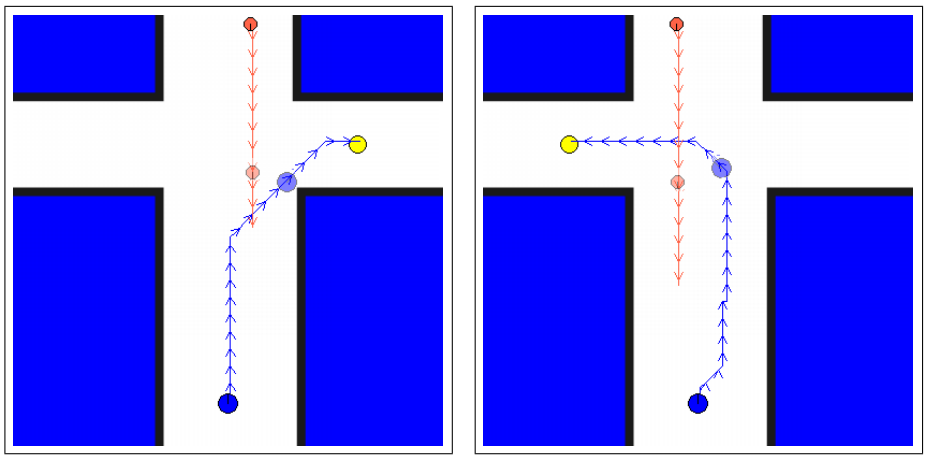
Figure 5.2: Three possible starting locations for the person. Note that the location names are given with respect to the robot's starting location and orientation (bottom of the hallway, facing up), rather than with respect to the person's orientation.

of 27 cases (67%), the robot stayed to the person's left; that is, the robot stayed to its own right in the hallway, which corresponds to "typical" human social behavior. In the remaining 9 cases (33%), the robot crossed to the person's right (i.e., the robot moved to the left of the hallway). These cases occurred under the following situations:

- The robot must turn left, and has sufficient time to make the turn before encountering the person—*2 cases*;
- The robot must turn (right or left), but a person is traveling quickly toward the robot on the "wrong" side of the hallway (i.e., on the robot's right)—*4 cases*;
- The robot must travel straight, but a person is traveling toward the robot on the "wrong" side of the hallway—*3 cases*.

In the cases where the robot moved to the left of the hallway, it also kept a greater side-to-side distance to the person (minimum 0.70 m, mean 1.22 m). Intuitively, all of these cases correspond to "reasonable" social behavior: either the robot crosses to the left side in advance of a turn, as people often do (Bitgood and Dukes, 2006), or the robot moves to its left in reaction to the person moving along the "wrong" side of the hallway.

Two interesting cases are shown in Figure 5.3. These two scenarios are mirror images of each other; in Figure 5.3(a), the robot's goal requires a right turn, and the



(a) Goal requiring a right turn, with a person to the robot's right.

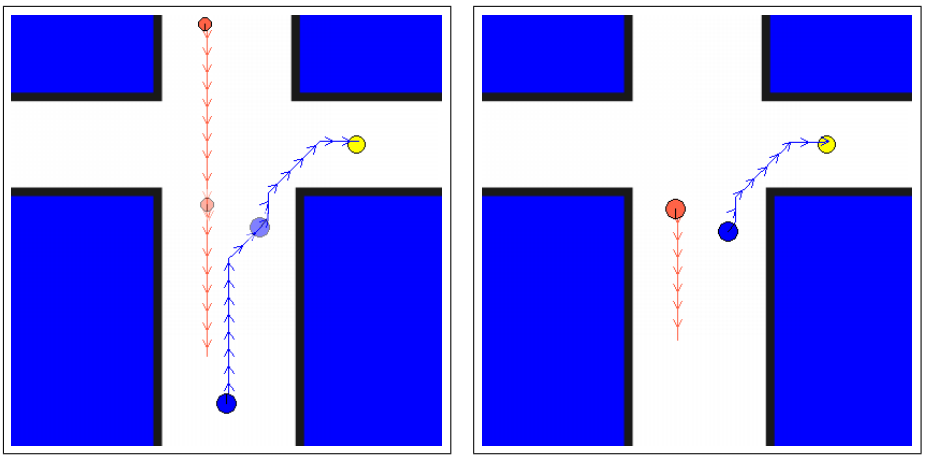
(b) Goal requiring a left turn, with a person to the robot's left.

Figure 5.3: The two scenarios pictured here are mirrored. In both cases, the person is moving at 0.3 m/s. Because of the asymmetric "tend to the right" constraint, the robot's paths differ markedly. The points at which the robot and person are closest on the path are marked.

person is traveling on the robot's right; while in Figure 5.3(b), both the goal and the person are on the robot's left. In both, the person is traveling at 0.3 m/s. However, the paths generated are not symmetrical, due to the "tend to the right" constraint. That is, the robot keeps to the right side of the hallway, despite the option for a shorter path, which would put the robot to the socially "wrong" side of the person. Of further interest in Figure 5.3(b) is that the actions the robot chooses to shift to the right side of the hallway are sideways holonomic moves, rather than turns.

Another interesting behavior can be seen in Figure 5.4. In this situation, the robot's goal is on the right, and a person is walking on the robot's left at a speed of 0.5 m/s. As the robot approaches the person, it plans a $45°$ turn toward the goal, at a point slightly earlier than would allow the robot to pass the hallway corner safely. It then keeps the same orientation but moves straight up the hallway—that is, driving sideways—for approximately one-half meter, during which time it passes the person. This behavior is due to the "robot 'personal' space" constraint. The robot prefers to drive along a sideways angle for a short distance rather than let the person enter the robot's "personal" space. This is similar to a person angling her shoulders away from someone as they pass each other.

Figures 5.5 and 5.6 indicate how the paths generated around people differ from the baseline paths (which were shown in Figure 5.1). These graphs aggregate the



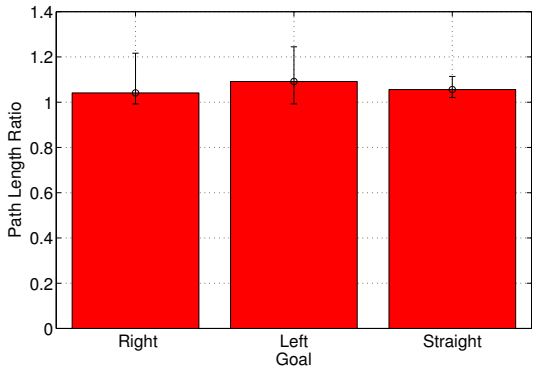
(a) Overall path.

(b) Passing the person: the robot is angled toward the goal but is driving straight along the hallway.

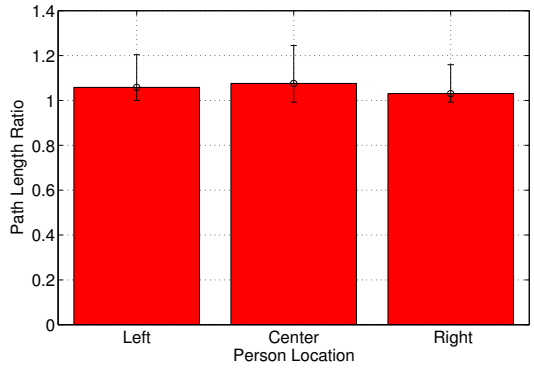
Figure 5.4: An interesting holonomic behavior. The robot turns and drives straight at a $45°$ angle, then keeps the same orientation but drives sideways, straight up the hallway, for a brief period before continuing along to the goal.

27 cases according to either the goal location, the person's location, or the person's speed. From Figure 5.5, which shows the change in path length as a ratio versus baseline, we see that all conditions caused about the same increase in path length. Interesting to note is that two cases actually caused the robot to take a slightly *shorter* path than the baseline; this is because the robot chose to travel closer to the hallway corner to avoid the person, incurring a larger obstacle buffer cost but a shorter overall path.

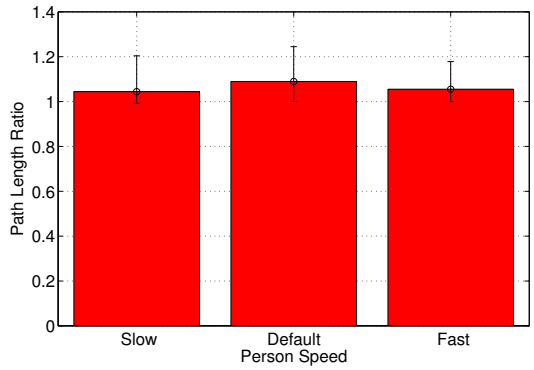
Figure 5.6 shows the change in actions selected versus the baseline cases. The actions are grouped by type: changes in v_x, changes in v_y, and changes in v_θ, which correspond to the "default velocity," "face direction of travel," and "inertia" constraints, respectively. The largest changes were sideways maneuvers (changes in v_y), particularly when the goal was straight ahead. With respect to the person's location in the hallway, the robot typically avoided a person on its left using more sideways maneuvers than velocity changes or turns, whereas other people were avoided with all three possible types of actions. In general, the robot planned to side-step people on the left (socially expected) side, while it preferred to drive quickly out of the way of people centered or on the right. Similarly, the robot avoided slower moving people with primarily sideways maneuvers, but used a combination of different actions to avoid faster people.



(a) By goal location

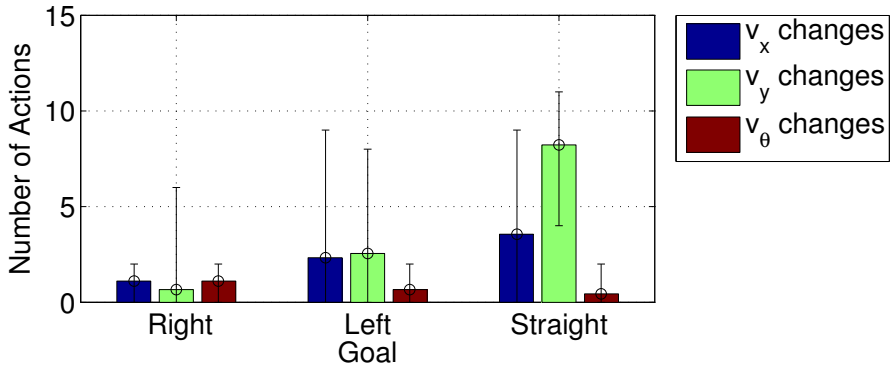


(b) By person location

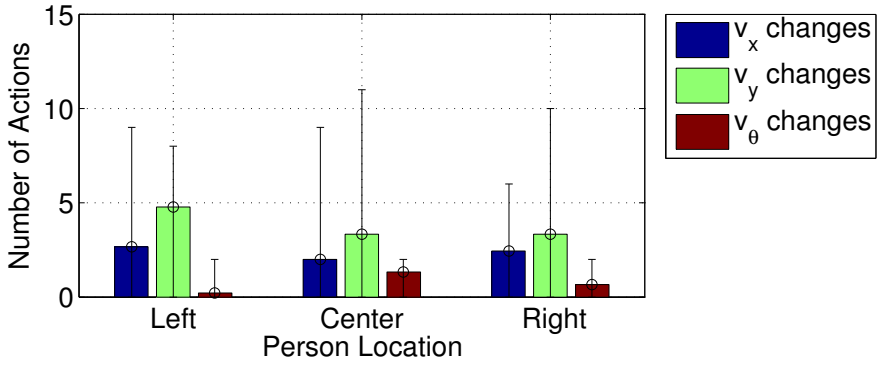


(c) By person speed

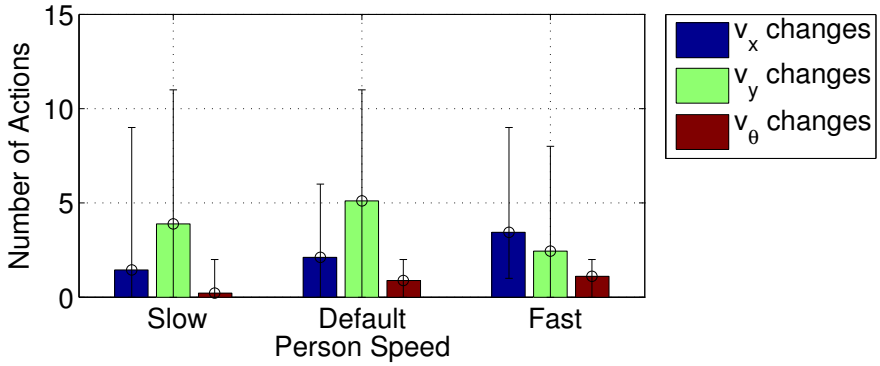
Figure 5.5: Ratio of path length required to travel around a person versus optimal path to goal with no person. Error bars indicate minimum and maximum values.



(a) By goal location



(b) By person location



(c) By person speed

Figure 5.6: Change in types of actions due to planning around a person, versus optimal path to goal with no person. Error bars indicate minimum and maximum values.

Table 5.3: Search times and node expansions required for the 27 test cases using different speed-improving techniques. Techniques include: variable grid (VG), reducing the action space (ActReduce), ignoring people behind the robot (Ignore), and searching on a gradient (Gradient).

Search techniques	**Search Time (s)**			**Nodes Searched**		
	Min	*Max*	*Avg*	*Min*	*Max*	*Avg*
None	7.33	134.2	61.4	52,324	463,357	253,730
Variable Grid (VG)	0.11	4.13	1.03	1037	35,135	10,901
VG + ActReduce	0.06	2.71	0.65	1031	29,650	9524
VG + Ignore	0.08	2.46	0.80	1024	23,303	8789
VG + Gradient	0.06	0.75	0.25	905	7864	2995
All	0.05	0.30	0.12	889	4770	2068

On average, these plans required searching $253,730$ nodes (minimum $52,324$, maximum $463,357$), with search times averaging over to one minute (mean 61.4 s, minimum 7.33 s, maximum 134.2 s). Obviously, these search times are unacceptable for real-time navigation. However, by implementing the various techniques discussed in Section 4.4.2, the search times can be reduced drastically. Table 5.3 shows the results of implementing the different speed-improvement techniques. Simply implementing a variable grid resulted in a large speed improvement, but with searches still requiring 1 s on average. Adding any or all of the additional techniques—reducing the action space at larger grid cells, ignoring people behind the robot, and searching only on a gradient toward the goal—provided additional speed increases. With all techniques, even the slowest search ran in under one-half of one second. See Appendix B for the difference in results between the default and the faster searches. In general, the paths planned using these techniques were similar (though not identical) to the optimal paths, particularly with respect to the first actions, as desired.

5.1.2 Alternate constraint weights

As discussed in Chapter 4, different behaviors can be achieved by altering the relative weights between the different constraints.

Changing the constraint weightings can be viewed as changing the robot's "personality." For example, by increasing the weight of the "tend to the right" constraint, the robot becomes deferential almost to the point of awkwardness, taking overly long paths to avoid passing a person while on the left side of a hallway. As

a second example, changing the relative weights of the velocity-based constraints can reduce similar deferential behavior by causing the robot to prefer side-stepping around people, rather than turning toward the wall. Both of these examples are further described below.

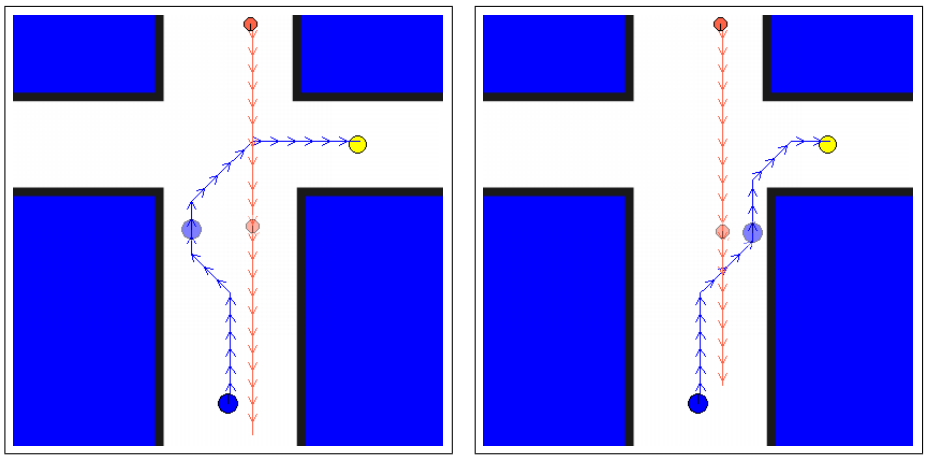
Always tending to the right

Greatly increasing the weight of the "tend to the right" constraint will cause the robot to have a much greater tendency to stay to the right of the hallway when passing a person, even if the space is narrow. All 27 scenarios from the previous section were run a second time, using the weights listed in Table 5.1 except for the "tend to the right" constraint, which was given a weight of 10.

With this weighting, all 27 cases resulted in the robot staying to the right to pass the person. Figures 5.7 and 5.8 demonstrate two of these cases. In Figure 5.7, we see a case where the robot initially (using the weights from Table 5.1) chose to move left to avoid a person traveling on the right side. That is, both parties traveled on the socially "incorrect" side of the hallway. By increasing the weight on the "tend to the right" constraint, the robot instead chose to stay to the right of the hallway, despite incurring higher costs from the "personal space," "robot 'personal' space," and "obstacle buffer" constraints. Figure 5.8 shows a case where the robot could easily turn in front of the person to get to the goal, but doing so caused the robot to pass the person on the left side of the hallway. Increasing the "tend to the right" weight caused the robot to take a significantly longer path to prevent this situation.

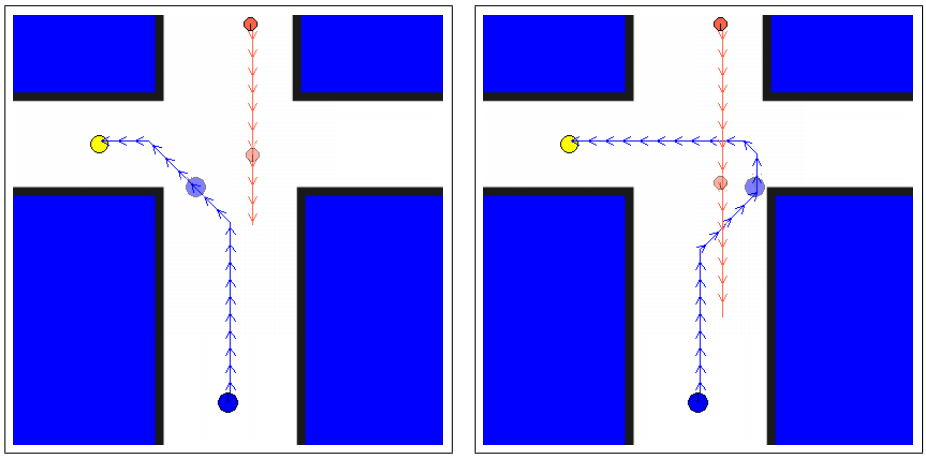
Sidestepping versus turning

As a second example of the effects of the constraint weights, consider the three constraints of "default velocity," "face direction of travel," and "inertia." Together, these place constraints on the robot's three velocities (v_x, v_y, and v_θ, respectively). Given the weights of these three constraints as defined in Table 5.1, the robot may turn away from an approaching person, such as demonstrated in Figure 5.9(a), which may appear awkward. If we reduce the cost of the "face direction of travel" constraint relative to the other two (e.g., by setting $w_{facing} = 1$ and $w_{inertia} = w_{default-v} = 3$), we force the robot to side-step the person instead, as shown in Figure 5.9(b). As we will argue in Section 5.2, this may be an important modification for a robot to be seen as socially appropriate.



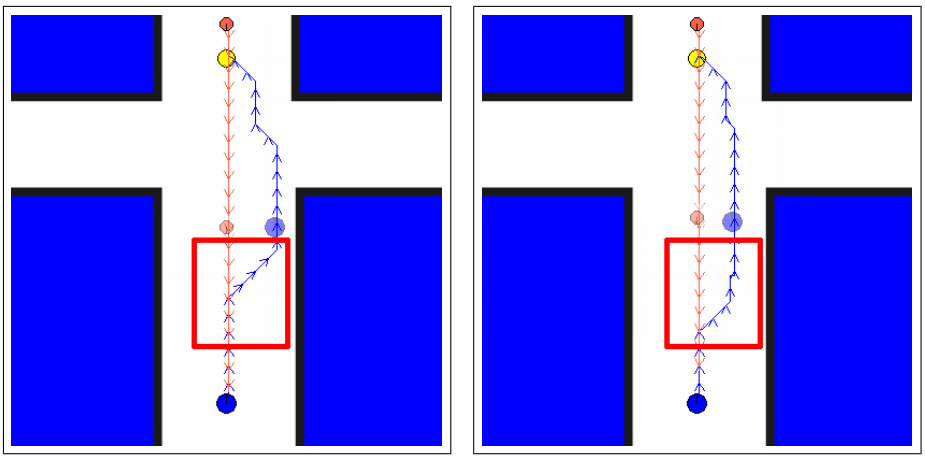
(a) Initial constraint weights (b) Increased tend-to-right

Figure 5.7: The path shown in (b) differs from that in (a) because it was generated with a higher weight on the "tend-to-the-right" constraint. Path (b) is also shorter than (a), but causes the person and robot to intrude further on each other's personal space.



(a) Initial constraint weights (b) Increased tend-to-right

Figure 5.8: Although the robot has space to turn in front of the person in this scenario (a), increasing the weight for the "tend-to-the-right" constraint results in the robot going far out of its way to keep to the "socially correct" side of the person (b).



(a) Initial constraint weights: the robot turns when moving away from the person.

(b) Decreased face-travel and increased inertia and default-velocity: the robot always faces straight down the hallway, moving sideways to avoid the person.

Figure 5.9: By changing the relative weights of the "face direction of travel," "inertia," and "default velocity" constraints, the robot can be made to always side-step a person, rather than turning to drive around. The areas outlined in red highlight this difference.

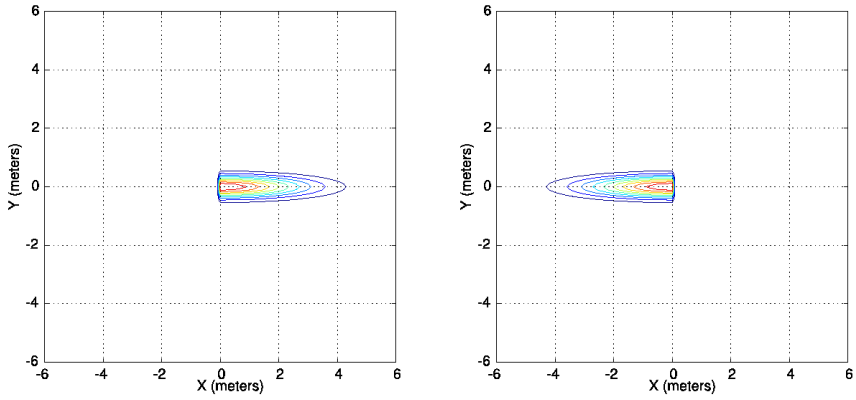
5.1.3 Different cultural norms

As discussed in Chapter 2, many human social conventions are culturally defined. The constraints defined in Chapter 4 are all intended to match the conventions used in the United States.

One such convention that differs across cultures is that of which side of the hallway people prefer when passing others. Figure 5.10 shows how the "pass on the right" constraint would need to be modified to represent preferring to pass on the left side, instead. Quite simply, the cost is mirrored; the Gaussian cost function is aligned to the person's left, $\theta_p + \pi/2$.

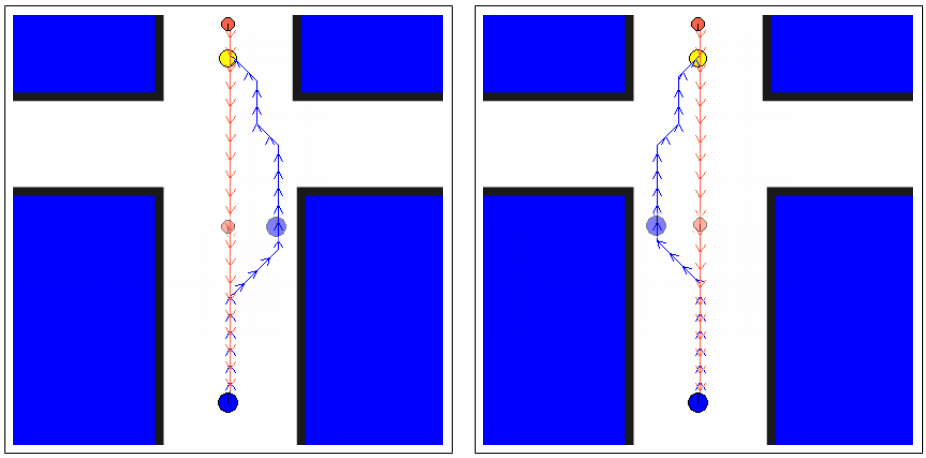
Figure 5.11 shows an example of how paths differ between the "pass on the right" and "pass on the left" constraints. These two paths use the same constraints and weights, except Figure 5.11(a) uses the "pass on the right" constraint while Figure 5.11(b) uses the "pass on the left" version. As expected, the two paths are mirrored.

Other ways in which social conventions vary across cultures, such as differently-sized personal space or different use of gaze, are addressed in Appendix C.



(a) Pass on the right (b) Pass on the left

Figure 5.10: Constraints for preferring to pass a person on the right versus on the left. In each case, the cost function displayed is for a person centered at (0,0) and moving along the positive Y-axis (up).



(a) Passing on the right (b) Passing on the left

Figure 5.11: Passing a person on the right versus on the left. In Figure (a), the robot adheres to the "pass on the right" constraint. In Figure (b), that is replaced with a mirrored "pass on the left" constraint. The resulting paths are mirror images of each other.

5.1.4 Other examples

While the set of trials presented above describes a wide range of the robot's behavior in head-on encounters, we present here some additional examples that demonstrate its behavior in other situations, namely, overtaking a slower person and navigating around crowds.

Overtaking a slower person

Consider the case where the robot must overtake a person who is heading in the same direction, but at a slower speed than the robot (0.2 m/s versus the robot's 0.5 m/s). This situation is demonstrated in Figure 5.12. The constraints used are those given in Table 5.1. For people on the right or centered in the hallway, the robot plans a path to the left, around the person, as is the social norm in human–human interaction. However, if the person is traveling up the left side of the hallway, the robot stays to the right, where there is sufficient space to pass.

Crowd navigation

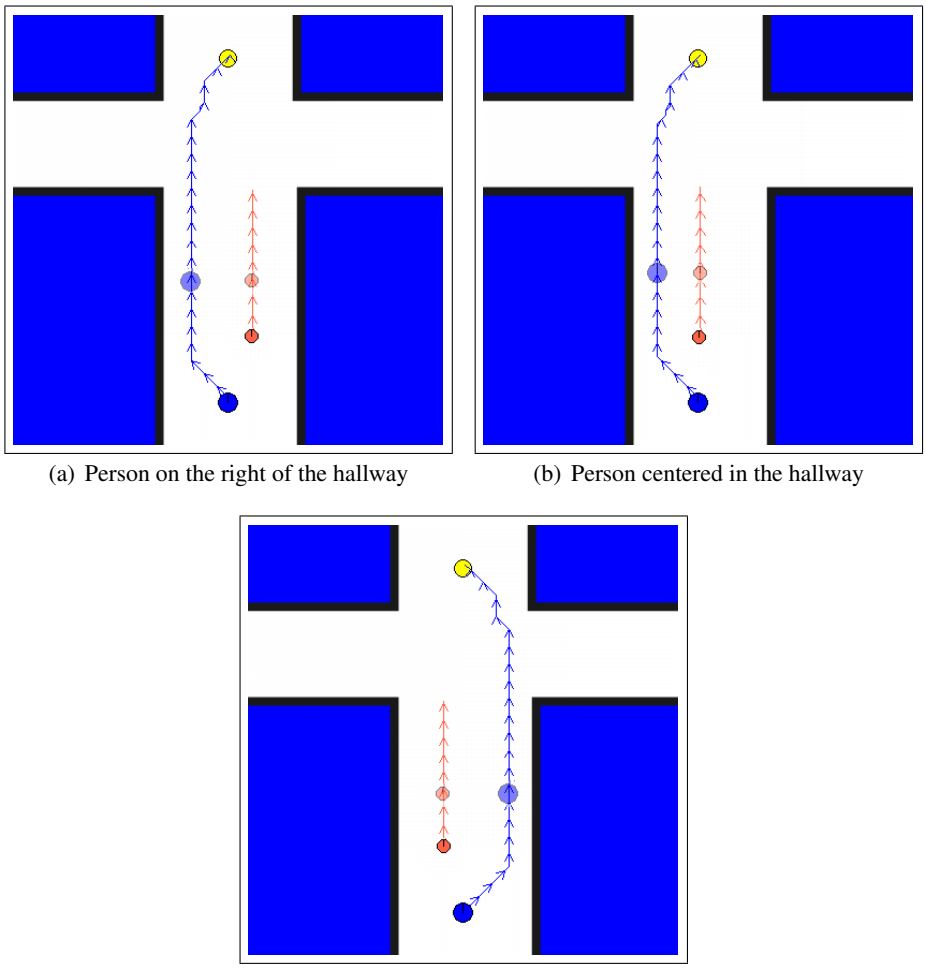
We now present some results of the planner's behavior around multiple people. Because additional people increase the size of the state space significantly, these paths were planned using the variable grid described in Section 4.4.2. The constraints used were the same as given in Table 5.1.

These tests use the more complex office-style map shown in Figure 5.13(a). Consider the scenario in which the robot begins in the lower left-hand corner of the map, and must plan a path to an office door near the top right. Initially, no people are visible; the optimal path is shown in Figure 5.13(b). We simulate the results of the robot detecting various people after it enters the center corridor.

Figure 5.14 shows some example paths planned around two people. Both of these represent minor deviations from the initial path.

- In Figure 5.14(a), the robot detects two stationary people who are facing each other (perhaps having a conversation). Though the robot would fit between the two, it chooses the narrower opening between the wall and the pair, rather than incur a cost from passing through both of the people's personal space zones. Since the people are stationary, the "pass on the right" constraint is not relevant in this case.

- In Figure 5.14(b), one of the two people from the previous encounter is moving toward the robot. The robot is able to plan a complex path around both people. For the person who is moving, the robot avoids his personal space

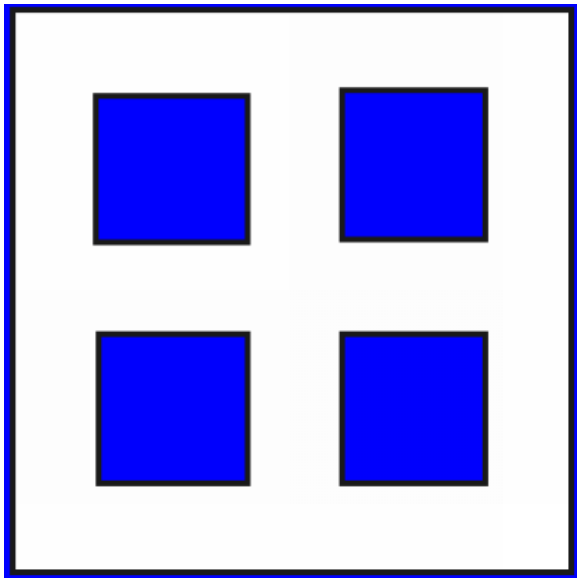


(a) Person on the right of the hallway

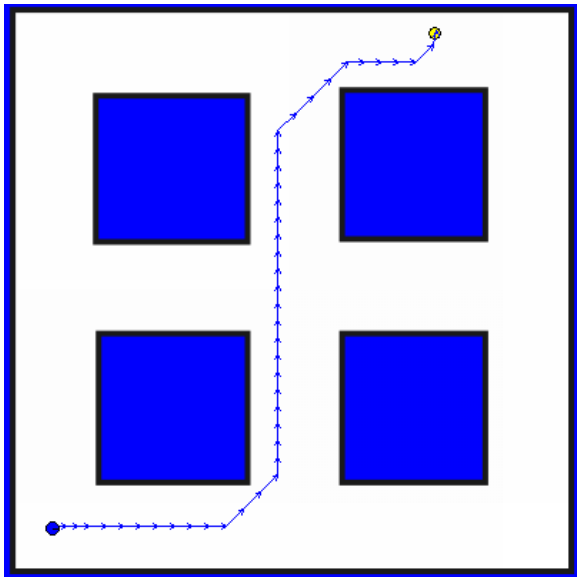
(b) Person centered in the hallway

(c) Person on the left of the hallway

Figure 5.12: Paths planned for the robot overtaking a single person, who is headed in the same direction as the robot but at a slower speed (0.2 m/s). As with human social conventions, the robot prefers to pass the person on the left, except in the case of the person who is already on the left side.

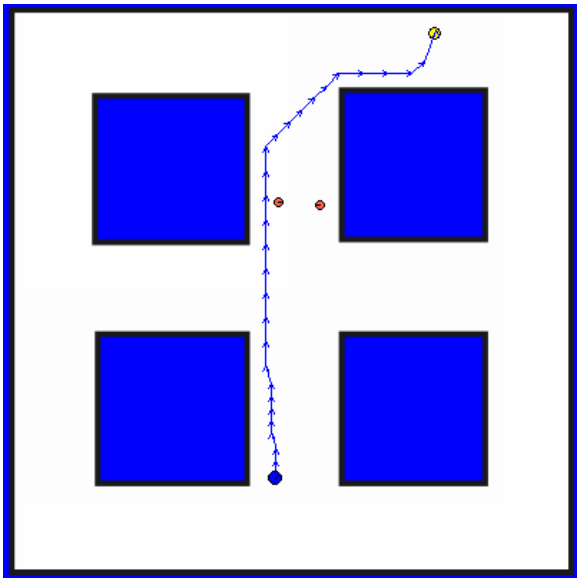


(a) A simplified office environment.

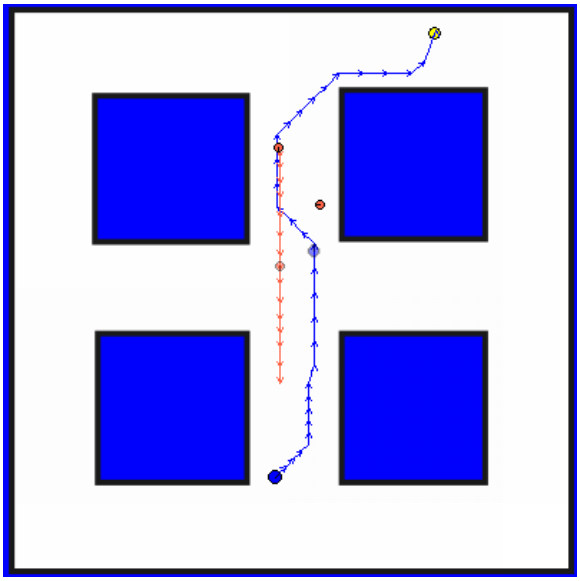


(b) A path within the office environment.

Figure 5.13: An office map and a path through the environment. This environment is 20 m by 20 m, and all hallways are 3 m wide.

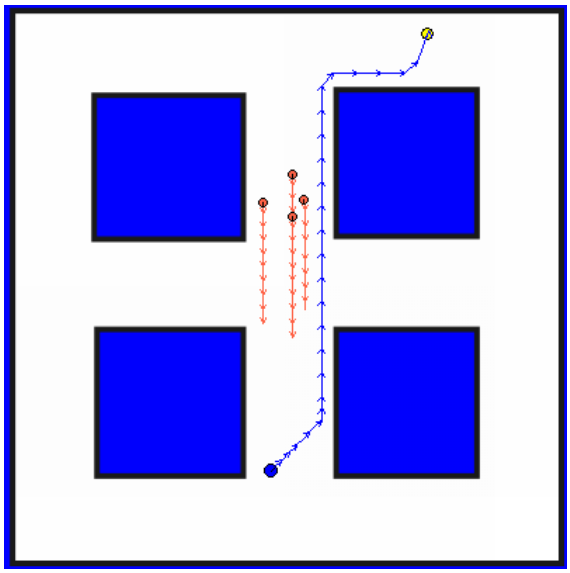


(a) A path around two stationary people; the robot avoids the overlapping personal space region between them.

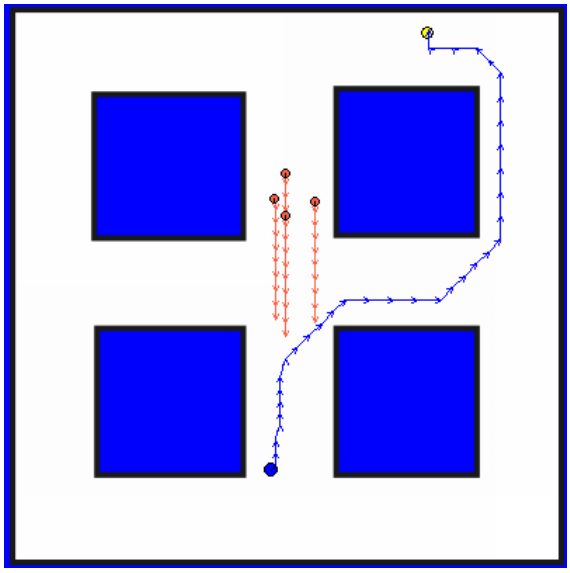


(b) A path around one moving person and one stationary person.

Figure 5.14: Paths planned around two people in the environment.



(a) People clustered on the left of the hallway; the robot passes on the right.



(b) People clustered on the right of the hallway; the robot chooses a longer path around to the goal.

Figure 5.15: Paths taken to avoid a slow-moving group of people. In (b), the cost of taking a longer route is less than that of passing four people on the left side of the hallway.

and stays to the right side of the hallway. For the stationary person, the robot moves to the left side of the hallway to avoid his personal space.

Finally, in the scenario in Figure 5.15, the robot encounters a large, slow-moving group of people.

- In Figure 5.15(a), the group is primarily on the left side of the hallway (with respect to the robot), so the robot passes the group on the right.
- In Figure 5.15(b), however, the group is primarily on the right of the hallway. Rather than pass next to this group (i.e., using the same path found in Figure 5.14(a)), which would incur a "tend to the right" cost for each of the four people, the robot instead chooses a much longer path that completely avoids the people. While this path is longer, note that it is still optimal according to the planning framework.

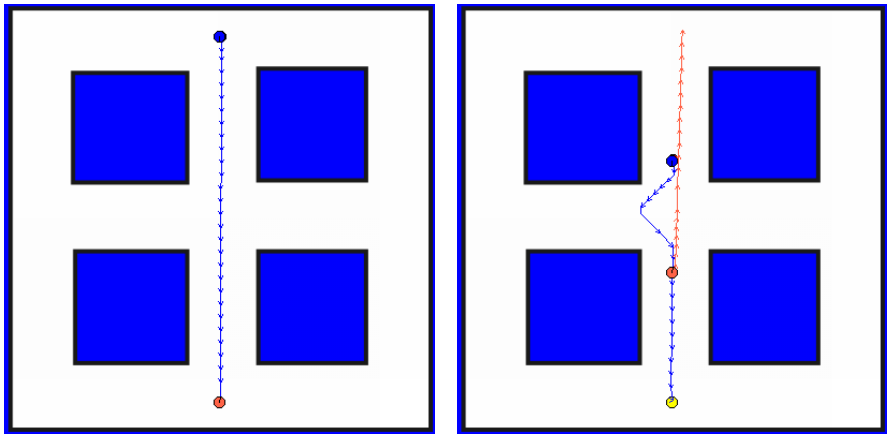
5.1.5 Navigation

All of the above examples show statically planned trajectories. The paths shown will be the robot's actual trajectory only if the person (or people) do not deviate from their trajectories and if the robot's kinematics allow it to follow the planned trajectory exactly. In practice, both of these assumptions are likely false. Thus, we wish to test the complete system of planning, navigation, and people-tracking. To simulate this repeatably, we would need a "social person" simulator. Instead, we present the scenario of two *social robots* encountering each other. If each robot behaves according to human social norms, then we would expect that the overall behavior should be that of two people encountering each other.

Using CARMEN, we ran two complete simulations of separate robots in the same environment. By linking the simulations, each robot was able to detect the other robot as though it were a person in the environment. Unfortunately, CARMEN cannot currently handle holonomic movements (the localization model fails[1]), so we reduced the action set available to the robot to only non-holonomic actions (see Section 4.4.1). In addition, we used the variable grid described in Section 4.4.2. However, we allowed all actions to be used at all grid sizes, and we did not force the search to follow the shortest-distance gradient. Since the planner still ran in sub-real-time, the simulator was run at $1/10^{th}$ real time.

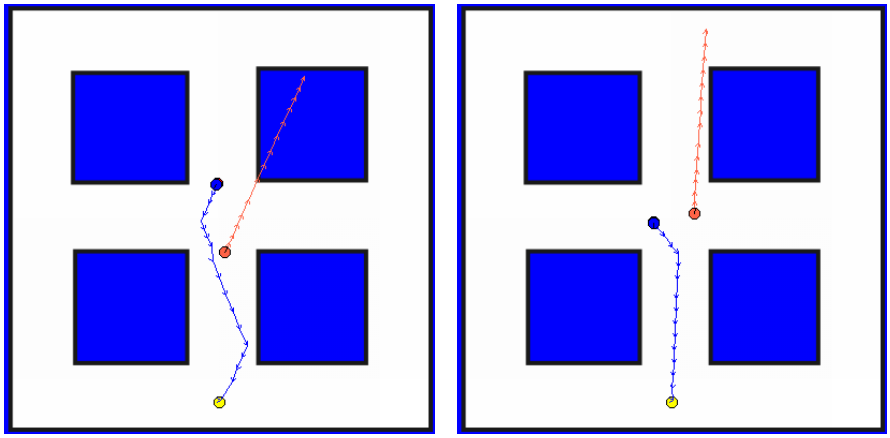
Since no holonomic actions were used, we removed the "face direction of travel" constraint. Additionally, we reduced the weight of the "inertia" constraint

[1]CARMEN's localization model, which is derived from Eliazar and Parr (2004), assumes that sideways movements occur from slippage only.



(a) Before the robots detect each other.

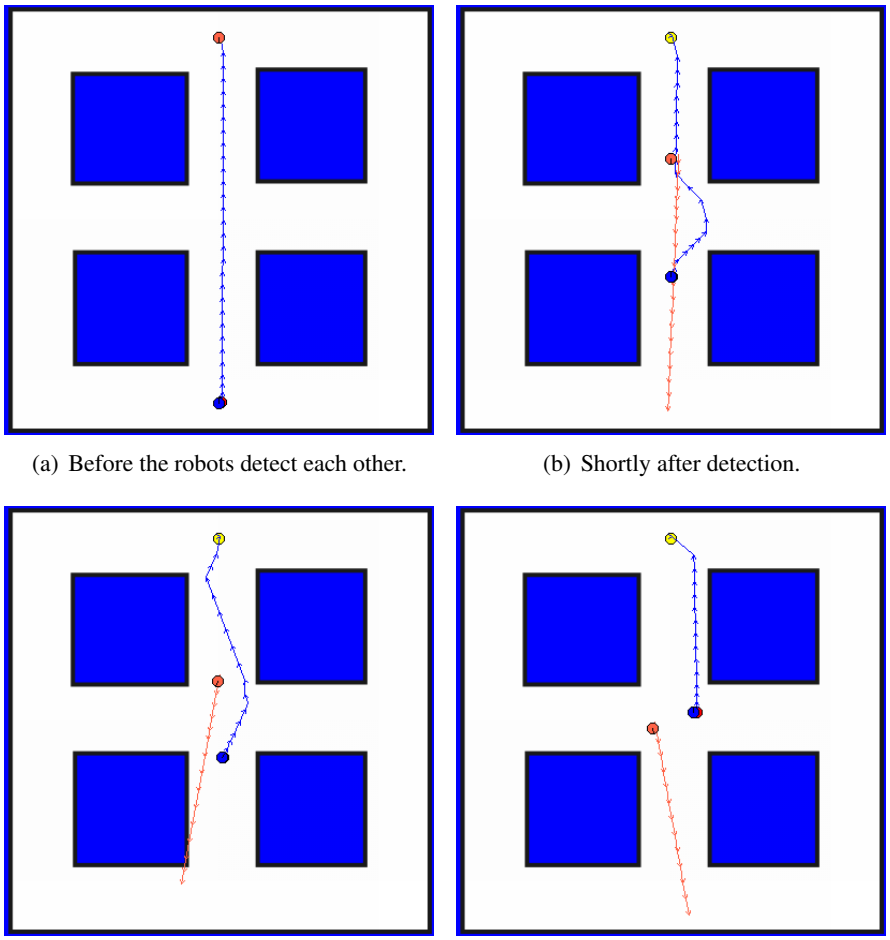
(b) Shortly after detection.



(c) After each robot starts to move out of the way.

(d) While passing each other.

Figure 5.16: Running two simulators against each other, from the perspective of the top robot (blue; second robot in orange). Both are using the same set of constraints and weights. Neither robot is aware of the other's planned path or desired goal. The second robot is detected and tracked as if it were a person, and is predicted to continue along straight trajectories, without regard for obstacles. Because the top robot assumes the other will not move out of its way, it initially chooses a longer path to stay away (b). Once each robot begins to move away, however, the robot determines that it can safely pass the other with less deviation from its own path (c).



(a) Before the robots detect each other.

(b) Shortly after detection.

(c) After each robot starts to move out of the way.

(d) While passing each other.

Figure 5.17: Running two simulators against each other, from the perspective of the bottom robot (blue; second robot in orange). Both are using the same set of constraints and weights. Neither robot is aware of the other's planned path or desired goal. The second robot is detected and tracked as if it were a person, and is predicted to continue along straight trajectories, without regard for obstacles. Because the bottom robot assumes the other will not move out of its way, it initially chooses a longer path to stay away (b). Once each robot begins to move away, however, the robot determines that it can safely pass the other with less deviation from its own path (c).

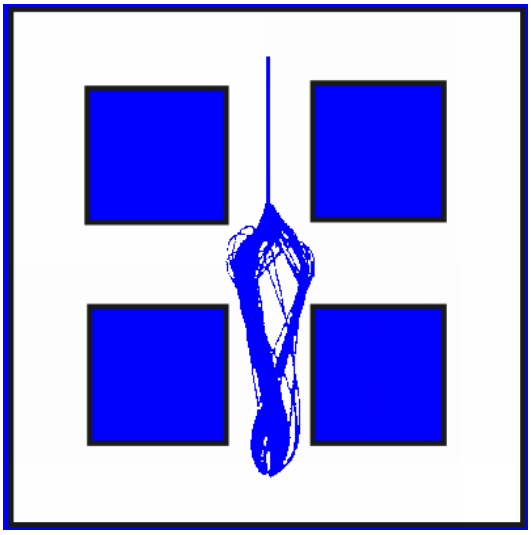


Figure 5.18: Actual trajectories taken by a simulated robot that started at the top of the map and drove toward the bottom, encountering a second robot near the hallway intersection. In the majority of trials, the robot moved to its right to avoid the other (as is socially expected). 100 paths in total.

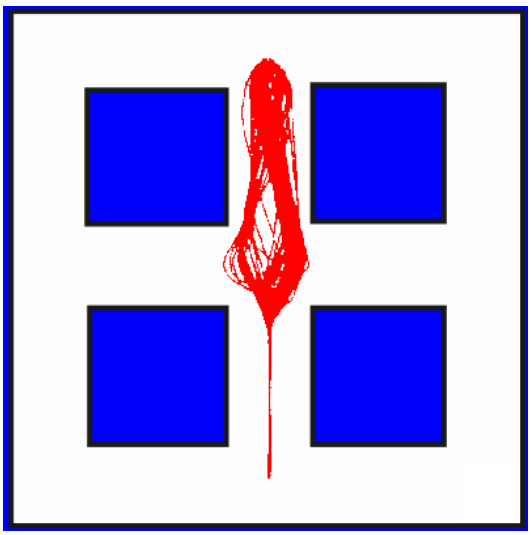


Figure 5.19: Actual trajectories taken by a simulated robot that started at the bottom of the map and drove toward the top, encountering a second robot near the hallway intersection. In the majority of trials, the robot moved to its right to avoid the other (as is socially expected). 100 paths in total.

to 1, as it was no longer competing with the "face direction of travel" constraint. Otherwise, the constraints used are those given in Table 5.1.

The scenario we used was of two robots at opposite ends of a corridor, as shown in Figures 5.16 (first robot) and 5.17 (second robot). Neither robot has knowledge of the other's path or goal. The two simulated robots were directed to switch places. As the simulated lasers had a detection range of only 6 m (equivalent to a Hokuyo URG laser), the robots were not able to detect each other until both had traveled part of the way through the corridor (see Figures 5.16(b) and 5.17(b)). At this first detection, each robot assumes that the other will not yield, and thus plans a large path deviation around the other. However, once each robot begins to move (Figures 5.16(c) and 5.17(c)), both robots are able to reduce their avoidance maneuvers.

This scenario was run 100 times. In 81 of the trials, both robots moved to their respective right sides, passing each other in the typical human-like way. In 18 trials, both robots moved to their left sides. In the remaining trial, both robots failed to properly detect the other, resulting in a collision. The trial-to-trial differences resulted from several probabilistic elements in the complete system: namely, the localization and the person-tracking modules. In particular, the person-tracking module does not always compute the accurate heading of the detected person (or other robot, in this case), which can result in one robot beginning to move to the left rather than right—which, if correctly detected by the other robot, causes it to also move left. The complete sets of paths taken by the robots are shown in Figure 5.18 (top robot) and Figure 5.19 (bottom robot).

5.2 User study

The simulation results presented above demonstrate that the COMPANION framework produces paths that observe the conventions, as we defined them. We also ran a user study to verify whether the paths are seen by people as socially appropriate. We tested the robot's behavior under the social conventions described in Chapter 4 in comparison to a "non-social" behavior produced by the same framework using only task-based constraints.

Our hypotheses included the following:

H1 Participants will perceive a difference between the two robot behaviors and will more highly rate the "social" behavior on scales of human-likeness and adhering to social conventions.

H2 Participants will feel more empowered with respect to the "social" robot behavior, since the "non-social" behavior will approach them more closely.

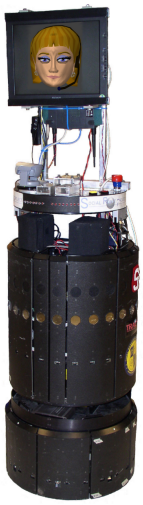


Figure 5.20: The robot Grace, as used in the hallway navigation study.

H3 Participants will demonstrate higher positive affect and lower negative affect with respect to the "social" behavior versus the "non-social" behavior.

5.2.1 Implementation details

For this study, we used the robot Grace, which was described in Section 3.2.2 and is shown in Figure 5.20. Grace is a B21 robot built by RWI. We added an additional on-board computer to the robot to perform the path-planning; the computer runs a quad-core Pentium processor at 2.4 GHz and contains 4 GB of RAM. The robot's base has only two degrees of freedom (forward translational velocity and rotational velocity); as such, the robot is not capable of instantaneous sideways movement.[2] Thus, for social behavior, we used the same set of constraints used in running two simulations against each other (Section 5.1.5, and repeated in Table 5.4). To produce a "non-social" behavior, the social conventions of "personal space," "robot 'personal' space," and "pass on the right" were removed (that is, their respective weights set to zero). Since the hard person-avoidance constraint was still used, the robot's "non-social" behavior was to move out of a person's way only if the person did not do so first, and to keep on a straight-line path otherwise.

In order to run the user study in real-time, we used several of the techniques described in Section 4.4.2. In particular, we used a variable search grid as defined

[2]Though we had already begun design of the holonomic robot Companion (see Chapter 7), the new robot was still in progress at the time this study was run.

Table 5.4: Constraint weights used on the robot Grace. The hard constraints of avoiding obstacles and people were also used.

	Weight	
Constraint Name	*Social*	*Non-social*
Minimize distance	1	1
Obstacle buffer	1	1
Personal space	2	0
Robot space	3	0
Pass on right	2	0
Default velocity	2	2
Face travel	0	0
Inertia	1	1

Table 5.5: Variable search grid sizing for use on Grace.

Distance from Robot	**Cell Dimensions**
less than 1 m	0.1×0.1 m
between 1 and 3 m	0.3×0.3 m
greater than 3 m	0.6×0.6 m

in Table 5.5, as well as requiring the search to follow a gradient toward the goal. Throughout the course of the user study, these techniques allowed the robot to generate a plan in under 0.2 s for 99.1% of its searches ($15,232$ searches performed in total; average time 0.02 s, maximum 1.77 s, SD 0.05 s). Only 5 paths required more than 1 second to plan, and in each case the tracker had mistakenly reported additional people in the environment.

Through pre-testing, we found that the person-tracking method described in Section 4.4 performed quite poorly in a real environment. In particular, since the tracker computes a person's speed and direction based on the change in position over time, the tracking has a hysteresis of 1–2 seconds. Since the laser has a limited detection range (roughly 8 meters for the SICK LMS), when the robot and person are moving toward each other, the person tracker typically cannot compute the person's speed and direction quickly enough for the robot to react as it would with perfect sensing. However, as the sensor problem is beyond the scope of this research, we configured the planner to use the computed position and speed of

Figure 5.21: Map view of the user study setup. In the first trial, the robot began at point 1 while the participant began at point 2; these positions were reversed for the second trial. The hallway is approximately 2.3 m wide, and the two points are approximately 7 m apart. A camera filmed each trial from behind the participant.

the person, but to assume that the person was traveling straight down the hallway. Since the "personal space" and "pass on the right" constraints are heavily dependent on position and direction, this assumption allowed the robot to better react to people despite the poor sensing.

5.2.2 Procedure

Participants were drawn from Carnegie Mellon University and the surrounding community, and were recruited through a combination of fliers, online bulletin boards, and word-of-mouth. They received candy in return for participation; no monetary incentive was provided. Participants were required to be 18 years of age, or older.

We used a within-subjects design, where each participant experienced both the "social" and the "non-social" robot behaviors, as described above. The order of trials was counterbalanced to minimize order bias. All participants were initially shown the robot driving away from them as it traveled to its initial position. Each trial required the participant to walk past the robot twice, down and back in a single hallway, which is approximately 2.3 m wide. The basic experimental setup is shown in Figure 5.21. Participants were asked to walk "however [they] feel the most comfortable." Additionally, they were asked to walk somewhat slowly, at a similar speed as the robot. As a safety precaution, participants were shown the location of the robot's "emergency stop" buttons and instructed on their use.

Participants were asked to complete a variety of surveys, including the Positive and Negative Affect Schedule (PANAS, see Table 5.6; Watson et al., 1988), which assesses a person's general emotional levels; the Self-Assessment Manikin

Table 5.6: The Positive and Negative Affect Schedule (PANAS). Participants were asked to "indicate to what extent you feel this way right now, that is, at the present moment" on a scale of 1–5, for each of the following items. From Watson et al. (1988).

interested*	irritable†
distressed†	alert*
excited*	ashamed†
upset†	inspired*
strong*	nervous†
guilty†	determined*
scared†	attentive*
hostile†	jittery†
enthusiastic*	active*
proud*	afraid†

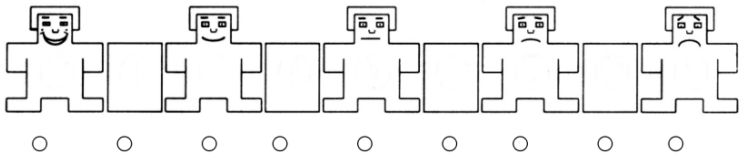
* Positive Affect scale
† Negative Affect scale

scales (SAM, see Figure 5.22; Bradley and Lang, 1994; Bethel et al., 2009), which assess a person's emotional reactions toward a robot; and several survey questions intended to determine how people understood the robot's behaviors in a social context (see Table 5.7). The PANAS was administered three times (before interacting with the robot and after each trial), while all other survey questions were administered twice, once per condition (after each trial). Additionally, participants were given a free-response question to solicit comments on each robot behavior.

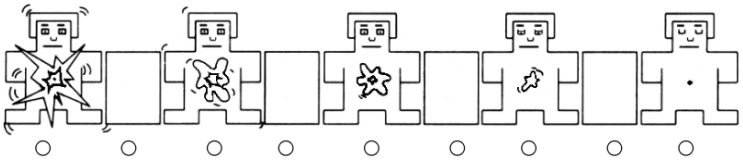
5.2.3 Results

A total of 27 people participated in this study (12 male and 15 female). Ages ranged from 18–35 years (mean 25.7). Participants had a wide range of self-reported prior robotics experience (mean 4.7, SD 1.95, on a scale of 1–7).

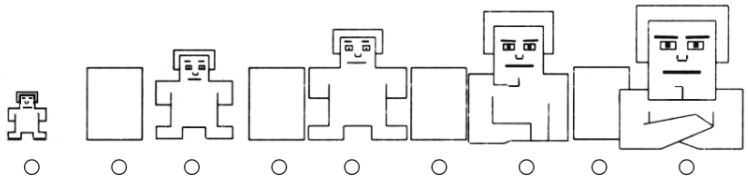
Figures 5.23 and 5.24 depict typical encounters with the robot, in the "non-social" and "social" conditions, respectively. In the "non-social" condition, the robot typically remained closer to the center of the hallway, only narrowly avoiding the person. In contrast, in the "social" condition, the robot typically turned away from the participant and drove closer to the wall. However, while these behaviors were common, we observed a great deal of variation between trials. In addition,



(a) Valence



(b) Arousal



(c) Dominance

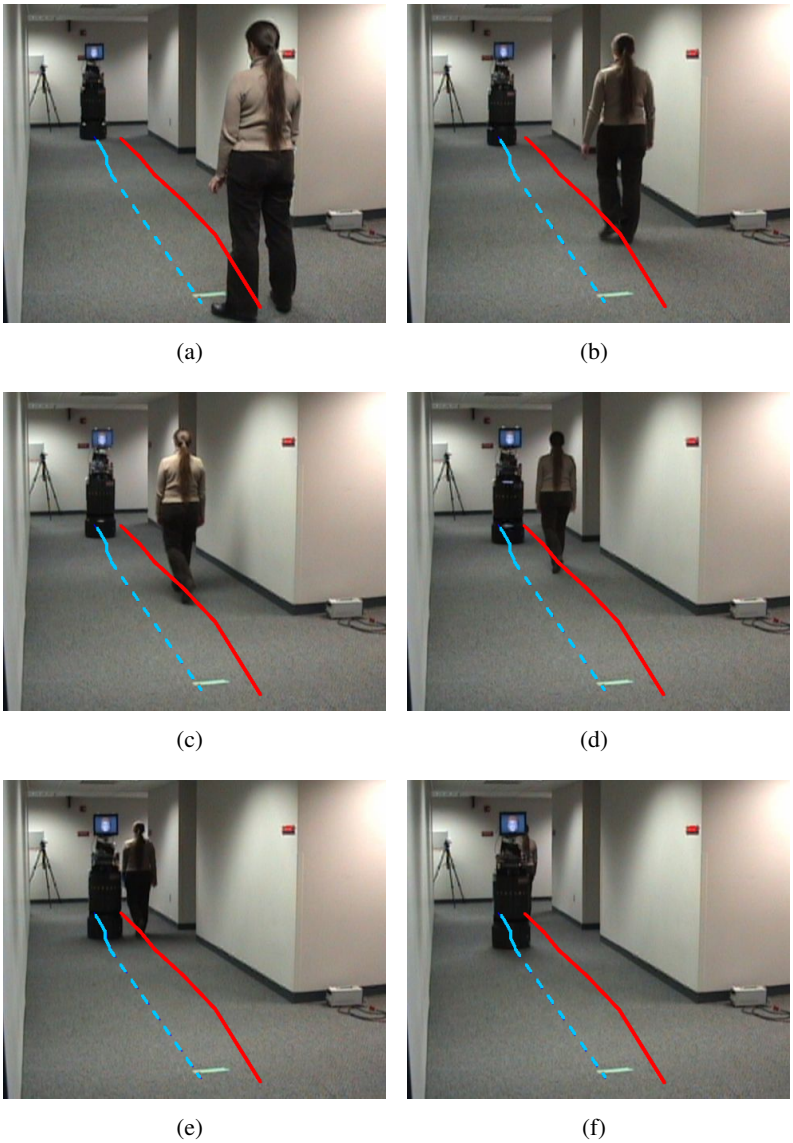
Figure 5.22: Images used for the Self-Assessment Manikin (SAM). Each image was presented twice for each scale, and participants were instructed to "mark the appropriate circle under each drawing that most closely reflects your feelings." From Bradley and Lang (1994).

several participants behaved counter to typical social norms, such as maintaining a straight path toward the robot or walking down the left side of the hallway.

We analyzed each set of survey questions for condition effects (that is, resulting from the "social" versus the "non-social" behaviors), as well as for effects of gender and experience with robots.

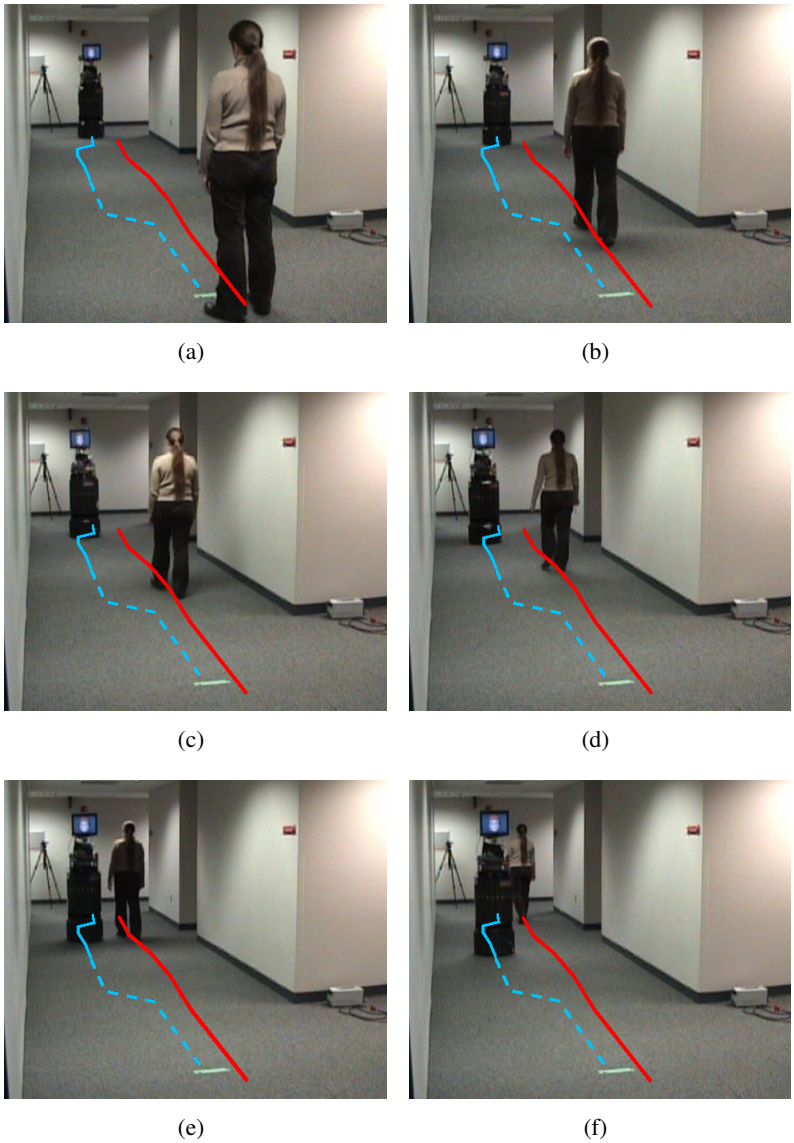
Positive and Negative Affect

Participants completed the Positive and Negative Affect Schedule (PANAS) three times: as a pre-test before encountering the robot, and after each robot behavior condition ("social" and "non-social"). Positive Affect (PA) and Negative Affect (NA) are distinct dimensions of mood; PA relates to social satisfaction and pleasant events while NA relates to subjective stress and unpleasant events. The two measures are generally uncorrelated, meaning that a person can have both PA and



(a) (b) (c) (d) (e) (f)

Figure 5.23: Participant walking past the robot in the "non-social" condition. Since the participant moves slightly to her right, the robot travels straight down the hallway with minimal deviation. The robot remains centered in the hallway and nearly touches the participant when they pass. The complete paths of the robot and person are overlaid in blue (dashed) and red (solid), respectively.



(a) (b) (c) (d) (e) (f)

Figure 5.24: Participant walking past the robot in the "social" condition. The robot turns toward its right (c), allowing more space between itself and the participant as they pass, and the robot approaches the wall more closely than in the "non-social" condition. The participant's path remains nearly straight. The complete paths of the robot and person are overlaid in blue (dashed) and red (solid), respectively.

Table 5.7: Survey questions asked of each participant after each robot behavior. All questions were asked on a 7-point scale from "Not at all" to "Very much." Bold-faced words were in the original, but scale titles were not included. N = 27.

General Robot Behavior Scale		*Cronbach's alpha = 0.85*
1	How **human-like** did the robot behave?	
2	How **social** was the robot's behavior?	
3	How **safe** did you feel around the robot?	
4	How **natural** was the robot's behavior?	
5	How **comfortable** did you feel near the robot?	
Robot Movement Scale		*Cronbach's alpha = 0.76*
6	How well did the robot's **movements** adhere to **human social norms**?	
7	How well did the robot **respect your personal space**?	
8	How well could you **anticipate** the robot's **movements**?	
9	How much did you have to **get out of the robot's way**?⋆	

⋆ scale reversed for analyses

NA—the measures are not simply opposites of a single scale. The PANAS was used to determine whether participants' general mood states changed after each encounter with the robot.

One-way analyses of variance (ANOVAs) of Positive Affect and Negative Affect by Condition showed no significant difference (PA $F = 0.46$, $p > 0.1$; NA $F = 0.68$, $p > 0.1$). Across all surveys, participants averaged a score of 2.85 in PA and 1.17 in NA, on scales of 1–5. That is, participants were generally enthusiastic (average PA) and minimally stressed (low NA).

Self-Assessment Manikin

The Self-Assessment Manikin (SAM) was administered after the robot encounter in each condition. The SAM is designed to measure three scales: valence, arousal, and dominance, each with respect to the robot. ANOVAs were run on each scale to look for differences between the two robot behavior conditions, the order of the trials, and the interaction between condition and order. No significant effects were found on any SAM scale (all $p > 0.1$). Across both conditions, participants averaged a valence of 6.27, arousal of 3.50, and dominance of 5.81, on scales of 1–9 (with 9 representing the highest value). That is, participants had medium levels of valence and dominance with low arousal.

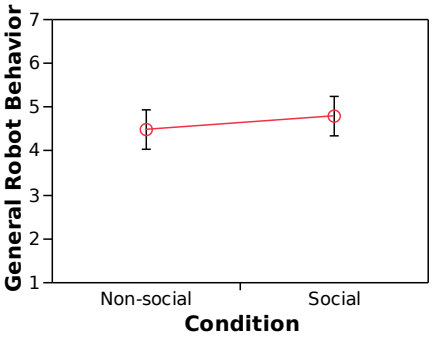


Figure 5.25: Results for the General Robot Behavior scale versus robot condition: $p > 0.1$ (errorbars indicate ± 1 std err).

Social Scales

Participants were given the nine questions, shown in Table 5.7, after their encounter with the robot in each condition. For analysis, we grouped the survey questions into two scales: the first measuring the overall robot behavior, and the second measuring more specific questions regarding the robot's movement. Both scales surpassed the commonly-used 0.7 level of reliability (Cronbach's alpha).[3] Each scale response was computed by averaging the results of the survey questions comprising the scale. ANOVAs were run on each scale to look for differences between the two robot behavior conditions, the order of the trials, and the interaction between condition and order.

On the "General Robot Behavior" scale, no effects were significant (all $p > 0.1$). The average rating on this scale was 4.80 for the "social" behavior and 4.50 for the "non-social" behavior, on a scale of 1–7. This is shown in Figure 5.25.

Analysis of the "Robot Movement" scale indicated a significant effect of behavior condition ($F = 9.76$, $p = 0.015$). Neither trial order nor the interaction of order and condition were significant (both $p > 0.1$). The average rating for the "social" behavior was higher than that of the "non-social" behavior (4.99 versus 4.14, on a scale of 1–7). This is shown in Figure 5.26.

Though the "Robot Movement" scale has high reliability, indicating that all four questions relate to a single measure, we wanted to understand which particular questions had the most influence on scale response. As such, we further analyzed the four individual survey questions that comprise the "Robot Movement"

[3]Cronbach's alpha is a measure used to determine how reliably a set of questions measures a single dimension. Values less than 0.7 imply that the scale is measuring more than one thing; higher levels indicate that the questions are essentially asking about the same thing, so the items can be combined for analysis.

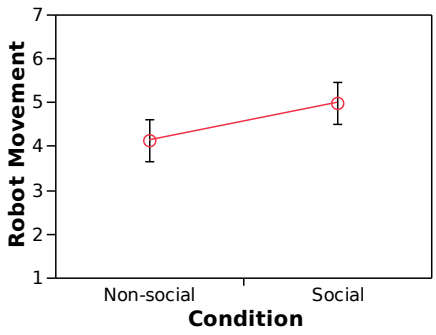


Figure 5.26: Results for the Robot Movement scale versus robot condition: $p = 0.015$ (errorbars indicate ± 1 std err).

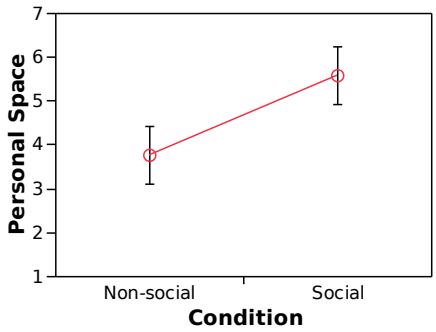


Figure 5.27: Results for "How well did the robot respect your personal space?" versus robot condition: $p = 0.0003$ (errorbars indicate ± 1 std err). Participants felt the "social" robot better respected their personal space.

scale. As with the complete scale, an ANOVA was run on each question to look for differences between the two robot behavior conditions, the order of the trials, and the interaction between condition and order. For the first question, "How well did the robot's movements adhere to human social norms," the average response across all results was 4.27, with no significant effects (all $p > 0.1$). The third question, "How well could you anticipate the robot's movements," also showed no significant effects (all $p > 0.1$), with an average response of 4.79 across conditions.

For the question, "How well did the robot respect your personal space," the "social" behavior was rated significantly higher than the "non-social" behavior (5.59 versus 3.78, on a scale of 1–7; $F = 15.37$, $p = 0.0003$). There were no main or interaction effects of the trial order. The results from this question are shown in Figure 5.27.

The question, "How much did you have to get out of the robot's way," also

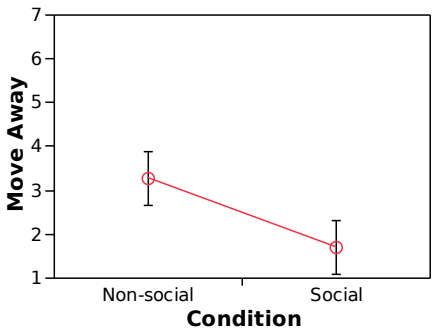


Figure 5.28: Results for "How much did you have to get out of the robot's way?" versus robot condition: $p = 0.0006$ (errorbars indicate ± 1 std err). People did not feel they had to move as far away when the robot was trying to be social.

showed a significant effect due to condition, with participants feeling they had to move further away from the "non-social" robot (3.26 versus 1.70; $F = 13.27$, $p = 0.0006$). This result is shown in Figure 5.28. No main or interaction effects from the trial order were found.

Additionally, we analyzed the social survey questions for effects of gender and robot experience. Gender effects were tested with one-way ANOVAs for each survey question, and no significant effects were found (all $p > 0.1$). Since robot experience was measured on a continuous scale, its effects were tested using linear regression for each social survey question. Two significant effects were found. Participants' ratings of how natural the robot's behavior was significantly decreased with greater experience ($F = 5.69$, $p = 0.02$); see Figure 5.29. In addition, participants with greater robot experience felt they had to move further away from the robot ($F = 7.96$, $p = 0.0067$); see Figure 5.30. Finally, participants' ratings of how well the robot respected human social norms decreased with robot experience, with marginal significance ($F = 3.04$, $p = 0.087$). No other significant effects of robot experience were found (all other $p > 0.1$).

Participant Comments

Each questionnaire provided several blank lines for comments immediately following the social scales. While we did not explicitly codify and analyze these comments, they may provide more insight into the robot's behaviors.

Comments on the "Non-social" behavior: Sixteen participants provided comments on the "non-social" behavior. Many of the comments reflect the close distance the robot often left when passing the person, such as:

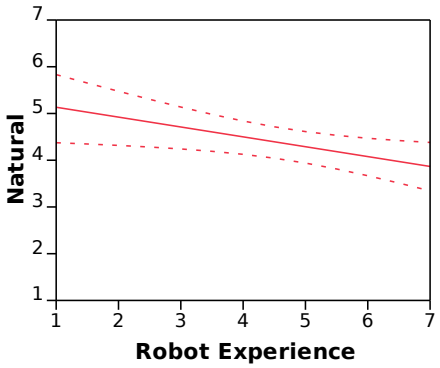


Figure 5.29: Best-fit line for "How natural was the robot's behavior?" versus experience with robots: $p = 0.02$. Dotted lines represent 95% confidence intervals. In general, people with more robot experience rated the robot as less natural.

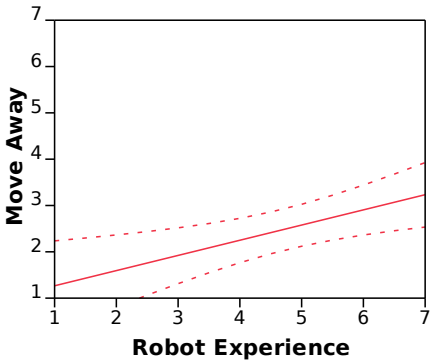


Figure 5.30: Best-fit line for "How much did you have to get out of the robot's way?" versus experience with robots: $p = 0.0067$. Dotted lines represent 95% confidence intervals. In general, people with more robot experience felt they had to move further away from the robot.

- "It felt this time like the robot came at me for a moment before turning and continuing down the hallway."
- "I didn't feel that the robot gave me enough space to walk on my side of the hallway."
- "Robot acted like I would expect a slightly hostile/proud human (male?) to act regarding personal space—coming close to making me move without actually running into me."
- "The robot came much closer to me than humans usually do."
- "It seemed obvious that the robot won't give me way."

Note that four of the comments on this behavior indicated that participants felt that the robot did, in fact, adhere to human social conventions for hallway encounters. For example, one participant wrote: "Passing in these trials felt very natural." Three out of the four participants who left similar comments saw the "non-social" behavior first.

Comments on the "Social" behavior: Thirteen participants left comments on the "social" behavior. Many of these comments indicated that participants felt the robot respected their personal space, but did not do so in a way that they expected, such as:

- "Sometimes it swerves away more than a person would, but that might be better since it's very large and heavy."
- "I felt that the robot obeyed social conventions by getting out of my way and passing me on the right. However, it seemed to turn away from me quite suddenly, which was very slightly jarring."
- "The robot seemed cold when moving away."
- "I think the robot gave me too *much* space." (emphasis in original)
- "It felt like the robot went very close to the wall...which a human wouldn't do as much (except maybe a very polite human...)"
- "It was really cool how it got out of *my* way." (emphasis in original)

5.2.4 Discussion

Hypothesis H1, which stated that participants will perceive a difference between the two robot behaviors and would rate the "social" robot more highly on social scales was shown to be generally correct. In particular, participants rated the robot's movements as better respecting their personal space and requiring them to move less out of the way when the robot was attempting to be socially correct. However, the ratings of human-likeness and other general social measures did not differ across the two behaviors. One possible explanation for this is that participants considered the robot as a whole for these questions; since the same robot was used, its overall human-likeness remained the same, despite a different movement pattern.

From participants' comments, we can infer that, while the "social" behavior did observe social conventions such as personal space, it did not do so in the same way that people do. Participants used terms such as "jarring" and "cold" to describe this behavior. We believe that this can be primarily explained by the fact that the robot used in this study is *non-holonomic*, and thus physically unable to move in ways that people do. In particular, for the robot to move to the side of the hallway, it must *turn* toward the wall (e.g., Figure 5.24(c)), rather than shifting sideways as a person might. Since this causes the robot to turn its face away from people, it is seen as less social. The "non-social" robot behavior, despite driving extremely close to participants, nevertheless does not turn its face as far away from them. We believe that this further demonstrates the ability of the COMPANION framework to produce different robot "personalities."

We were unable to prove hypotheses H2 and H3, regarding participants' affect and empowerment toward the robot. Participants felt equally dominant toward the robot in each behavior; in all cases people felt slightly above the mid-point of the dominance scale. That is, people felt slightly dominant toward the robot itself, but the robot's behavior had no influence on this feeling. Furthermore, participants' emotional states did not change after any encounters with the robot.

Finally, we noted several effects relating to prior robot experience. One effect was that participants with more experience tended to rate the robot as less natural and less in line with human social norms. We believe that this may be because people with more experience with robots are more likely to think of the robots as machines, while less experienced people may be more likely to anthropomorphize the robot. Additionally, participants with greater experience tended to feel they had to move further away from the robot. We suspect this is due to the fact that most existing robots tend not to avoid people, so experienced roboticists expected that they needed to move out of the robot's way; people less familiar with robots would have no such expectations.

5.3 Summary

In this chapter, we have presented the behavior of the COMPANION framework in multiple hallway scenarios, both in simulation and on a physical robot.

In simulation, we have described a wide variety of scenarios that help to understand the behavior of the COMPANION framework in hallway navigation. We have demonstrated that the use of mathematical cost functions can produce robot behavior that mimics human social norms. Additionally, we have shown how these behaviors can be modified by using alternate constraint weights, and how simple modifications to individual constraints can produce behaviors appropriate for other cultures.

In a user study, we verified that people do interpret the robot's behavior according to human social norms. However, though people felt the robot respected their personal space when it moved out of the way, they described the robot's method of doing so as "jarring." Participants also ascribed personality to both of the robot's behaviors; in particular, when the robot attempted to move according to social norms, it was either "cold" or "overly polite;" when it merely avoided running into people, it was "hostile" or "proud." Since the two robot behaviors differed only according to the constraints they used, the tendency of participants to ascribe different human-like personalities to the two behaviors supports our hypothesis that different constraint weights produce different types of socially-acceptable behaviors. Neither behavior was perceived as particularly *anti*-social.

Finally, we argue that the "jarring" behavior of the robot when it avoided participants' personal space arose from the non-holonomic nature of the robot used in the study. As we identified in Chapter 4, the ability to side-step obstacles is an important human behavior. A robot that is unable to perform such maneuvers cannot produce behaviors that are viewed as quite as human-like as a robot that is able to move sideways. This finding does not imply that non-holonomic robots cannot produce social behavior—the robot used in our study produced social behavior, if not optimally so—nor do we advocate the necessity of, for example, walking humanoid robots to produce truly human-like movement. Rather, we argue that holonomic actions are a significant part of human social movement, and a robot that is capable of such actions (such as the Companion robot, introduced in Chapter 7) should make use of them.

Chapter 6

Side-by-Side Escorting

The previous chapters have discussed the COMPANION framework in the context of a robot operating independently. This chapter focuses on an extension to the basic framework that allows a robot to navigate jointly with a person, particularly for the case of traveling side-by-side with a person, for the purpose of leading him somewhere.

6.1 Motivation

Beyond just navigating *around* people, we are interested in situations where a robot must travel *with* a person. We consider the following hypothetical scenarios as motivation for this focus:

Scenario 3 (A smart shopping cart)**.** *Alice checks out her SmartCart as she enters the grocery store. She enters her shopping list, and the cart immediately plans the optimal path through the store. The cart begins driving autonomously through the aisles, stopping whenever it encounters an item on Alice's list. The cart gracefully navigates around the other shoppers and continually watches that Alice is still following. When Alice remembers an item she forgot to place on her list, she turns around to return to the necessary aisle. The cart notices this, and switches into intelligent following mode. When space allows, the cart travels next to Alice in order to remain within her field of view and provide assurance that it is following her correctly, though it falls behind her as necessary to allow other shoppers to pass. Once Alice has retrieved her forgotten item, she pushes the cart forward, putting it back into leading mode. The cart replans its path for the remaining items, and continues on its way.*

Scenario 4 (A hospital and nursing home assistant)**.** *Bill has recently entered the assisted living retirement home, and he has made little effort to interact with the other residents. He has trouble finding his way around but is too embarrassed to ask for help. Noticing his withdrawal from other people, the home's staff sends the robotic assistant to his room to ask him if he would like to visit with other residents. Bill agrees, and the robot leads him to the common social room, where he can interact with others. Along the way, the robot travels by Bill's side and chats with him about other activities available in the retirement home.*

Current robotic systems have been used with people in the context of accompanying residents in nursing homes (Montemerlo et al., 2002) or guiding tours in museums (Thrun et al., 1999; Nourbahksh et al., 2003). Unlike our hypothetical scenarios, existing technology requires people to follow behind the robot at all times, as in the following:

- In an assisted living facility, a robot can travel to residents' rooms and guide them to appointments, or even just accompany them while they walk for exercise. Residents must follow behind the robot (Montemerlo et al., 2002). However, our own observations indicate that people in such an environment tend to walk side-by-side (see Section 3.3).

- In various museums, robots can lead groups of people (Thrun et al., 1999; Nourbahksh et al., 2003). Such systems assume that a group will follow behind the robot. Furthermore, these robots cannot perform personalized, one-on-one tours.

These systems, and others like them, are currently the status quo for human–robot interaction. In each case, the robot plans its own path of travel without regard for how the person is to travel; the person is always assumed to follow behind the robot. However, when two people walk together, they enter a *collaborative process*, attempting to minimize not only their own but also each other's required effort (Klein et al., 2005). To model this behavior, we use the concept of *joint planning*, as described in the next section.

6.2 General approach

We argue that, if a robot plans a path that accounts for *joint* behavior, then its use of social conventions will allow the person to understand the robot and walk along with it. That is, the social conventions of walking together will provide *common ground* between the robot and the person. In that way, walking next to the robot

will require only a person's prior knowledge of the conventions of walking with another person, rather than additional knowledge of the robot's conventions.

From our observations of people walking together (Section 3.3), we know that people use various physical behaviors as cues for joint travel—such as moving closer or further away to indicate a turn, or speeding up to pass through a choke-point first. We believe that, by planning for joint behaviors, the robot can be made to display such appropriate social cues. We make the assumption that the person is a willing participant in the social interaction; that is:

- The person agrees to participate in the joint behavior, and is thus not adversarial; and
- The person will also be attempting to move in socially appropriate ways.

By assuming that the person will move with the robot in socially appropriate ways, the robot can approximate the cost of the person's behaviors as well as its own behavior costs. The robot can thus plan joint behaviors by attempting to minimize both its own effort and the approximate effort of the person. The resulting path for the robot may cost more than the optimal path planned for the robot individually; the differences between the two paths result from accommodating the person's social desires, and represent the social cues that the robot may give the person. Even though the robot's approximation of the person's costs is unlikely to be perfect, as long as the planner is able to run repeatedly in real-time, the robot will continually react to the person's movements.

As presented in the previous chapters, the basic COMPANION framework allows a robot to plan paths for itself only, traveling around any people. For the robot to plan to travel *with* a particular person, all aspects of the path planning must account for the person as well as the robot. We thus extend the COMPANION framework for joint planning by introducing the concepts of *joint goals*, *joint actions*, and *joint constraints*.

6.2.1 Joint goals

Definition 6.1. *A* joint goal *is the desired final world state, including the desired goals of both the robot and a particular person.*

The path planner must have a desired goal state in order to compute a path. The simplest form of a joint goal state would be for the robot to reach its goal location, with the person within some pre-defined region around the robot's goal. Since we are interested in side-by-side travel, we further restrict the goal state such that the person must be by the robot's side.

The goal may be task-specific in other ways, as well. For example, consider a museum tour-guide robot that can best describe various exhibits if the person is in a particular location with respect to each exhibit. If the goal state is defined to have the person in a particular position, then the planner can compute the best path for the robot that will encourage the person to end in that location.

6.2.2 Joint actions

Definition 6.2. *A* joint action *between a robot and a person is composed of an action to be taken by the robot as well as an action to be taken by the person, both for the same length of time.*

For robot path planning, we defined 14 actions (see Section 4.4.1): straight, forward left turn, and forward right turn, each at three speeds; stop; and the holonomic actions of sideways left, forward sideways left, sideways right, and forward sideways right. To perform joint planning, the robot must also plan for where the person might go. Currently, we assume only three possible person actions: straight, forward left turn, and forward right turn. Ideally, the robot will want to match the person's speed. However, we currently make the simplifying assumption that the person travels only at the robot's default speed (0.5 m/s).[1] A joint action is composed of one robot action and one person action, so the planner has a total of 42 available actions. While in theory the person could perform at least as many actions as the robot, the addition of more actions quickly becomes intractable. If the planner is, as before, assumed to run repeatedly in real-time, the use of only a few possible human actions should still produce social paths for the robot.

6.2.3 Joint constraints

Definition 6.3. *A* joint constraint *is a function representing the cost of the robot and a person transitioning from one state to another via a joint action.*

The robot should plan paths that attempt to minimize the *joint cost*, that is, the path costs for both the robot and the person. To do so, the robot must consider what costs the person might incur in addition to its own costs. One approach to this is the concept of "reflective navigation" (Kluge, 2004); this would mean that the robot considers the costs to the person to arise from the same constraints and weights as its own costs. That is, the robot could consider each person action as if it were its own, and determine the cost from its own constraints.

[1] The planner could adjust the robot's default speed to the tracked speed of the person, for example.

However, rather than use the same set of constraints for the person and robot, we define separate constraints that relate to the person of interest, so that the new constraints could be weighted differently from the robot costs. Since we are considering only straight and turning actions for the person, we need to define two new person-related constraints:

- Minimize the person's distance traveled; and
- Reduce the number of turns the person makes (i.e., inertia).

Additionally, the robot must consider hard obstacle avoidance for the person, so that it does not plan paths requiring the person to walk through walls or through other people. Currently, we do not consider other social conventions for the person, such as avoiding other's personal space or tending to the right. This is done to simplify computation, though future work may indicate the necessity of such constraints.

Finally, we make one additional change to the "tend to the right" constraint given in Section 4.2. Since that constraint models *passing* people, it must be ignored for the person with whom the robot is traveling. This is done as a special case within the constraint; the person traveling with the robot produces no cost region associated with the "tend to the right" constraint. Note that if this change were not made, the robot would demonstrate a strong preference for remaining on the person's left.

6.3 Constraints for side-by-side escorting

While the extended COMPANION framework for joint path planning allows the robot to plan a joint path to a goal, it does not place any constraints on the relative positions of the robot and the person. In order to encourage the robot to travel next to the person, as would be the case for socially escorting the person, we must define additional task-specific constraints. In particular, we define two constraints for side-by-side escorting: walking with a person, and remaining next to the person. That is, the two constraints define the preferred distance and angle, respectively, between the robot and the person.

6.3.1 Walk with a person

As discussed in Chapter 4, the robot and all people in the environment each have a personal space (or "robot space") zone around them, which will tend to keep the robot a fairly large distance away from all people. For the robot to walk *with* a person, then, we can define a constraint that acts as a spring between the robot and

the person, which will act in concert with the personal spaces to maintain a socially appropriate separation.

To walk with a person, then, the robot should try to keep some preferred distance d_p between itself and the person. We do this by defining a "walk with a person" cost that increases linearly with the distance between the robot and the person, for distances greater than d_p. For $d_p = 1$ m, this cost is shown in Figure 6.1. This cost is 0 for distances under d_p because the "personal space" and "robot 'personal' space" constraints already provide a repelling force between the robot and person. We currently keep the preferred distance a constant value, but note that Sviestins et al. (2007) hypothesizes that the preferred distance should actually *decrease* with faster speeds.

Because the "walk with" constraint is intentionally defined as a competing force against the "personal space" and "robot 'personal' space" constraints, the relative weights of all three constraints must be balanced to achieve the desired robot behavior. In particular, we want that the three constraints combine to form a trough of low-cost actions when the robot and the person are the preferred distance apart. We start with the weights given to the personal space constraints as discussed in Chapter 5 and given in Table 5.1. That is, we let the "personal space" $w_{ps} = 2$ and the "robot 'personal' space" $w_{rps} = 3$. We find that a distinct low-cost trough occurs when the "walk with" constraint weight w_{ww} is equal to the sum of the personal space constraint weights; that is:

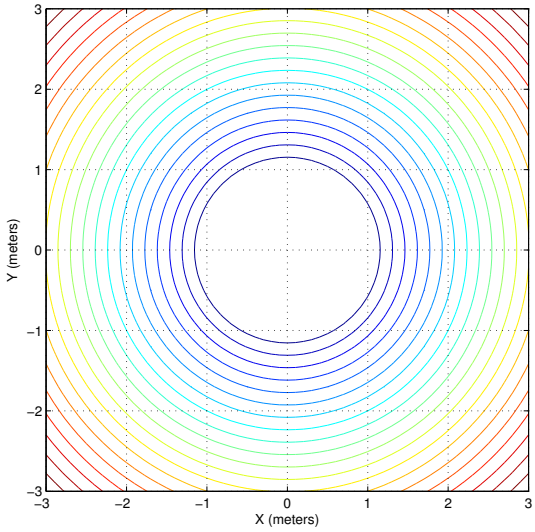
$$w_{ww} = w_{ps} + w_{rps} \tag{6.1}$$

If the robot and person are traveling side-by-side (with the same heading), this cost can be seen in two dimensions in Figure 6.2. Expanding this visualization to three dimensions, so that the person can be at any position relative to the robot, yields the cost regions shown in Figure 6.3.

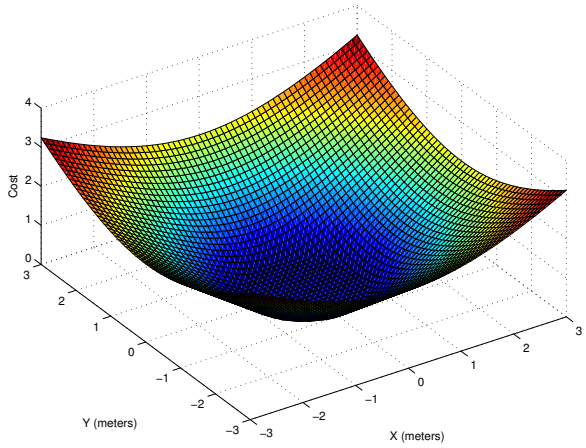
As with many other constraints given in Chapter 4, this cost is scaled by the length of time for each given (joint) action.

6.3.2 Side-by-side

From Figure 6.3, we see that the largest region of low cost results from the person remaining in the U-shaped region to the side or behind the robot. However, for side-by-side escorting, we want the robot to prefer to keep the person to its side, rather than behind. We do this with the addition of the "side-by-side" constraint, which adds a cost proportional to the relative angle between the robot and person. In particular, we define two angles: α_{r-p}, the angle from the front of the robot to the person's position, and α_{p-r}, the angle from the front of the person to the robot's



(a) Contour map



(b) Surface plot

Figure 6.1: Different views of the "walk with a person" constraint, shown as the cost of the relative position between the person and the robot, with the robot centered at (0,0).

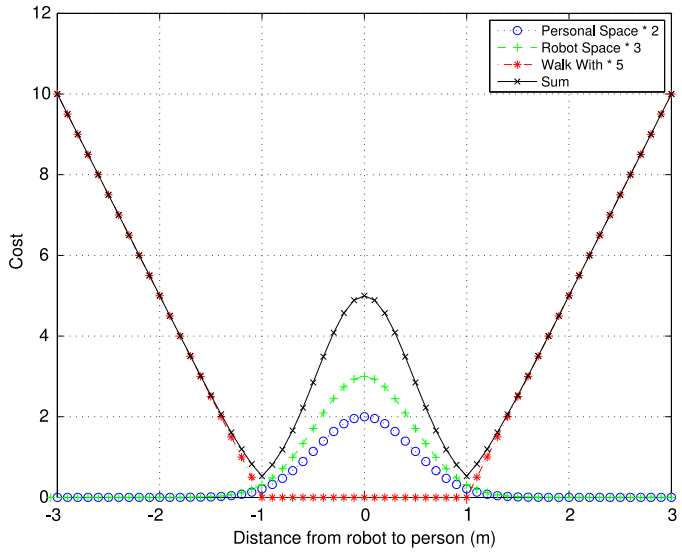


Figure 6.2: 2D view of the weighted constraints of "personal space" ($w = 2$), "robot 'personal' space" ($w = 3$), and "walk with a person" ($w = 5$), as well as their sum. This is shown for the robot and person directly side-by-side, with the same heading, and each traveling at 0.5 m/s.

position. In the desired side-by-side positioning, these angles will be $\alpha_{r-p} = 0$ and $\alpha_{p-r} = \pi/2$, or vice versa. Both angles are necessary for the cases where the robot and person are not facing the same direction, as shown in Figure 6.4. The cost for each angle is the absolute difference from the desired angle:

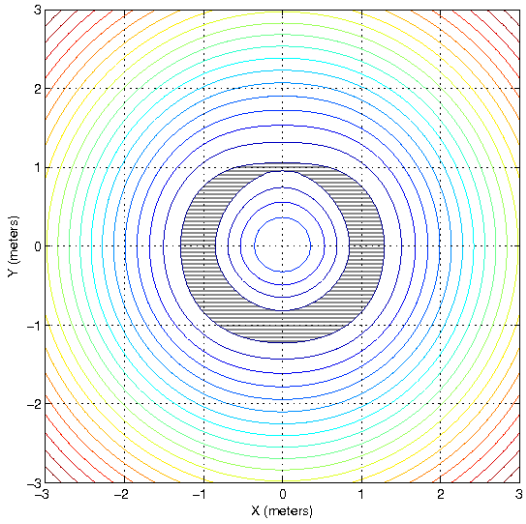
$$c_\alpha = \begin{cases} |\alpha + \pi/2| & \text{if } -\pi < \alpha < 0 \\ |\alpha - \pi/2| & \text{if } 0 < \alpha < \pi \end{cases} \tag{6.2}$$

The "if" clause simply allows the person to be to the robot's right or left, with equal cost.

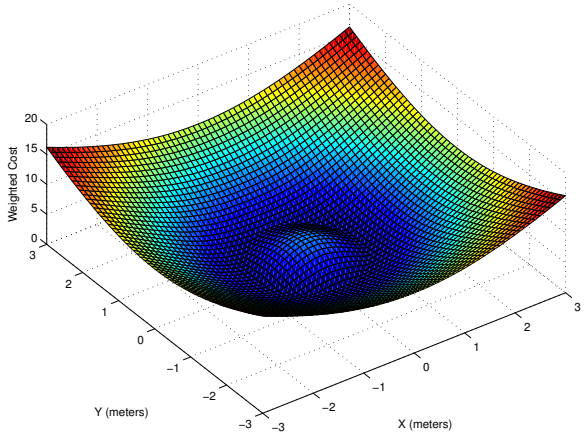
The cost c_{ss} is the sum of these costs times the action time ($a.t$):

$$c_{ss} = (c_{\alpha_{r-p}} + c_{\alpha_{p-r}}) \cdot a.t \tag{6.3}$$

This cost could trivially be changed to represent a different preferred angle between the robot and person. For example, if the preferred position has the robot in front of the person, the preferred angles are $\alpha_{r-p} = \pi$ and $\alpha_{p-r} = 0$.



(a) Contour map



(b) Surface plot

Figure 6.3: The result of adding the weighted constraints of "personal space" ($w = 2$), "robot 'personal' space" ($w = 3$) and "walk with a person" ($w = 5$), shown as the cost of the relative position between the person and the robot. The robot is centered at (0,0) and both the person and robot are heading at 0.5 m/s along the positive Y-axis ("up"). The lowest cost region is largest when the person is positioned to either side of, or behind, the robot (shaded).

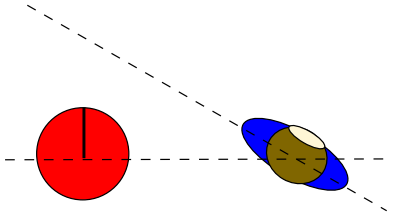


Figure 6.4: A robot (left) and a person (right). Because they are not facing the same direction, the person is next to the robot (with respect to the robot), but the robot is not next to the person (with respect to the person).

6.4 Heuristics

By definition, the escorting constraints incur cost with each step that the robot and person take together. As discussed in Section 4.4.2, the A* search relies on predictive heuristics to find paths through cost regions. Individually, none of the "personal space," "robot 'personal' space," or "walk with" constraints afford useful heuristics. Together, however, the three constraints have a minimum instantaneous cost. Since the "side-by-side" constraint adds additional cost when the robot and person are not directly next to each other, we can use the 2D side-by-side cost function shown in Figure 6.2, from which we can see that this minimum cost is approximately 0.5.

Since the cost is dependent on travel time, we must further approximate the remaining time needed for the robot and person to reach the goal. Since the person is assumed to keep a constant speed, we can estimate the time-to-goal t_g based on the Euclidean distance remaining divided by the person's speed. Thus, the predictive heuristic should be approximately $0.5t_g$.

In practice, we over-estimate the heuristic by using a cost of $0.7t_g$. Though 0.7 is greater than the true minimum value, it affords a faster search time, and we know that the sub-optimality of the resulting paths are bounded (e.g., Chakrabarti et al., 1987). We found that this over-estimation is necessary in part due to the use of a variable grid, which resulted in planner not always being able to align the robot and person at the optimal distance apart (due to the coarseness of the grid). Thus, the true minimum cost actions are typically not available. The variable grid is necessary due to the extremely large state space.

Table 6.1: Constraints and their weights used in the objective function for side-by-side escorting. The first set of constraints are described in Chapter 4; the remaining constraints are specific to joint planning. The hard constraints of avoiding obstacles and people are also used.

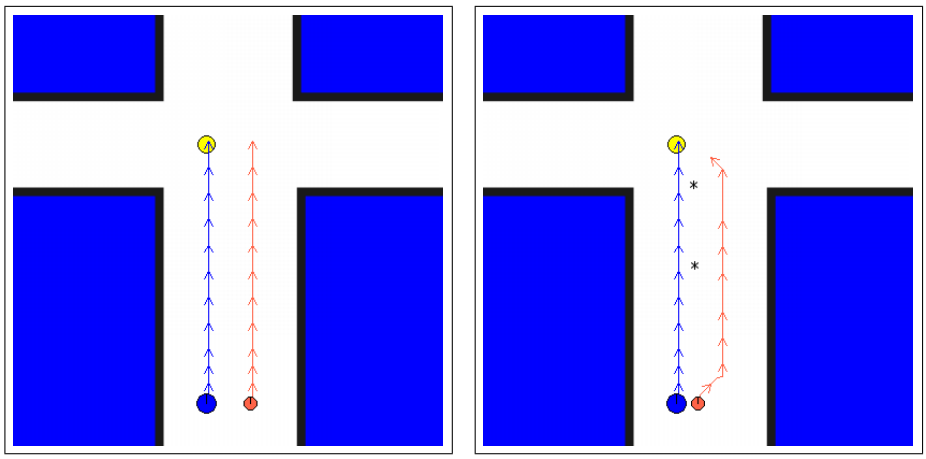
Constraint Name	**Weight (w_c)**
Minimize distance	1
Obstacle buffer	1
Personal space	2
Robot space	3
Pass on right	2
Default velocity	2
Face travel	2
Inertia	2
Minimize person's distance	1
Person's inertia	1
Walk with a person	5
Stay side-by-side	1

6.5 Escorting in simulation

To understand the behavior of the joint path-planner, we present several scenarios here. We use the constraint weighting given in Table 6.1. These are the same weights used in Section 5.1.1, for robot-only path planning, but with the addition of the joint constraints given above. Furthermore, all paths shown here were produced using a variable grid (see Section 4.4.2), both to improve search time and because more complex searches become intractable given the extremely large state space. The hardware used to produce these results was also that used in Section 5.1.1.

We first address the simple case of a goal straight ahead, down a single hallway. Two examples are shown in Figure 6.5. In Figure 6.5(a), the robot and person begin at the optimal distance of 1 m apart, so the resulting plan simply requires both the robot and person to travel straight at a constant speed. In Figure 6.5(b), however, the robot and person begin only 0.5 m apart. Since the robot's goal is straight ahead of it, the robot assumes the person will take the initiative to move further away. However, since the person moving on a diagonal takes longer than the robot moving straight, two actions require the robot to drive more slowly, to allow the person to catch up.

Figure 6.6 shows examples of paths that require left- or right-turns, with the robot on the inside or outside of the turn, respectively. On the inside turn (to the



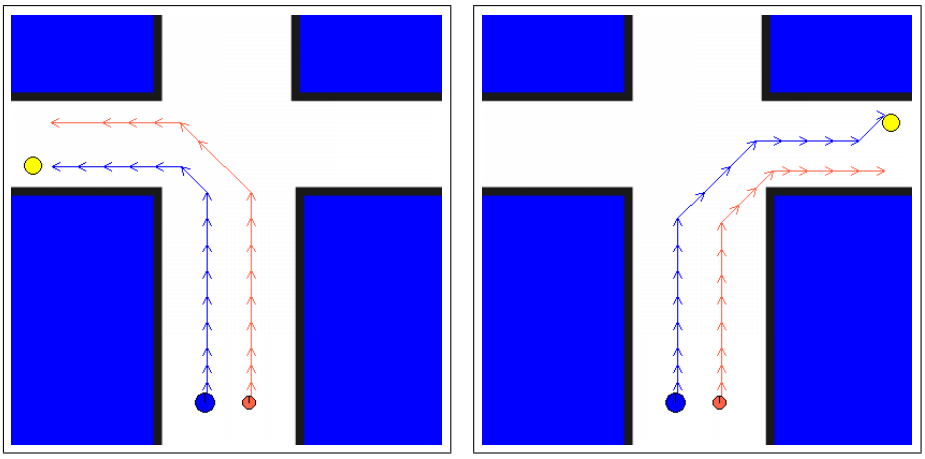
(a) Robot and person start 1 m apart (b) Robot and person start 0.5 m apart

Figure 6.5: Joint plans for a robot and a person with the goal straight ahead. In (a), the robot and person start at the best distance apart, so both simply travel straight. In (b), the robot and person start too close to each other. Since the robot's goal is straight ahead, the best plan is for the person to move slightly further away. The two points marked with asterisks indicate segments where the robot drives more slowly, to allow the person to catch up.

left, Figure 6.6(a)), the robot plans to slow down, allowing the person to travel the longer distance around. On the outside turn (to the right, Figure 6.6(b)), the robot plans to travel faster around the turn, so that it remains next to the person at all times. Each of these plans required less than 0.4 seconds to generate.

This framework can also handle cases where the person does not follow the robot's planned joint path. Figure 6.7 shows a scenario where the person has lagged behind the robot and drifted toward the right side of the hallway. Since, as before, the robot is assumed to be able to replan rapidly, the plan it produces at this step is for the robot to move sideways, toward the person's location. Even though this increases the robot's path length, it reduces the cost of the "walk with" constraint. Because the person is far behind the robot, this path is lower cost than if the robot were to wait for the person to move back into position; this is in contrast to Figure 6.5(b) when the robot and person begin too close to each other. This plan took approximately 30 seconds to generate.

Finally, consider the case where the goal requires the person and robot to travel through a chokepoint, that is, through a narrow section of the hallway, such as a doorway. Such a scenario is shown in Figure 6.8, in which the chokepoint appears



(a) Goal to the left, with the robot on the inside of the turn.

(b) Goal to the right, with the robot on the outside of the turn.

Figure 6.6: Joint plans for the robot and a person that require turning left (a) or right (b). The robot plans to slow down on the inside turn and speed up around the outside turn, so that it remains side-by-side and at the preferred distance from the person. The person is assumed to maintain a constant speed.

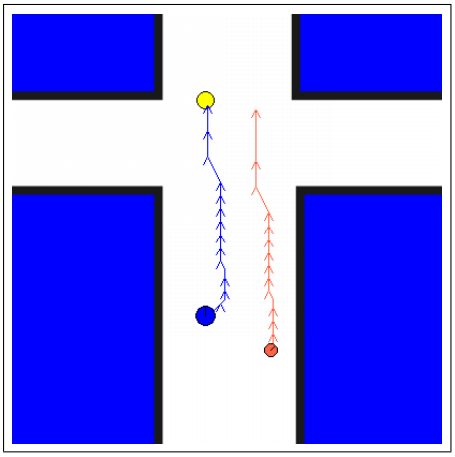


Figure 6.7: Joint plans for the robot and a person, where the person starts at a non-optimal location. The robot begins by moving sideways, closer to the person, even though its shortest path would be to drive straight to the goal.

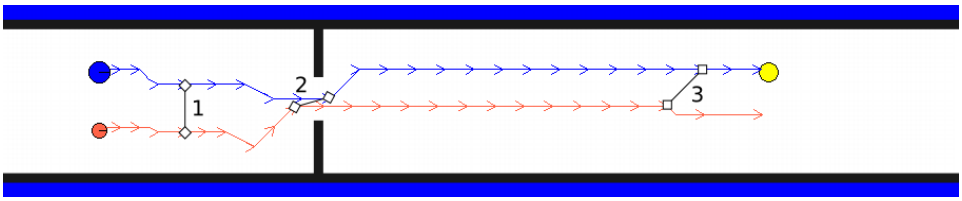


Figure 6.8: A joint plan for the robot and a person that requires that both pass through a narrow chokepoint (e.g., a doorway) in the hallway. In this plan, the robot speeds up (1) to pass the person and drive through the chokepoint first (2). The robot remains a short distance in front of the person for much of the remainder of the walk, slowing down to allow the person to catch up near the goal (3). Note that the hallway is approximately 20 m long.

approximately 4 m ahead of the robot and person, while the goal is about 15 m away. In this environment, the robot and person cannot pass through the chokepoint side-by-side. Instead, the robot plans to increase its speed so that it may pass through the chokepoint ahead of the person. Eventually, the robot reduces its speed so that the person is able to catch up. This plan required 99 seconds to compute, generating nearly 4 million states.

6.6 Summary

In this chapter, we have presented an extension to the COMPANION framework to allow for joint human–robot paths, and we have presented the specific implementation of side-by-side escorting. The extension relies on the notion of social conventions as common ground between the robot and person, so that if the robot presents socially correct movement cues, the person will react appropriately. We defined the concepts of joint goals, joint actions, and joint constraints, so that the robot can plan paths that attempt to minimize both its own and also the person's expected path costs.

For the specific task of side-by-side escorting, we introduced two additional constraints: a cost corresponding to the distance between the robot and the person, and a cost corresponding to the relative angle between them. These costs, combined with joint path planning, allow for plans in which the robot speeds up or slows down appropriately around corners or through chokepoints.

In simulation, we have demonstrated the ability of the framework to find joint plans for a robot and a person traveling together. The planned paths model social behaviors such as having the robot slow down when on the inside of a turn and

speed up on the outside. In addition, the planner is able to handle situations when the side-by-side constraint cannot be maintained, such when the robot and person must pass through a chokepoint that is only wide enough for one at a time.

Unfortunately, the joint planning currently cannot execute in real-time: while some simple plans are produced quickly, planning a path through a narrow chokepoint requires nearly two minutes. This is due to the enormous state space—since the planner must consider a person's position and orientation along with the robot's position, orientation, and velocity, the state space has 9 dimensions. Furthermore, each state may have as many as 42 unique successor states. Since both the size of the state space and the number of successors are factors in the time complexity of the planning algorithm, real-time planning is difficult to achieve. Furthermore, since A* stores all generated nodes in memory, planning in such a high-dimensional state space can easily overwhelm a computer's resources (Russell and Norvig, 2003). The solution to faster planning may require either a different type of basic path planner or simply faster computer processors; such approaches will be addressed in Chapter 8.

Despite the current planning speed limitations, we believe that the joint planning extension to COMPANION is an extremely powerful framework. It allows for the robot to consider an interaction partner as a social entity. Since social conventions are encoded directly into the path planner, we believe that the robot will automatically present the necessary cues for joint movements.

Chapter 7

Companion Robot Design

In addition to the theoretical COMPANION framework, implementation, and results presented in the previous chapters, the final contribution of this thesis is a new platform for social robotics research. This chapter details the design process and final robot, which we call Companion.

The Companion robot has two main components: a holonomic mobile base (Section 7.1) and a fiberglass outer shell (Section 7.2). While both components were designed concurrently, we will discuss each in turn.

The design of the Companion robot was a highly collaborative process. The author's role in this process was that of team leader, coordinating team members and defining the desired capabilities of the robot. The author drove the design effort and was the primary decider in design choices.

7.1 Holonomic base design

The base of the robot consists of those components necessary for motion: wheels, motors, batteries, electronics, and sensors. In this section, we present our rationale for, and final design of, a new robot base.

7.1.1 Rationale

The overall goal of this research is to design methods for robots to navigate around people in social ways. Our approach is to model robotic behavior on human social norms, so that people may apply their knowledge of these conventions to their interactions with the robots. However, humans are able to maneuver themselves in far more complex ways than most mobile robots. In particular, the vast majority of commercially available robots used in human–robot interaction research are *non-*

holonomic. In this context, a non-holonomic robot is capable of moving forwards and driving along arcs, but is not capable of instantaneously moving sideways.

We believe that this capability to side-step is an important aspect of human social navigation. In designing the COMPANION framework (see Chapter 4), we represented this capability as a preference for facing the direction of travel, creating a trade-off between sideways movement versus turning. When we identified this constraint, we began the design of a holonomic robot, which would be capable of sideways movements. The new robot was not yet completed when we ran the user studies presented in Chapter 5, and the results we obtained using Grace—a non-holonomic robot—further demonstrated to us the importance of holonomic movements for social robots. In particular, while participants felt Grace avoided their personal space, they also felt that her manner of doing so was awkward and even jarring. We believe that this response was due to Grace's turning away to drive around people; a robot that could shift sideways without turning may be seen as much more social.

7.1.2 Design Process

The Companion robot began as a small base designed by Botrics, LLC, as a 3-wheeled holonomic version of their Obot d100 robot.[1] However, as the body design became more ambitious (see Section 7.2), we determined that the existing base would not be sufficient. Since we still desired a holonomic robot, and were unable to find a suitable platform available commercially, we redesigned the Obot robot base to support the following design criteria:

- Max 30 kg robot weight, including base, batteries, and shell;
- 1.5-2.0 m/s maximum velocity (e.g., fast walking speed);
- Support continuous acceleration (e.g., repeated stops and starts);
- 3-6 hours of driving time.

We defined the given speed and driving time as a result of the author's intent to use Companion in various human–robot interaction studies, including the side-by-side escorting discussed in Chapter 6. To do so, the robot must be able to keep pace with a typical human walker, preferably with the ability to speed ahead if necessary (such as around corners). The extended battery life is extremely beneficial to such user studies.

[1] The Botrics Obot robot: http://botrics.com/products/obot/

7.1.3 Final design

The final design of the base is shown in Figure 7.1. The base is comprised of three 1/4"-thick aluminum plates, each 45 cm in diameter. The upper-most plate provides a mounting surface for a computer, sensors, and various electronics. This plate is supported above the center plate using aluminum rods topped with Sorbothane® rubber shock-mounts (not shown). The center plate provides space for four lithium polymer batteries and chargers, as shown in Figure 7.2. Finally, the bottom-most plate holds the three motors and wheels, as shown in Figure 7.3.

The robot base is a modified Killough platform (Pin and Killough, 1994). This design utilizes omniwheels, as shown in Figure 7.4, rather than active steering (like a car) or differential drive (like many commercial research robots). Each omniwheel is driven around its major axis of rotation, but has rollers that allow for sideways slippage. The wheels are arranged around the base at 120° intervals, as shown in Figure 7.5.

With this wheel setup, the robot can instantaneously achieve any arbitrary translational and rotational velocity, within the physical limits of the motors. In particular, to achieve a given translational velocity $|V|$ in the θ direction (relative to the front of the robot) and rotational velocity $\dot{\psi}$, the rotational velocity of each wheel must be set to:

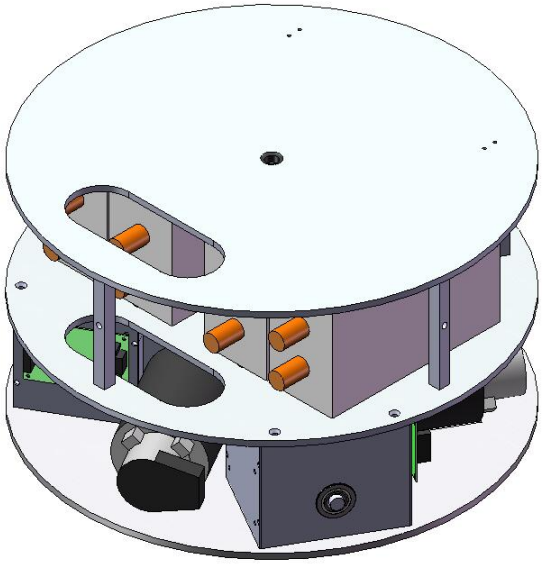
$$w_1 = \frac{|V|}{2r}\left(-\sin\theta + \sqrt{3}\cos\theta\right) + \frac{\dot{\psi}d}{r} \tag{7.1}$$

$$w_2 = \frac{|V|}{r}\sin\theta + \frac{\dot{\psi}d}{r} \tag{7.2}$$

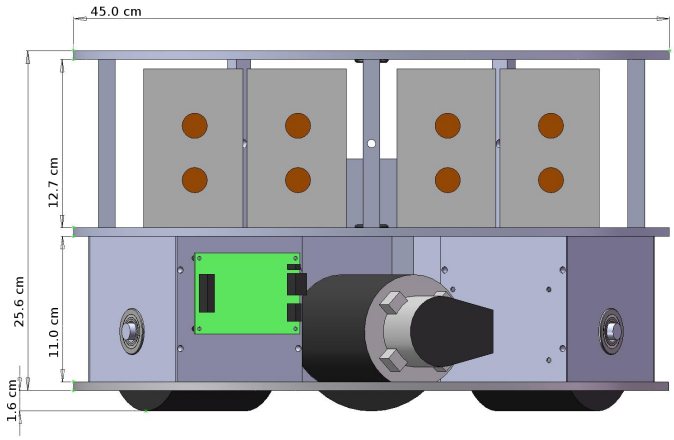
$$w_3 = \frac{|V|}{2r}\left(-\sin\theta - \sqrt{3}\cos\theta\right) + \frac{\dot{\psi}d}{r} \tag{7.3}$$

where r is the radius of one wheel, d is the distance from the center of the robot to the center of each wheel, and w_i is the rotational velocity in radians per second of wheel i. For the Companion robot, $r = 0.06$ m and $d = 0.155$ m.

The electronics for the base are composed of five custom-built boards, designed by David Bromberg and Brian Kirby. The boards, as mounted on the robot, are shown in Figure 7.6. The main board controls communications and power distribution between the on-board computer and the rest of the robot hardware. This board allows for AC or DC (battery) power and automatically switches to battery charging when plugged into AC. In addition, the main board allows for simultaneous communication to the three motor controllers. Each motor is controlled by a



(a) Isometric view



(b) Side view, with dimensions

Figure 7.1: Two views of the Companion robot base rendered in SolidWorks. The top plate provides a mounting surface for the robot computer, electronics, and housing frame. The upper level holds the batteries and chargers, while the lower level contains the motors and wheels; through-holes (visible in (a)) allow cables to be run between the levels.

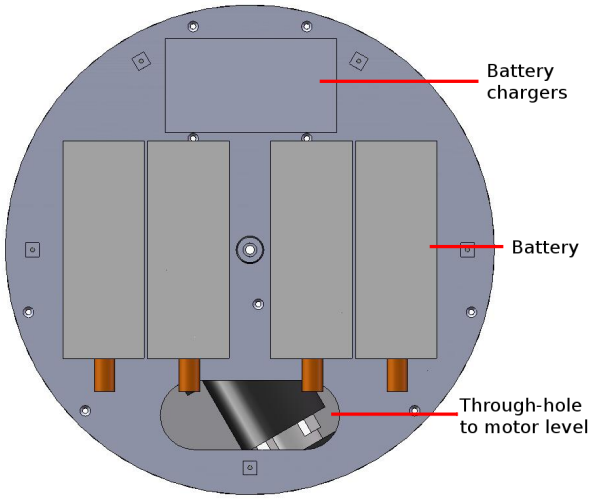


Figure 7.2: Top-down view of the robot base, with the top plate removed. This level holds the lithium polymer batteries and smart chargers. A through-hole allows for cable connections between the levels.

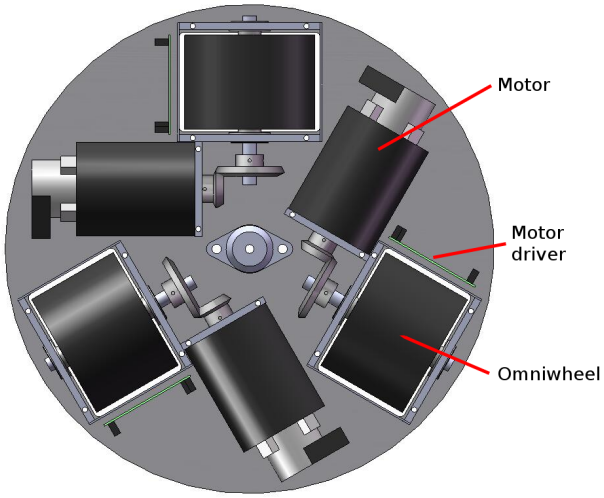


Figure 7.3: Top-down view of the robot base, with the top two plates removed. This level supports the three motors and three omniwheels, arranged symmetrically around the base.

Figure 7.4: An omniwheel produced by the Kornylak Corporation. The wheel as shown is composed of two separate omniwheels, each with three rollers. Combined, the wheel can provide sideways slippage over a full 360° rotation.

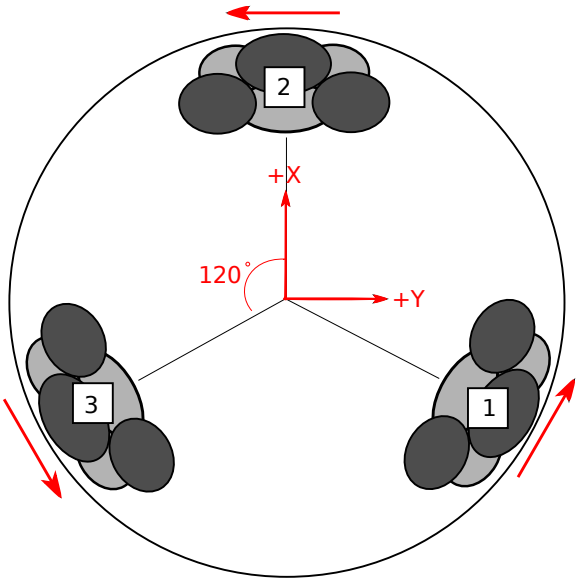


Figure 7.5: The layout of the three-wheel omniwheel drive. The wheels are at a 120° offset from each other. Each wheel is driven along the direction of the red arrows, and can freely slip in the direction perpendicular to its corresponding arrow. Wheel 2 corresponds to the front of the robot.

dedicated digital motor driver, made by ADVANCED Motion Controls.[2] (see the parts list in Table 7.1). To reduce the footprint, the drivers are manufactured with banks of pin connectors, and so require a separate interface board in order to have standard connectors (such as a serial port). While AMC makes such a board, we required a smaller footprint, and so designed our own interface boards. The robot uses three such boards, one per motor. Finally, a small daughter board provides three power switches (robot power, computer power, and computer on/off), as well as a small LCD screen and several LEDs that can be used for status outputs. The status board is mounted on a pole above the base for easier access.

High-level control of the robot is achieved with a standard mini-ITX desktop computer. In particular, Companion's computer, made by Portwell Technology,[3] runs on a quad-core Pentium processor at 2.4 GHz, with 4 GB of RAM. In addition, the computer uses a DC-input power supply, so that it can be run off batteries. The computer connects to the robot electronics via USB. Finally, the computer also runs the robot's primary sensors, two Hokuyo URG scanning laser rangefinders, mounted at the front and rear of the robot. The URG lasers each have a 240° field of view with 0.36° resolution, approximately 6 m range, and scanning rate of 10 Hz. The URG lasers are situated so that the robot can produce a 360° sensor sweep.

A parts list can be found in Table 7.1. The assembled robot base is shown in Figure 7.7.

7.2 Housing design

Although the base alone is a functional autonomous robot, we wanted to design a physical "body" for the robot, for both aesthetic and functional reasons—in particular, since the purpose of the robot is interaction with moving people, the robot needs to be easily visible. Early in the process, we defined the following design criteria as desirable for a social robot:

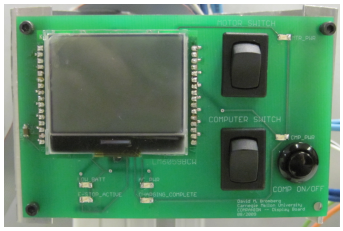
- The body of the robot should have a more organic shape than is typical of most research robots, which are often likened to cylindrical trash cans.
- The robot should be tall enough to be noticeable when it is amongst standing people, but it should not feel intimidating to its interaction partners.
- The robot's body should not suggest skills beyond its capabilities, such as hands on a robot that cannot grasp.

[2] ADVANCED Motion Controls: http://www.a-m-c.com AMC provided a discount on the motor drivers under their University Outreach program.

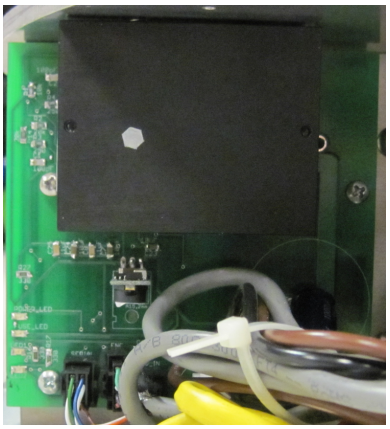
[3] American Portwell Technology: http://www.portwell.com/index.htm

Table 7.1: Major parts of the holonomic base and their costs. Total cost for the base was approximately $15,000.

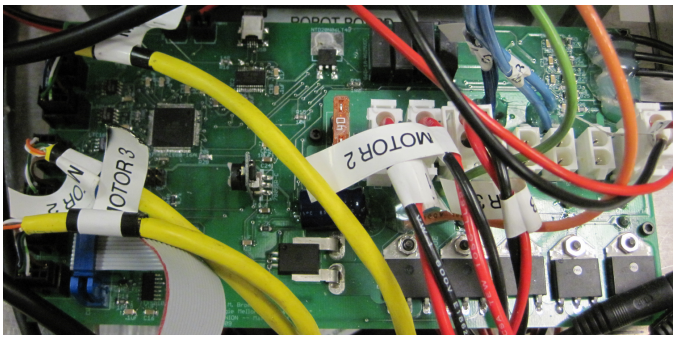
Part	**Retailer**	**Cost**
Omnidirectional wheels (6) *120 mm RW28*	Kornylak Corporation http://www.omniwheel.com	$402.50
Motors (3) *BR344C70102100*	Click Automation http://www.clickautomation.com	$1324.12
Motor controllers (3) *DZRALTE-020L080A*	ADVANCED Motion Controls http://www.a-m-c.com	$767.68
Gear sets (3) *M16P-2*	WM BERG http://www.wmberg.com	$426.14
Custom cut aluminum plates (3)	Atomatic Manufacturing Pittsburgh, PA	$730.00
Mini-ITX computer *WADE-8656* and *WADE-2231*	Portwell Technology http://www.portwell.com	$1100.00
25.9V Polymer Li-Ion Batteries (4) *HPL-BX25.9V10AhWR-FG*	AA Portable Power Corp. http://www.batteryspace.com	$1600.00
Smart Battery Chargers (4) *CH-LI2225-7*	AA Portable Power Corp. http://www.batteryspace.com	$120.00
Sorbothane® Shock Mounts (5) *V10Z59-MF2515050*	Advanced Antivibration Components http://www.vibrationmounts.com	$30.00
Hokuyo URG lasers (2) *R283-HOKUYO-LASER1*	Acroname http://www.acroname.com	$5000.00
Custom circuit boards *Designed by David Bromberg*	Advanced Circuits http://www.4pcb.com	$148.44
Misc. circuit parts	Digikey: http://www.digikey.com Newark: http://www.newark.com	$400.00 $60.00
Machining costs	In-house	$2500.00



(a) Power switches and small LCD for status outputs.



(b) Motor controller interface board. The large black component is the AMC driver.



(c) Main board for power distribution and serial communications.

Figure 7.6: Custom designed circuit boards for the Companion robot.

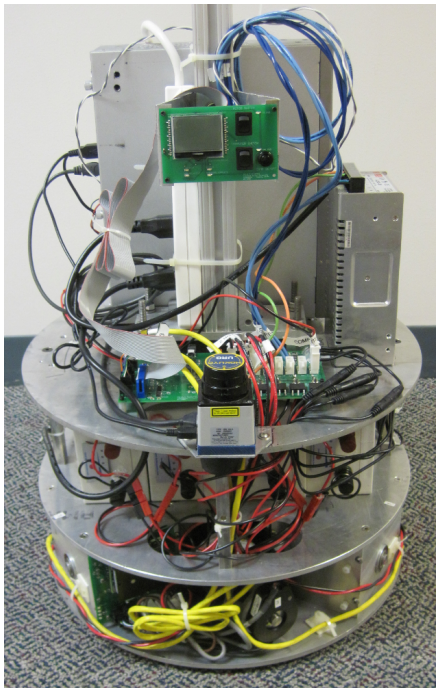
- The robot should have face, to be a locus of interaction.
- Finally, the robot's body should have a distinct front, back, and sides, to provide orientation for human companions.

These criteria were drawn primarily from the team's own experiences with human–robot interaction research.

7.2.1 Early design sketches

The preliminary design work was done by Scott Smith, an undergraduate design student at Carnegie Mellon University. While he initially sketched a wide variety of

(a) Front view. Visible on the top plate are the front-mounted laser and the computer. To the left of the computer is the power supply.



(b) Rear view. The rear-facing laser is centered in the image; behind that is the main robot board. The power switch / LCD board is mounted on the 80/20 pole.

Figure 7.7: Front and back views of the completed Companion robot base. On the center of the top plate is a large 80/20 pole, primarily used for mounting the housing.

body designs, we selected the basic form shown in Figure 7.8. These sketches also show an early idea of a simple face display, using fixed LEDs. During the time that these sketches were produced, we also performed an informal survey to determine a preferred height for the robot. The survey used full-sized cardboard cut-outs of various heights; participants were simply asked which height they would prefer for an interactive mobile robot. The general opinion was for the robot to be approximately 4.5' (1.4 m) tall, making it shorter than most adults but taller than most children.

Concurrent with the design of the shell was the decision to utilize an LCD to display a graphical face, rather than the fixed LED features shown in the early sketches. Such fixed features can convey only minimal expressions (such as color

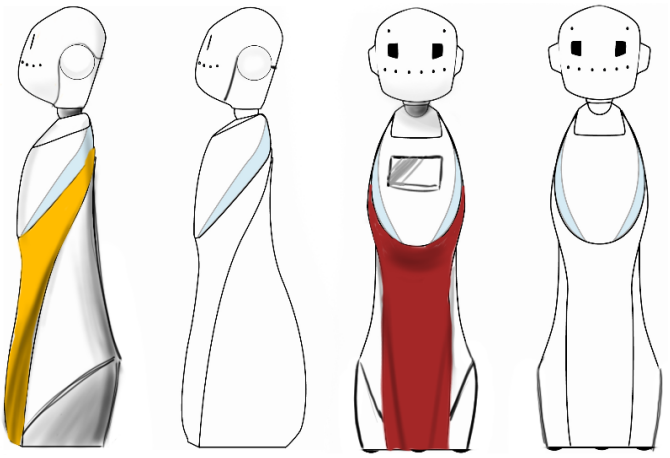


Figure 7.8: Early design sketches for Companion by Scott Smith.

changes) that are often difficult to understand. A graphical face, in contrast, is capable of a much wider range of expressions. Graphical faces are also easily changed, allowing for more experimentation with the robot. (Note that we focused our design on non-mechanical faces. Though a mechanical face may in some cases be more compelling than either a fixed-display or a graphical face, such a face is composed of many moving parts, and is thus difficult both to create and to maintain.) However, rather than rely on the "head in a monitor" type display as used by Grace (see Figure 3.3 in Chapter 3), we instead selected a small monitor that could be mounted inside a more organically-shaped head. In particular, we selected a 10.4" LCD manufactured by CMO[4] (part number G104x1-L01). This LCD panel has a very high contrast ratio (1200:1) and wide viewing angle ($\pm 88^\circ$ in all directions), which allows the face to be seen well from all sides. We purchased the LCD with a mounting bracket and electronics from Industrial Electronic Displays, Inc.[5] for $425. The choice of display set a minimum size for the robot's head, which is reflected in the majority of sketches. Figure 7.9 depicts an exploration of some simple emotive facial expressions.

As we explored options for materials, we followed the lessons learned from the Snackbot robot design (Lee et al., 2009), planning to reduce weight by replacing parts of the hard shell with soft fabric, as shown in Figure 7.10. This began to make the separation between the robot's torso and base parts more apparent. To simplify the design even further, we chose to make the base and torso completely

[4]CMO, now Chimei InnoLux: http://www.chimei-innolux.com/opencms/cmo/index.html?__locale=en

[5]Industrial Electronic Displays: http://www.industrialdisplays.com

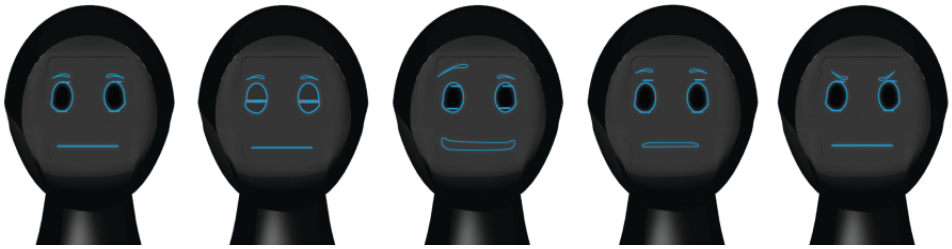


Figure 7.9: Ideas for a simplistic face display for Companion; by Scott Smith.

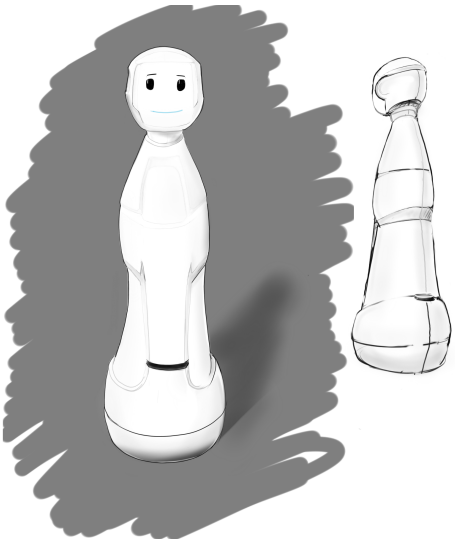
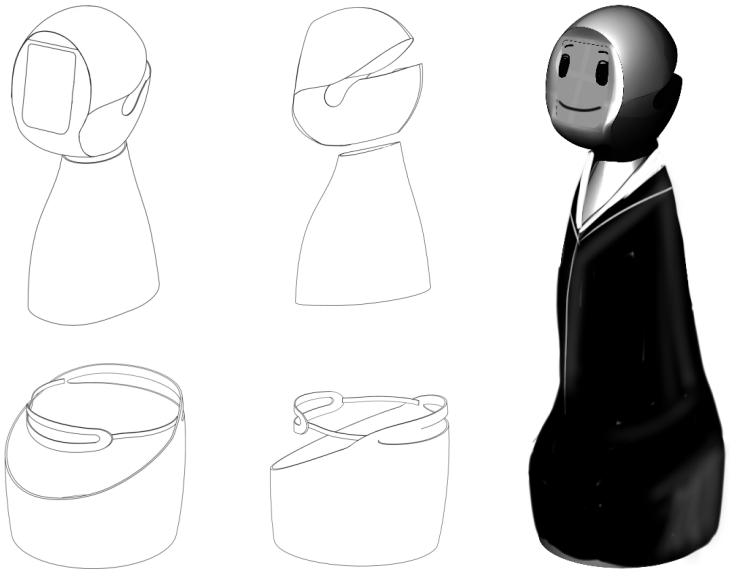


Figure 7.10: Early design sketches for Companion, resulting from the decision to take away some of the hard shell and replace it with fabric (around the sides); by Scott Smith.

separate pieces, reducing much of the bulk of the shell. A CAD model of this design is shown in Figure 7.11. This design was carved out of a solid blue foam material and informally shown to colleagues. The overwhelming response was that the head was too large, resembling either a space helmet or the top of a bowling pin. The head size was scaled back in the final design, as discussed below.

7.2.2 Final design

In response to comments about the robot's head size, undergraduate design students Josh Finkle and Erik Glaser redesigned the head, resulting in the model show in



(a) Front view (b) Side view (c) "Dressed"

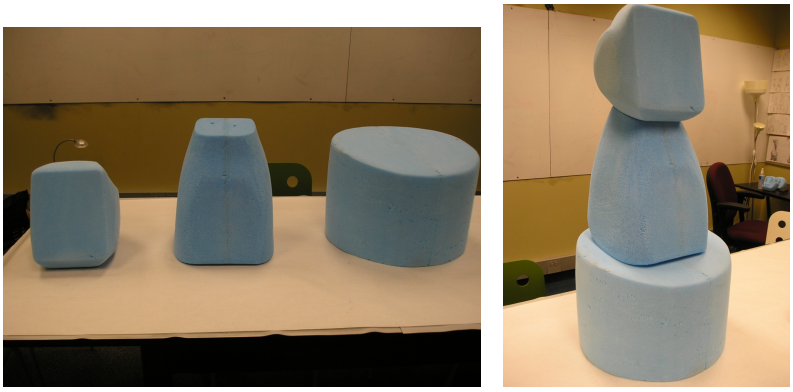
Figure 7.11: CAD model of a late version of the Companion housing. The space between the torso and base is meant to be covered with fabric, as shown in (c). Design by Scott Smith.

Figure 7.12. This model was taken to a local manufacturer, Outlaw Performance, Inc., that created fiberglass forms from the foam models. The production costs were $11,000 to generate molds based on the foam models, and $1000 for the set of three fiberglass body pieces. Additional body pieces can be manufactured from the existing molds, should damage occur.

Color options for the shell were limited to a set of pre-mixed colors available from the manufacturer. We selected a light teal color ("seafoam") because it was bright (and thus visible), but not obtrusively so (such as a safety-cone orange might be).

The base housing piece contains internal fiberglass ledges that sit flush on the top plate of the robot base. The base piece is then secured with bolts directly to the plate. The back of the base piece can be removed to access the robot's batteries, as shown in Figure 7.13. The torso and head pieces mount to a long pole of 80/20 extruded aluminum.[6] To mount the torso, mechanical engineer Roni Cafri designed

[6] 80/20 Inc: http://www.8020.net/



(a) (b)

Figure 7.12: Final model of the housing for Companion cut from blue foam, by Josh Finkle and Erik Glaser. While the torso is not meant to sit directly on the base, (b) is intended to give an idea of the overall robot shape.

an armature from sheet metal; the torso fiberglass fits snugly over the armature, and is secured with Velcro® strips. This mount is shown in Figure 7.14.

The completed robot (other than fabric coverings) is shown in Figure 7.15. In keeping with the desire for an organic shape, fabric will be used to cover the area between the torso and base, as well as the neck. A suitable fabric is still under research, though we anticipate the use of a micro-mesh to allow air flow around the electronics. A simple mock-up with cotton muslin is shown in Figure 7.15(b).

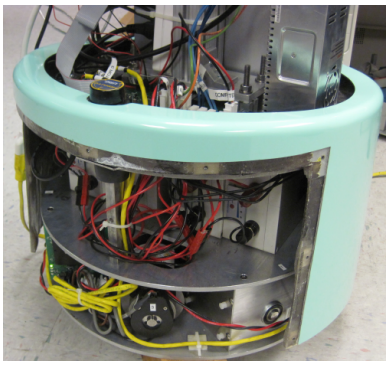
7.3 Summary

The Companion robot is a new platform for social robotics research. Its key features include a holonomic base and a fiberglass housing. The robot was designed to support social navigation—moving around people in socially acceptable ways. The design process was lead by the author, in order to best support the research directions of this thesis. The author oversaw all aspects of the robot's development and drove the design decisions.

The base of the robot is capable of producing fast movements in any direction, similar to people's abilities. Since it can move sideways as well as forwards and on arcs, it can side-step around obstacles without having to turn, which is an important behavior socially, as we showed in our user studies with Grace (Chapter 5).

The robot's body was designed through an iterative process involving several design students. The final housing design is composed of three pieces: a head, a

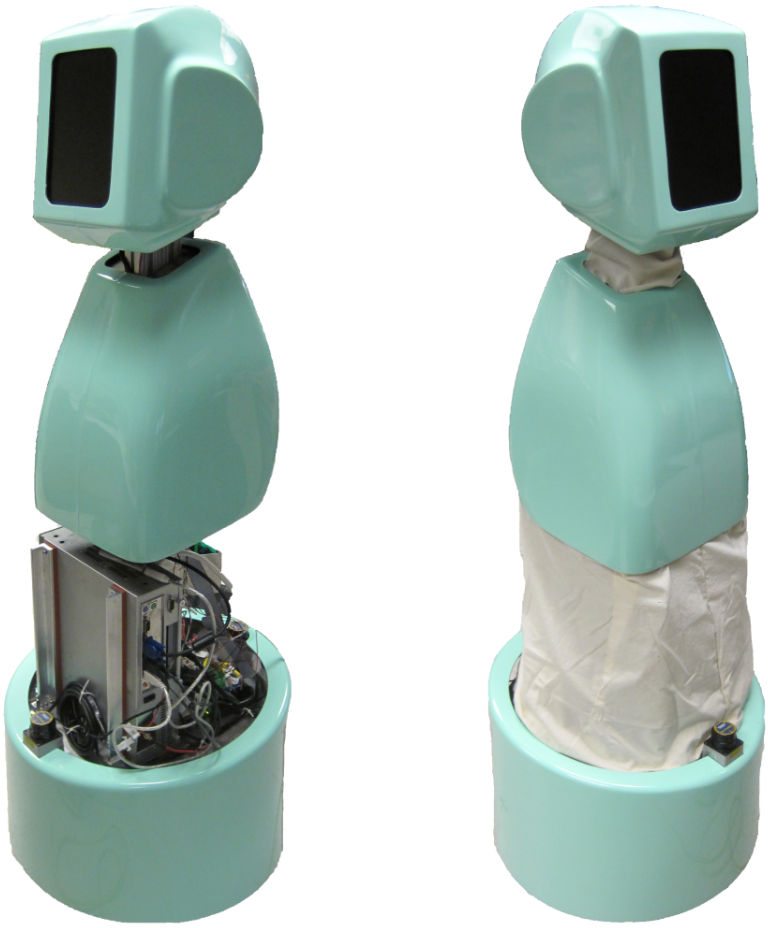
(a) Base housing piece mounted to the robot.



(b) Open battery access panel on the base.

Figure 7.13: The robot body piece that covers the base of the robot.

Figure 7.14: The mounting mechanism for the torso body piece is composed of a sheet-metal armature that fits onto the 80/20 pole. The mount was designed by Roni Cafri.



(a) Electronics exposed

(b) Mock-up of fabric covering

Figure 7.15: The Companion robot, with the fiberglass body mounted and electronics exposed. During operation, the components on the base will be covered with fabric. The completed height is approximately 4'8" (1.4 m).

torso, and a base. All pieces were formed in fiberglass by a local manufacturer. Space was left between the pieces to reduce weight and manufacturing effort, as well as to incorporate fabric coverings into the design. The robot's head holds an LCD panel for displaying a graphical face, which can serve as a locus of interaction.

Several aspects of the robot are still under development. Firmware for the custom circuit boards is still being refined by the author: the circuitry can sense a number of features of the robot's state (such as power usage and remaining battery life) that are not currently being reported in a meaningful way. These features need to be communicated to the robot's on-board computer so that it can react appropriately to events like low batteries or stalled motors. Furthermore, we need to either extend CARMEN or migrate to a different robot control framework in order to support the robot's holonomic capabilities.[7]

Many avenues of research will be available for exploration with the Companion robot. In particular, we intend to use Companion for further research on the COMPANION framework (see Chapter 8). It can additionally be used for face-to-face interaction research, as well as many other forms of social human–robot interaction.

7.4 Acknowledgements

The Companion project was a collaboration between many people, including the author, faculty members Jodi Forlizzi and Reid Simmons; research staff members Brian Kirby, Ben Brown, and Greg Armstrong; design students Scott Smith, Josh Finkle, and Erik Glaser; electrical engineering student David Bromberg; and mechanical engineer Roni Cafri. Companies involved in the robot design and manufacture include Botrics, LLC (consulting on the base design), Advanced Motion Controls (educational discount for the motor controllers), and Outlaw Performance (shell manufacturing).

Finally, the Companion project drew funding from numerous sources, including NSF CNS grant #0709077 to Sara Kiesler, an NPRP grant from the Qatar National Research Fund,[8] an NSF IGERT Graduate Research Fellowship, and the Quality of Life Technology Center in Pittsburgh, PA.

[7] Recall from Section 5.1.5 that the motion model used by CARMEN's localization module does not support sideways maneuvers.

[8] A copy of the robot's head is being used as part of a robotic receptionist on the Carnegie Mellon campus in Qatar.

Chapter 8

Future Work

The contributions of this thesis, including the COMPANION navigational framework, its extension to joint path planning, and the Companion robot, are all intended as foundational work for future social human–robot interaction research. In this chapter, we address current limitations of the work, as well as several directions for further research.

8.1 Limitations of the current work

The focus of this thesis has been on the integration of human social conventions in robot path planning, in the form of the COMPANION framework. However, the current implementation of the framework has several limitations that must be addressed before the framework can be used as part of a complete system. In particular, the primary limitations relate to real-time operation and to person tracking.

8.1.1 Real-time planning

In order to be used in real-time planning and navigation, the COMPANION framework needs to be able to generate new paths whenever new sensory information is received—which is typically several times per second. However, the current implementation of the framework achieves such rates only at the expense of optimality, using the techniques described in Section 4.4.2. Furthermore, even with such techniques, the joint planning extension runs several orders of magnitude too slowly for real-time use.

Several methods may be useful in improving the run-time of the COMPANION framework. For example, the A* search can be made to execute more rapidly with the use of parallelization techniques on multi-core processors (e.g., Cvetanovic and

Nofsinger, 1990). The search itself may also be better optimized with the use of state lattices instead of an 8-connected grid (Pivtoraiko et al., 2009). Furthermore, though we initially rejected randomized planners because they do not produce optimal paths (see Section 4.1), the increase in speed that can be obtained from random planners may be worth the reduction in optimality. Finally, we note that computer processor speeds have increased rapidly over time, and continued hardware advances may result in real-time execution of even the current implementation.

An alternate approach to improving the search speed is to relax the requirement for global planning. In Section 4.1, we argue that the robot must react to all obstacles (including people) in an intentional, goal-directed manner. To a large extent, such behavior can be achieved by using a fast, high-level planner (that perhaps considers only static obstacles) in order to provide very short-term goals to the COMPANION planner. This method may fail to provide optimality over greater distances, such as when the globally optimal path requires a significantly different path than the shortest-distance path (such as described in Section 5.1.4). However, the resulting behavior may sufficiently adhere to human social norms for acceptable robot navigation.

8.1.2 Person detection and tracking

Another limitation of the COMPANION framework is the current state of person detection and tracking. Since a key tenet of the framework is that people must be treated as social entities (rather than just obstacles), the robot must be able to accurately detect where people are in the environment. Unfortunately, the laser-based tracking system we currently employ (see Section 4.4.3) performs quite poorly in practice. The tracker could be improved in many ways, most notably by using a multi-sensor approach to better determine the locations of people in the environment. That is, while a laser provides fairly accurate range readings, determining which readings correspond to people is difficult. By combining the laser ranges with, for example, a vision system that detects people by shape, the tracker could achieve greater accuracy.

In addition, the path planner must predict people's future trajectories in order to plan optimal paths around them. While the current method of assuming straight-line travel does result in socially acceptable behavior, we expect that the behavior could be improved with better trajectory prediction. One method of better prediction is to learn likely trajectories in a given environment (e.g., Kanda et al., 2009; Ziebart et al., 2009). Furthermore, the robot may be able to employ a type of reflective navigation (Kluge and Prassler, 2004) to predict how people may change their trajectories based on the robot's actions.

8.2 Additional on-robot experiments

While this thesis has presented a wide range of results from simulations, the actual behavior of a physical robot was addressed in only a single scenario (Section 5.2). This resulted from many factors, most notably the difficulties in running the complete system in real-time and limited resources for user trials. Once the hurdle of real-time operation has been addressed (Section 8.1), more on-robot experiments can be run. In particular, interesting user studies include (but are certainly not limited to):

- a behavioral analysis of the robot in situations beyond the head-on encounter, such as overtaking a person or navigating through crowds;
- a mapping of constraint weights to robot "personalities;"
- user preferences for different robot behaviors in different situations, such as different environments.

Furthermore, the experiment presented in this thesis was performed with the robot Grace, rather than the new Companion robot (Chapter 7). This was done for multiple reasons—primarily because the Companion robot was not yet operational, and additionally due to the need to re-write CARMEN's localization module to support holonomic movements, which was beyond the scope of this thesis. However, an obvious direction for future research is to run user studies with Companion. Beyond the research topics listed above, further research could compare the differences between Grace and Companion, particularly relating to people's perceptions of the robot's behaviors.

8.3 Learning constraint weights

In all of the results presented in this thesis, the weights assigned to each constraint were assigned by hand. While we have sought to address ways in which the robot's behavior changes due to different weights, future research could further quantify these changes. In particular, one interesting avenue of research could involve applying machine learning to the constraint weighting problem. We imagine that the robot could be tele-operated to produce social behavior according to an operator's preferences. The paths produced by the operator could then be used as training data to learn a set of constraint weights that would produce similar behavior. That is, the problem of computing constraint weights based on a desired behavior could be treated as an "inverse reinforcement learning" problem (Ng and Russell, 2000).

8.4 Additional constraints

While we have argued that the constraints presented in this thesis produce socially acceptable behavior, we acknowledge that people do employ many additional social conventions when walking around others. For example, people change their behaviors according to whether another person is a friend or stranger, their relative social status, gender, and so on. Many of these behaviors can be represented in the COMPANION framework, with the requirement that the robot be able to detect such relationships. In addition, conventions that correspond to other capabilities, such as speech or gaze, could be added to the framework. Future research could work both to identify (and implement) interesting social conventions and to understand when their use may be beneficial to human–robot interaction.

8.5 Additional tasks

We believe that the COMPANION framework, particularly with the joint planning extension, can represent a wide variety of social situations. Future research could identify and implement different social tasks as well as seek to understand the limits of what the framework can represent. Some tasks that we think could be represented with COMPANION include side-by-side following, standing in line, and entering and exiting elevators.

8.5.1 Side-by-side following

The task of following someone side-by-side is similar to the side-by-side escorting task presented in Chapter 6, but with the roles reversed. The constraints for traveling side-by-side—"walk with a person" and "remain side-by-side"—are the same, whether the robot is leading or following. However, when following rather than leading, the final goal is *unknown* to the robot. Instead, the robot must dynamically predict its desired goal location based on the person's movements. The robot will need to estimate the person's location at some point in the near future, probably on the order of 1–2 meters ahead (though the optimal distance likely depends on travel speed and other factors, and is a subject for further research). Additional constraints might be necessary for the robot to react properly to cues from the person (such as verbal directions). As long as the robot is able to predict a relatively likely future location of the person, the COMPANION framework should generate social paths for following next to a person.

8.5.2 Standing in line

Another socially-guided task that a robot may need to execute is that of waiting in a line, which we also believe can be represented in the COMPANION framework. In this case, the goal is typically the counter at the front of the line, but the robot must be constrained to remain in line. In particular, the planner will need the addition of a *hard* "stay in line" constraint that denies movement around the line. The line itself would need to be tracked in some way in order to compute the sides of the line as well as the end of the line, where the robot may enter. The existing "personal space" and "robot 'personal' space" constraints will allow the robot to maintain proper spacing in the line.

Note that for this task, the robot may need to have some model of how people move in a line—under the current person-prediction model, stationary people (such as those waiting in line) are assumed to remain stationary, so if the robot is forbidden from leaving the line, and the people are assumed not to move, the path planner will declare failure to plan to the goal. If the robot assumes that people will require some set amount of time at the front of the line and then move away, the planner should be able to generate trajectories of the form "remain stopped until a person finishes, then move forward."

Interestingly, if the "stay in line" constraint is formed as a *soft* rather than hard constraint, it may allow the robot to *cut* in line, if either a large enough gap is detected between people (so the "personal space" constraint cost is not overwhelming) or if the line moves much more slowly than the robot had predicted. Whether a particular line allows such behavior tends to be culturally defined (Norman, 2009).

8.5.3 Elevator etiquette

Another task that may be represented in the COMPANION framework is that of riding an elevator. Elevator etiquette (at least in the United States) dictates that people who are already on the elevator should have the opportunity to exit before any additional people enter. However, a person wishing to enter the elevator cannot wait indefinitely for people who might exit, as the elevator doors remain open for only a short while. Once inside the elevator car, people tend to stand around the edges, facing the door. Standing in front of the door, either while outside the elevator waiting to enter or while riding inside, is considered rude, as it may block others' access to the elevator door—unless a person intends to exit the elevator at the next floor, in which case standing by the door indicates his plan.

As with the "standing in line" task, most of the conventions for riding an elevator arise from the already-defined constraints of "personal space" and "robot 'personal' space." An additional constraint relating to the cost of various positions

and orientations within the elevator car will likely be necessary—relying on personal space alone, the lowest-cost positions may have the robot facing the wall, so that its back is toward others on the elevator. Since facing the wall is considered rather anti-social, we would need to define a cost function that favors facing the elevator door.

The main difficulty in representing this task in the COMPANION framework arises from the need to model people's behaviors. In particular, the robot may need a *probabilistic* model of how people enter and exit elevators (Broz et al., 2008). The planner will need to be modified to search in a probabilistic space; that is, the planner should find paths that have low *expected* cost. Such a planner should generate plans in which the robot waits before getting onto an elevator until the probability of anyone trying to exit the elevator is low.

8.6 Summary

In this chapter, we have presented several limitations to the current implementation of the COMPANION framework, namely, the need for improved search speed and person tracking. These limitations result from the implementation only, and are not fundamental to the overall framework. As a result, we believe that the COMPANION framework, as well as the Companion robot, provide an excellent foundation for future research.

Some areas that are particularly interesting for future exploration include performing more on-robot experiments (particularly with Companion), ways of learning constraint weights that produce particular behaviors, and researching additional social conventions that may be represented as constraints. Furthermore, we believe that a wide variety of other social tasks, beyond hallway navigation and escorting people, can be represented in the COMPANION framework. We have suggested several tasks, including side-by-side following, standing in line, and entering and exiting elevators. Future research could work toward implementing these and other tasks. Finally, future work could seek to define and understand the limit of what types of tasks can and cannot be represented in the COMPANION framework.

Chapter 9

Conclusions

This thesis has argued that human social conventions for movement can be represented as mathematical cost functions, and that robots that navigate according to these cost functions are interpreted by people as being socially correct. To support this claim, we developed the COMPANION framework, implemented the task of navigating through hallways, demonstrated this behavior in both simulation and in user studies, and extended the framework to the task of escorting someone while remaining side-by-side. Finally, we support future social robotics research with a new platform, the holonomic Companion robot.

The first contribution of this thesis is the COMPANION framework: a Constraint-Optimizing Method for Person–Acceptable NavigatION. By studying how people navigate around each other, we formulated a key set of social and task-related conventions, represented as mathematical constraints. In particular, we argued that the norms used for general social navigation include:

- Minimizing the distance traveled;
- Avoiding static obstacles;
- Keeping a safety buffer around obstacles;
- Avoiding people, including keeping out of their personal space;
- Protecting the robot's own "personal" space;
- Tending to the right when passing people;
- Keeping a default velocity, so as not to expend extra energy;
- Facing the direction of travel, but allowing for side-stepping obstacles; and

- Maintaining forward inertia.

Drawing on psychological descriptions of human behavior, each of these social norms was described according to a mathematical cost function. The various functions are weighted and combined into a single objective function, which is then used for optimal path planning. This path-planning framework supports the first part of our thesis statement, that social conventions for movement can be represented as mathematical cost functions, and represents the primary contribution of the thesis.

The second contribution of this thesis is an implementation and analysis of the COMPANION framework for hallway navigation tasks. In simulation, we showed that this set of constraints does result in behavior that mimics social norms. Since the constraints are applied to a global path planning problem, the resulting behaviors model the flexible ways that people adhere to social conventions, such as generally tending to the right side of hallways, except when turning left or when another person is in the way. We further showed how different behaviors can be produced by changing the weights of the constraints used, or by modifying the constraints to match conventions of other cultures.

Using the robot Grace, we demonstrated that the robot's behavior, when planning under the COMPANION framework, was interpreted according to human social norms. The robot was seen as generally more social, particularly regarding personal space zones, when it navigated according to all of the social conventions we identified. We showed that participants ascribed different personalities to the same robot depending on its behaviors. These results support the second part of our thesis statement, that robots that navigate according to the cost functions we defined for social conventions are interpreted by people as being socially correct.

In addition, we have contributed an extension to the COMPANION framework, designed for joint tasks between a robot and a person. By generating plans that assume both the robot and the person will follow social norms, the paths created for the robot inherently account for the conventions used in joint tasks. We discussed the necessary changes to the path planner in order to allow such joint planning. Furthermore, we identified the constraints necessary for an escorting task, where the robot is expected to guide a person to a goal while remaining by his or her side. We demonstrated that this approach works to produce behaviors such as speeding up or slowing down when going around corners, as well as traveling through narrow doorways where the robot must move in front of the person. This work presents further support for our statement that social conventions—including conventions for joint behaviors—can be represented as mathematical cost functions used in path planning.

Finally, to support the theoretical COMPANION framework and results, we have also contributed the Companion robot, a new platform for social human–robot interaction research. We detailed the design process of the robot, including both the electro-mechanical base and the fiberglass housing. Companion is a holonomic robot, able to move sideways without having to turn first. We believe this to be an important feature for robots that travel around people, because turning aside while passing a person is considered impolite. All aspects of the Companion design process were intended to support social interaction research, and we expect it to be an invaluable resource for future work.

The COMPANION framework and Companion robot are designed as foundational work for future social human–robot interaction research. We have suggested several interesting directions for future work, including additional experiments, other social conventions, and other social tasks. We believe that the findings we have presented in this thesis represent only a small portion of the potential of this research.

Overall, our research has demonstrated the need for robots that operate around people to behave according to human social norms. The COMPANION framework is a representation of these norms in a manner that can be utilized by a robot's path planner to produce socially acceptable behavior around people. While much research remains, we believe that this work will be greatly beneficial to future robots, as well as the people who work with them.

Bibliography

Aiello, J. R. (1987). Human spatial behavior. In Stokols, D. and Altman, I., editors, *Handbook of Environmental Psychology*, volume 1, pages 389–504. John Wiley & Sons, New York.

Aiello, J. R. and Thompson, D. E. (1980). Personal space, crowding, and spatial behavior in a cultural context. In Altman, I., Rapoport, A., and Wohlwill, J. F., editors, *Human Behavior and Environment: Advances in Theory and Research*, volume 4, chapter 4, pages 107–178. Plenum, New York.

Althaus, P., Ishiguro, H., Kanda, K., Miyashita, T., and Christensen, H. I. (2004). Navigation for human-robot interaction tasks. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, pages 1894–1900, New Orleans, LA.

Arulampalam, S., Maskell, S., Gordon, N., and Clapp, T. (2002). A tutorial on particle filters for on-line non-linear/non-Gaussian Bayesian tracking. *IEEE Transactions of Signal Processing*, 50(2):174–188.

Ashton, N. L. and Shaw, M. E. (1980). Empirical investigations of a reconceptualized personal space. *Bulletin of the Psychonomic Society*, 15(5):309–312.

Baxter, J. C. (1970). Interpersonal spacing in natural settings. *Sociometry*, 33(4):444–456.

Bennewitz, M., Burgard, W., Cielniak, G., and Thrun, S. (2005). Learning motion patterns of people for compliant robot motion. *International Journal of Robotics Research*, 24(1):31–48.

Bennewitz, M., Burgard, W., and Thrun, S. (2003). Adapting navigation strategies using motions patterns of people. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2000–2005.

Bethel, C. L., Salomon, K., and Murphy, R. R. (2009). Preliminary results: Humans find emotive non-anthropomorphic robots more calming. In *Proceedings of Human–Robot Interaction*, pages 291–292, La Jolla, CA.

Bianco, R., Caretti, M., and Nolfi, S. (2003). Developing a robot able to follow a human target in a domestic environment. In Cesta, A., editor, *Proceedings of the First RoboCare Workshop*, pages 11–14, Rome, Italy.

Bitgood, S. and Dukes, S. (2006). Not another step! Econonmy of movement and pedestrian choice point behavior in shopping malls. *Environment and Behavior*, 38(3):394–405.

Borenstein, J. and Koren, Y. (1989). Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5):1179–1187.

Bradley, M. M. and Lang, P. J. (1994). Measuring emotion: The self-assessment manikin and the semantic differential. *Journal of Behavioral Therapy and Experimental Psychiatry*, 25(1):49–59.

Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter. *IBM Systems Journal*, 4(1):25–30.

Broz, F., Nourbkahsh, I., and Simmons, R. (2008). Planning for human-robot interaction using time-state aggregated pomdps. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*, volume 3, pages 1339–1344, Chicago, IL.

Bruce, A. and Gordon, G. (2004). Better motion prediction for people-tracking. In *Proceedings of the IEEE International Conference on Robotics and Automation*.

Bruce, A., Nourbakhsh, I., and Simmons, R. (2002). The role of expressiveness and attention in human-robot interaction. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4138–4142.

Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and Thrun, S. (1999). Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3–55.

Burgess, J. W. (1983). Interpersonal spacing behavior between surrounding nearest neighbors reflects both familiarity and environmental density. *Ethology and Sociobiology*, 4:11–17.

Burgoon, J. K., Buller, D. B., and Woodall, W. G. (1989). *Nonverbal Communication: The Unspoken Dialogue*. Harper & Row, New York.

Castro, D., Nunes, U., and Ruano, A. (2002). Obstacle avoidance in local navigation. In *Proceedings of the IEEE Mediterranean Conference on Control and Automation*, Portugal.

Castro, D., Nunes, U., and Ruano, A. (2004). Feature extraction for moving objects tracking system in indoor environments. In *Proceedings of the IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, pages 329–334, Lisbon, Portugal.

Chakrabarti, P. P., Ghose, S., and DeSarkar, S. C. (1987). Admissibility of AO* when heuristics overestimate. *Artificial Intelligence*, 34:97–113.

Clark, H. H. (1996). *Using Language*. Cambridge University Press, Cambridge.

Clark, H. H. and Brennan, S. E. (1991). Grounding in communication. In *Perspectives on Socially Shared Cognition*, pages 127–149. APA.

Coulter, R. C. (1992). Implementation of the pure pursuit path tracking algorithm. Technical Report CMU-RI-TR-92-01, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Cui, J., Zha, H., Zhao, H., and Shibasaki, R. (2006). Robust tracking of multiple people in crowds using laser range scanners. In *Proceedings of the IEEE International Conference on Pattern Recognition (ICPR)*, pages 857–860.

Cvetanovic, Z. and Nofsinger, C. (1990). Parallel Astar search on message–passing architectures. In *Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences*, volume 1, pages 82–90.

Ducourant, T., Vieilledent, S., Kerlirzin, Y., and Berthoz, A. (2005). Timing and distance characteristics of interpersonal coordination during locomotion. *Neuroscience Letters*, 389(1):6–11.

Eliazar, A. I. and Parr, R. (2004). Learning probabilistic motion models for mobile robots. In *Proceedings of the Twenty First International Conference on Machine Learning (ICML)*.

Feghali, E. (1997). Arab cultural communication patterns. *International Journal of Intercultural Relations*, 21(3):345–378.

Fiorini, P. and Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. *International Journal of Robotics Research*, 17(7):760–772.

Foka, A. (2005). *Predictive Autonomous Robot Navigation*. PhD thesis, University of Crete.

Foka, A. F. and Trahanias, P. E. (2003). Predictive control of robot velocity to avoid obstacles in dynamic environments. In *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, pages 370–375, Las Vegas, Nevada.

Fox, D., Burgard, W., and Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*, 4(1):23–33.

Fraichard, T. (1999). Trajectory planning in a dynamic workspace: a 'state-time space' approach. *Advanced Robotics*, 13(1):75–94.

Frith, C. D. and Frith, U. (2006). How we predict what other people are going to do. *Brain Research*, 1079:36–46.

Fujimura, K. and Samet, H. (1989). A hierarchical strategy for path planning among moving obstacles. *IEEE Transactions on Robotics and Automation*, 5(1):61–69.

Gérin-Lajoie, M., Richards, C. L., and McFadyen, B. J. (2005). The negotiation of stationary and moving obstructions during walking: Anticipatory locomotor adaptations and preservation of personal space. *Motor Control*, 9(3):242–269.

Gilbert, M. (1990). Walking together: a paradigmatic social phenomenon. *Midwest Studies in Philosophy*, 15:1–14.

Gockley, R. (2007). Developing spatial skills for social robots. In *Proceedings of the AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics*, pages 15–17, Palo Alto, CA.

Gockley, R., Forlizzi, J., and Simmons, R. (2006). Interactions with a moody robot. In *Proceedings of Human-Robot Interaction*, pages 186–193, Salt Lake City, Utah.

Gockley, R., Forlizzi, J., and Simmons, R. (2007). Natural person-following behavior for social robots. In *Proceedings of Human–Robot Interaction*, pages 17–24, Arlington, VA.

Gockley, R. and Matarić, M. (2006). Encouraging physical therapy compliance with a hands-off mobile robot. In *Proceedings of Human-Robot Interaction*, pages 150–155, Salt Lake City, Utah.

Hall, E. T. (1966). *The Hidden Dimension*. Doubleday, New York.

Hall, E. T. (1974). Proxemics. In Weitz, S., editor, *Nonverbal Communication: Readings with Commentary*, pages 205–227. Oxford University Press, New York.

Hart, P. E., Nilsson, N. J., and Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths in graphs. *IEEE Transactions on Systems Science and Cybernetics*, SSC-4(2):100–107.

Hoffman, G. and Breazeal, C. (2004). Robots that work in collaboration with people. In *Proceedings of the 2004 CHI Workshop on Shaping Human–Robot Interaction*, Vienna.

Ikeura, R., Monden, H., and Inooka, H. (1994). Cooperative motion control of a robot and a human. In *IEEE International Workshop on Robot and Human Communication*, pages 112–117.

Jan, D., Herrera, D., Martinovski, B., Novick, D., and Traum, D. (2007). A computational model of culture-specific conversational behavior. In *Lecture Notes in Computer Science: Intelligent Virtual Agents*, volume 4722, pages 45–56. Springer Berlin.

Kanda, T., Glas, D. F., Shiomi, M., and Hagita, N. (2009). Abstracting people's trajectories for social robots to proactively approach customers. *IEEE Transactions on Robotics*, 5(6):1382–1396.

Kanda, T., Hirano, T., Eaton, D., and Ishiguro, H. (2003). Person identification and interaction of social robots by using wireless tags. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003)*, pages 1657–1664.

Kendon, A. and Ferber, A. (1990). A description of some human greetings. In *Conducting Interaction*, pages 153–207. Cambridge University Press.

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *International Journal of Robotics Research*, 5(1):90–98.

Kirby, R., Simmons, R., and Forlizzi, J. (2009a). COMPANION: A constraint-optimizing method for person–acceptable navigation. In *Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 607–612, Toyama, Japan.

Kirby, R., Simmons, R., and Forlizzi, J. (2009b). Variable sized grid cells for rapid replanning in dynamic environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent RObots and Systems (IROS)*, pages 4913–4918, St. Louis, MO.

Klein, G., Feltovich, P. J., Bradshaw, J. M., and Woods, D. D. (2005). Common ground and coordination in joint activity. In Rouse, W. B. and Boff, K. R., editors, *Organizational Simulation*. Wiley.

Kleinehagenbrock, M., Lang, S., Fritsch, J., Lömker, F., Fink, G. A., and Sagerer, G. (2002). Person tracking with a mobile robot based on multi-modal anchoring. In *Proceedings of the 2002 IEEE Int. Workshop on Robot and Human Interactive Communication*, pages 423–429, Berlin, Germany.

Kluge, B. (2003). Recursive probabilistic velocity obstacles for reflective navigation. In *Proceedings of the IEEE International Workshop on Advances in Service Robots*, Bardolino, Italy.

Kluge, B. (2004). *Motion Coordination for a Mobile Robot in Dynamic Environments*. PhD thesis, University of Würzburg.

Kluge, B., Illmann, J., and Prassler, E. (2001a). Situation assessment in crowded public environments. In *Proceedings of the International Conference on Field and Service Robitics*, Helsinki, Finland.

Kluge, B., Köhler, C., and Prassler, E. (2001b). Fast and robust tracking of multiple moving objects with a laser range finder. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1683–1688, Seoul, Korea.

Kluge, B. and Prassler, E. (2004). Reflective navigation: Individual behaviors and group behaviors. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 4172–4177, New Orleans, LA.

Ko, N. Y. and Simmons, R. (1998). The lane-curvature method for local obstacle avoidance. In *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems*, pages 1615–1621, Victoria, BC, Canada.

Kobilarov, M., Sukhatme, G., Hyams, J., and Batavia, P. (2006). People tracking and following with mobile robot using an omnidirectional camera and a laser. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 557–562, Orlando, Florida.

Koenig, S. and Likhachev, M. (2002). D* lite. In *Proceedings of the AAAI Conference of Artificial Intelligence (AAAI)*, pages 476–483.

Koenig, S. and Likhachev, M. (2006). Real-time adaptive A*. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 281–288.

Koenig, S., Likhachev, M., Liu, Y., and Furcy, D. (2004). Incremental heuristic search in artificial intelligence. *Artificial Intelligence Magazine*, 25:99–112.

Kozima, H., Nakagawa, C., and Yano, H. (2003). Attention coupling as a prerequisite for social interaction. In *Proceedings of the 2003 IEEE International Workshop on Robot and Human Interactive Communicaton*.

Laugier, C., Petti, S., Vasquez, D., Yguel, M., Fraichard, T., and Aycard, O. (2005). Steps toward safe navigation in open and dynamic environments. In *Proceedings of the IEEE ICRA Workshop on Autonomous Navigation in Dynamic Environments*, Barcelona, Spain.

LaValle, S. M. (1998). Rapidly-exploring random trees: A new tool for path planning. Technical Report TR 98-11, Computer Science Department, Iowa State University.

LaValle, S. M. and Kuffner, Jr., J. J. (1999). Randomized kinodynamic planning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 473–479.

Lee, M. K., Forlizzi, J., Rybski, P. E., Crabbe, F., Chung, W., Finkle, J., Glaser, E., and Kiesler, S. (2009). The Snackbot: Documenting the design of a robot for long-term human-robot interaction. In *Proceedings of Human–Robot Interaction (HRI)*, pages 7–14.

Li, S., Wrede, B., and Sagerer, G. (2006). A computational model of multi-modal grounding for human robot interaction. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 153–160, Sydney, Australia.

Marsh, K. L., Richardson, M. J., Baron, R. M., and Schmidt, R. C. (2006). Contrasting approaches to perceiving and acting with others. *Ecological Psychology*, 18(1):1–38.

Mazur, A. (1977). Interpersonal spacing on public benches in "contact" vs. "noncontact" cultures. *Journal of Social Psychology*, 101:53–58.

McClave, E., Kim, H., Tamer, R., and Mileff, M. (2007). Head movements in the context of speech in Arabic, Bulgarian, Korean, and African-American Vernacular English. *Gesture*, 7(3):343–390.

McPhail, C. and Wohlstein, R. T. (1986). Collective locomotion as collective behavior. *Americal Sociological Review*, 51(4):447–463.

Michalowski, M. P. and Simmons, R. (2006). Multimodal person tracking and attention classification. In *Proceedings of Human–Robot Interaction*, pages 347–348, Salt Lake City, Utah.

Mishra, P. K. (1983). Proxemics: Theory and research. *Perspective in Psychological Researches*, 6(1):10–15.

Montemerlo, M., Pineau, J., Roy, N., Thrun, S., and Verma, V. (2002). Experiences with a mobile robotic guide for the elderly. In *Proceedings of the National Conference of Artificial Intelligence (AAAI)*, pages 587–592, Edmonton, AB.

Mutlu, B. and Forlizzi, J. (2008). Robots in organizations: The role of workflow, social, and environmental factors in human–robot interaction. In *Proceedings of Human–Robot Interaction (HRI)*, pages 287–294.

Nakauchi, Y. and Simmons, R. (2000). A social robot that stands in line. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, pages 357–364.

Ng, A. Y. and Russell, S. (2000). Algorithms for inverse reinforcement learning. In *Proceedings of the 17th International Conference on Machine Learning*, pages 663–670. Morgan Kaufmann.

Norman, D. A. (2009). Designing waits that work. *MIT Sloan Management Review*, 50(4):23–28.

Nourbahksh, I. R., Kunz, C., and Willeke, T. (2003). The mobot museum robot installations: A five year experiment. In *Proceedings of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 4, pages 3636–3641, Las Vegas, NV.

Olivera, V. M. and Simmons, R. (2002). Implementing human-acceptable navigational behavior and a fuzzy controller for an autonomous robot. In *Proceedings WAF: 3rd Workshop on Physical Agents*, pages 113–120, Murcia, Spain.

Owen, E. and Montano, L. (2005). Motion planning in dynamic environments using the velocity space. In *Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems (IROS)*, pages 997–1002, Edmonton, Alberta, Canada.

Pacchierotti, E., Christensen, H. I., and Jensfelt, P. (2005a). Embodied social interaction for service robots in hallway environments. In *Proceedings of the International Conference on Field and Service Robots (FSR)*.

Pacchierotti, E., Christensen, H. I., and Jensfelt, P. (2005b). Human-robot embodied interaction in hallway settings: a pilot user study. In *Proceedings of the IEEE International Workshop on Robots and Human Interactive Communication (RO-MAN)*, pages 164–171, Nashville, TN.

Patterson, M. L., Webb, A., and Schwartz, W. (2002). Passing encounters: Patterns of recognition and avoidance in pedestrians. *Basic and Applied Social Psychology*, 24(1):57–66.

Pin, F. G. and Killough, S. M. (1994). A new family of omnidirectional and holonomic wheeled platforms for mobile robots. *IEEE Transactions on Robotics and Automation*, 10(4):480–489.

Pivtoraiko, M., Knepper, R. A., and Kelly, A. (2009). Differentially constrainted mobile robot motion planning in state lattices. *Journal of Field Robotics*, 26(3):308–333.

Powers, A., Kramer, A., Lim, S., Kuo, J., Lee, S.-L., and Kiesler, S. (2005). Common ground in dialogue with a gendered humanoid robot. In *Proceedings of the IEEE Int. Conf. on Robot and Human Interaction (RO-MAN)*, Nashville, TN.

Prassler, E., Bank, D., and Kluge, B. (2002). Key technologies in robot assistants: Motion coordination between a human and a mobile robot. *Transactions on Control, Automation and Systems Engineering*, 4(1):56–61.

Richardson, M. J., Marsh, K. L., and Schmidt, R. C. (2005). Effects of visual and verbal interaction on unintentional interpersonal coordination. *Journal of Experimental Psychology: Human Perception and Performance*, 31(1):62–79.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall, New Jersey, second edition.

Safadi, M. and Valentine, C. A. (1990). Contrastive analysis of American and Arab nonverbal and paralinguistic communications. *Semiotica*, 82(3/4):269–292.

Sanders, J. L., Hakky, U. M., and Brizzolara, M. M. (1985). Personal space amongst Arabs and Americans. *International Journal of Psychology*, 20:13–17.

Schlegel, C., Illmann, J., Jaberg, K., Schuster, M., and Wörz, R. (1998). Vision based person tracking with a mobile robot. In *Proceedings of the Ninth British Machine Vision Conference (BMVC)*, pages 418–427, Southampton, UK.

Schulz, D., Burgard, W., Fox, D., and Cremers, A. B. (2003). People tracking with a mobile robot using sample-based joint probabilistic association filters. *International Journal of Robotics Research*, 22(2).

Sebanz, N., Bekkering, H., and Knoblich, G. (2006). Joint action: Bodies and minds moving together. *TRENDS in Cognitive Sciences*, 10(2):70–76.

Shi, D., Collins, Jr., E. G., Donate, A., Liu, X., Goldiez, B., and Dunlap, D. (2008). Human-aware robot motion planning with velocity constraints. In *International Symposium on Collaborative Technologies and Systems (CTS)*, pages 490–497.

Shockley, K., Santana, M.-V., and Fowler, C. A. (2003). Mutual interpersonal postural constraints are involved in cooperative conversation. *Journal of Experimental Psychology: Human Perception and Performance*, 29(2):326–332.

Sidenbladh, H., Kragić, D., and Christensen, H. I. (1999). A person following behaviour for a mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 670–675, Detroit, Michigan.

Sidner, C. L. and Dzikovska, M. (2002). Hosting activities: Experience with and future directions for a robot agent host. In *Proceedings of the ACM International Conference on Intelligent User Interfaces*, pages 143–150.

Siino, R. M. and Hinds, P. J. (2004). Making sense of new technology as a lead-in to structuring: The case of an autonomous mobile robot. In *Best Paper Proceedings of the Academy of Management*, New Orleans, LA.

Siino, R. M. and Hinds, P. J. (2005). Robots, gender & sensemaking: Sex segregation's impact on workers making sense of a mobile autonomous robot. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2773–2778, Barcelona, Spain.

Simmons, R. (1996). The curvature-velocity method for local obstacle avoidance. In *Proceedings of the Intl. Conference on Robotics and Automation*, Minneapolis MN.

Simmons, R., Goldberg, D., Goode, A., Montemerlo, M., Roy, N., Sellner, B., Urmson, C., Schultz, A., Abramson, M., Adams, W., Atrash, A., Bugajska, M., Coblenz, M., MacMahon, M., Perzanowski, D., Horswill, I., Zubek, R., Kortenkamp, D., Wolfe, B., Milam, T., and Maxwell, B. (2003). GRACE: An autonomous robot for the AAAI robot challenge. *AAAI Magazine*, 24(2):51–72.

Sisbot, E. A., Marin, L. F., Alami, R., and Simeon, T. (2006). A mobile robot that performs human acceptable motions. In *Proceedings of the IEEE/RJS Conference on Intelligent Robots and Systems*, pages 1811–1816, Beijing, China.

Sisbot, E. A., Marin-Urias, L. F., Alami, R., and Siméon, T. (2007). A human aware mobile robot motion planner. *IEEE Transactions on Robotics*, 23(5):874–883.

Sparrow, W. A. and Newell, K. M. (1998). Metabolic energy expenditure and the regulation of movement economy. *Psychonomic Bulletin and Review*, 5(2):173–196.

Stentz, A. (1994). The D* algorithm for real-time planning of optimal traverses. Technical Report CMU-RI-TR-94-37, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Stubbs, K., Hinds, P., and Wettergreen, D. (2006). Challenges to grounding in human-robot collaboration: Errors and miscommunications in remote exploration robotics. Technical Report CMU-RI-TR-06-32, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Sun, X., Koenig, S., and Yeoh, W. (2008). Generalized adaptive A*. In *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 469–476.

Sviestins, E., Mitsunaga, N., Kanda, T., Ishiguro, H., and Hagita, N. (2007). Speed adaptation for a robot walking with a human. In *Proceedings of Human–Robot Interaction*, pages 349–356, Arlington, VA.

Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hähnel, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. (1999). MINERVA: A second-generation museum tour-guide robot. In *IEEE International Conference on Robotics and Automation (ICRA)*.

Topp, E. A. and Christensen, H. I. (2005). Tracking for following and passing persons. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 70–76, Edmonton, Alberta, Canada.

Trafton, J. G., Cassimatis, N. L., Bugajska, M. D., Brock, D. P., Mintz, F. E., and Schultz, A. C. (2005). Enabling effective human–robot interaction using perspective-taking in robots. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 35(4):460–470.

Urmson, C. and Simmons, R. (2003). Approaches for heuristically biasing RRT growth. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS)*, pages 1178–1183, Las Vegas, Nevada.

Walters, M. L., Dautenhahn, K., te Boekhorst, R., Koay, K. L., Kaouri, C., Woods, S., Nehaniv, C., Lee, D., and Werry, I. (2005). The influence of subjects' personality traits on personal spatial zones in a human-robot interaction experiment. In *Proceedings of CogSci-2005 Workshop: Toward Social Mechanisms of Android Science*, pages 29–37, Stresa, Italy.

Watson, D., Clark, L. A., and Tellegen, A. (1988). Development and validation of brief measures of positive and negative affect: The PANAS scales. *Journal of Personality and Social Psychology*, 54(6):1063–1070.

Watson, O. M. (1970). *Proxemic Behavior: A Cross-Cultural Study*. Mouton, The Hague.

Watson, O. M. and Graves, T. D. (1966). Quantitative research in proxemic behavior. *American Anthropologist*, 68(4):971–985.

Whyte, W. H. (1988). *City: Rediscovering the Center*. Doubleday, New York.

Williams, H. P. (1999). *Model Building in Mathematical Programming*. Wiley, New York, 4 edition.

Wolfinger, N. H. (1995). Passing moments: Some social dynamics of pedestrian interaction. *Journal of Contemporary Ethnography*, 24(3):323–340.

Yahja, A., Stentz, A., Singh, S., and Brumitt, B. L. (1998). Framed-quadtree path planning for mobile robots operating in sparse environments. In *Proc. IEEE Intnl Conf on Robotics and Automation, (ICRA)*, pages 650–655.

Yamato, J., Shinozawa, K., and Naya, F. (2004). Effect of shared-attention on human-robot interaction. In *Proceedings of the 2004 CHI Workshop on Shaping Human–Robot Interaction*, Vienna.

Ziebart, B. D., Ratliff, N., Gallagher, G., Mertz, C., Peterson, K., Bagnell, J. A., Hebert, M., Dey, A. K., and Srinivasa, S. (2009). Planning-based prediction for pedestrians. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3931–3936, St. Louis, USA.

Zucker, M., Kuffner, J., and Branicky, M. (2007). Multipartite RRTs for rapid replanning in dynamic environments. In *Proceedings of the IEEE Int. Conference on Robotics and Automation (ICRA)*.

Appendices

Appendix A

Asymmetric Gaussian Integral Function Definition

Several of the constraints given in Chapter 4 refer to an "Asymmetric Gaussian" function, which we define here. This function is our own formulation to model the shape of several human social conventions, such as personal space.

A standard 1-dimensional Gaussian function is defined in terms of its mean μ and variance σ:

$$f(x) = e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{A.1}$$

In two dimensions, the mean is the center of the function (x_0, y_0), and the variance is represented by two values, σ_x and σ_y:

$$f(x,y) = e^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)} \tag{A.2}$$

A typical 2-dimensional Gaussian function is symmetric along both the x and y axes. We generate a 2-dimensional *Asymmetric Gaussian* by composing two such functions with shared σ_x and differing σ_y values. This reduces the symmetry of the function to only one axis. Furthermore, we allow the function to have an arbitrary rotation, so that it is not necessarily aligned to the x and y axes. We use the following notation:

θ	rotation of the function
σ_h	variance along the θ direction
σ_s	variance to the sides ($\theta \pm \pi/2$ direction)
σ_r	variance to the rear ($-\theta$ direction)

Since the two functions share the value for σ_s along their joining axis, the overall Asymmetric Gaussian function is continuous and smooth.

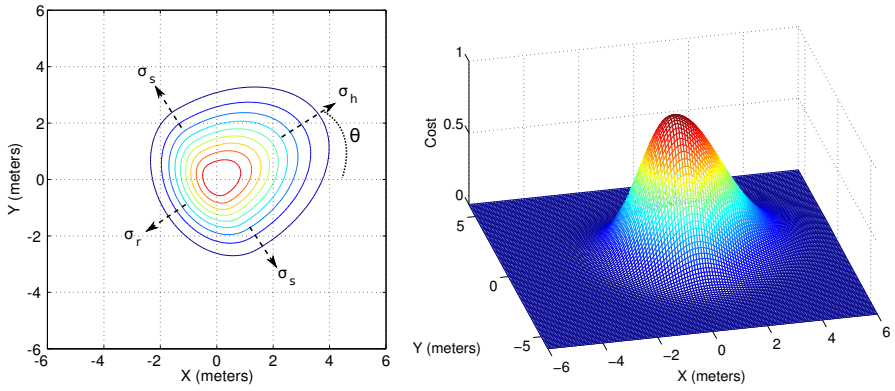
Algorithm A.1 Algorithm to compute the value at (x, y) of an Asymmetric Gaussian function centered at (x_c, y_c), with a rotation of θ and variances of σ_h, σ_s, and σ_r.

1: $\alpha \leftarrow \text{atan2}(y - y_c, x - x_c) - \theta + \pi/2$;
2: Normalize α;
3: $\sigma \leftarrow (\alpha \leq 0 \; ? \; \sigma_r : \sigma_h)$;
4: $a \leftarrow (\cos\theta)^2/(2\sigma^2) + (\sin\theta)^2/(2\sigma_s^2)$;
5: $b \leftarrow \sin(2\theta)/(4\sigma^2) - \sin(2\theta)/(4\sigma_s^2)$;
6: $c \leftarrow (\sin\theta)^2/(2\sigma^2) + (\cos\theta)^2/(2\sigma_s^2)$;
7: **return** $exp(-(a(x - x_c)^2 + 2b(x - x_c)(y - y_c) + c(y - y_c)^2))$;

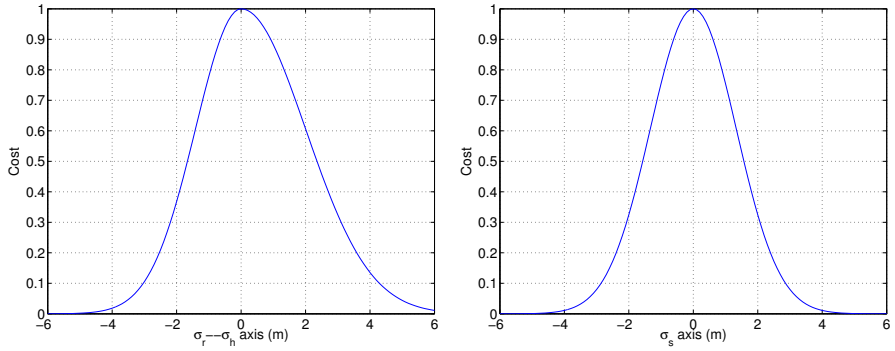
Algorithm A.1 details the computation of the value of an arbitrarily rotated Asymmetric Gaussian at some point (x, y). Lines 1 through 3 compute the normalized angle of the line running in the σ_s direction; that is, α points along the side of the function, and $-\pi < \alpha \leq \pi$. Line 3 determines in which of the two 2D Gaussian functions the point of interest, (x, y), is located. If $\alpha = 0$, the point of interest falls directly to the side of the function center, and thus relies only on σ_s.

Figure A.1 depicts various views of one such Asymmetric Gaussian cost function. The function shown is centered at $(0, 0)$, has a rotation of $\theta = \pi/6$, and has variances $\sigma_h = 2.0$, $\sigma_s = 4/3$, and $\sigma_r = 1.0$. The maximum cost is 1.0 at the center of the function.

The Asymmetric Gaussian function represents a continuous cost function, but it must be discretized for use in the A* search. To further complicate matters, during one timestep, typically both the center of the Gaussian function moves (e.g., the location of the person), as does the robot's location. The true solution would be to integrate the (moving) function over time, but such an integral is intractable for real-time search. Instead, we approximate the integral by sampling the values at k intervals, multiplying each sample by $1/k$ times the timestep. For $k \rightarrow \infty$, this approximation approaches the true integral. For speed of computation, we use $k = 4$.



(a) Contour map (b) Surface map



(c) Plane along the $\sigma_h - \sigma_r$ axis (d) Plane along the σ_s axis

Figure A.1: Various views of an Asymmetric Gaussian function centered at (0, 0), rotated by $\theta = \pi/6$, and having variances $\sigma_h = 2.0$, $\sigma_s = 4/3$, and $\sigma_r = 1.0$.

Appendix B

Simulation Results for Hallway Navigation

The following images represent the paths planned in the simulations described in Section 5.1.1. To summarize, the experiment was run as a set of 3 goal x 3 person location x 3 person speed simulations (27 possible scenarios). The 10 m by 10 m environment contained two hallways: a 3 m wide main corridor and a 2 m wide corridor that intersected the first at right angles. The robot and person each began in the main corridor, approximately 8 m apart. The robot always began in the same location and orientation, with a preferred speed of 0.5 m/s. The three possible goals were: a right turn down the intersecting hallway, a left turn down the intersecting hallway, or straight ahead. The person began either 0.5 m to the left of the robot (i.e., on the right of the hallway from the person's perspective), centered in the hallway (aligned with the robot), or 0.5 m to the right of the robot (i.e., on the left of the hallway from the person's perspective), and traveled at a constant speed of either 0.3 m/s, 0.5 m/s, or 0.7 m/s.

In the following images, the robot is depicted as a blue circle, the goal as a yellow circle, and the person as an orange circle. In each set of images, the first image (a) depicts the path as planned on a constant grid (cells 10 cm on a side), using the constraints and weights given in Table B.1. Overlaid on these images is the positions of the robot and the person when they are closest to each other on the paths. The second image in each set (b) was planned using the same constraints and weights, but with the addition of the speed improvement methods described in Section 4.4, namely:

- The size of the grid varied, becoming increasingly coarse further from the robot. The cell sizes used are given in Table B.2.

Table B.1: Constraint weights used in the objective function. In addition, the hard constraints of avoiding obstacles and people were used.

Constraint Name	**Weight (w_c)**
Minimize distance	1
Obstacle buffer	1
Personal space	2
Robot space	3
Pass on right	2
Default velocity	2
Face travel	2
Inertia	2

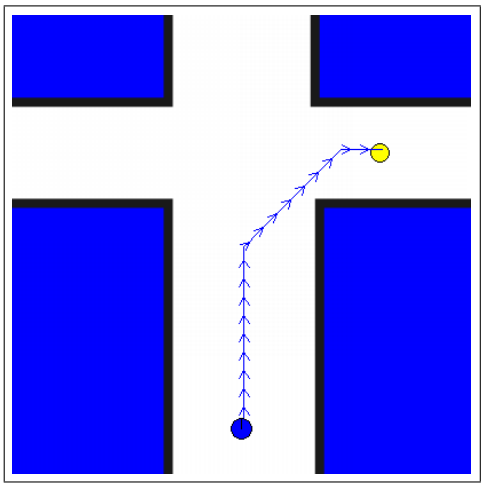
Table B.2: Variable search grid sizing.

Distance from Robot	**Cell Dimensions**
less than 1 m	0.1×0.1 m
between 1 and 3 m	0.3×0.3 m
greater than 3 m	0.6×0.6 m

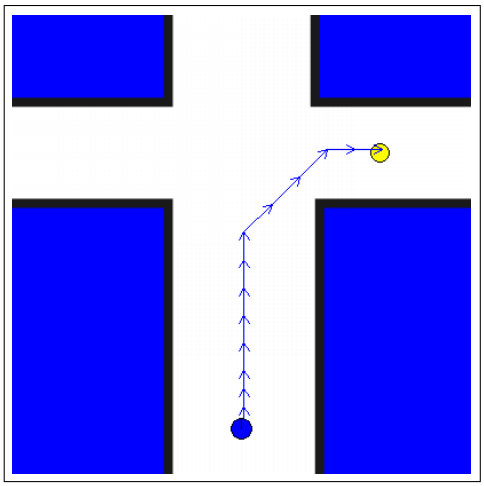
- At distances beyond the smallest grid sizing (that is, beyond 1 m), the action space was reduced to only forward, left turn, and right turn.
- Once the search identified a state in which the robot had passed the person, the person was dropped from searches outward from that state.
- A shortest-distance gradient was imposed on the search such that the robot could not deviate from the gradient by more than a fixed amount at each step.

Arrows along the paths represent the robot's (or person's) heading, and are drawn every 40 cm along the paths on the constant grid, or at a minimum of 40 cm (or at each path step, whichever is larger) on the variable grid.

Refer to Section 5.1.1 for statistics regarding all paths.

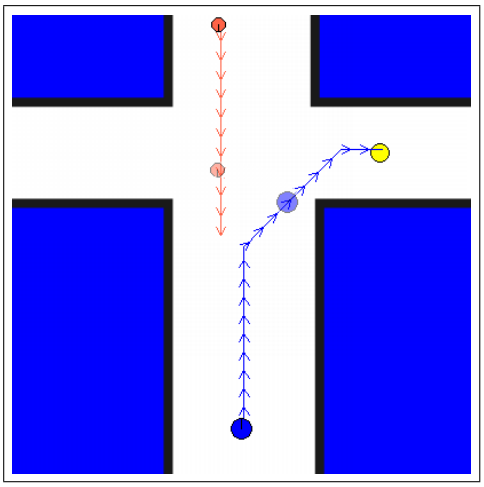


(a) Constant grid

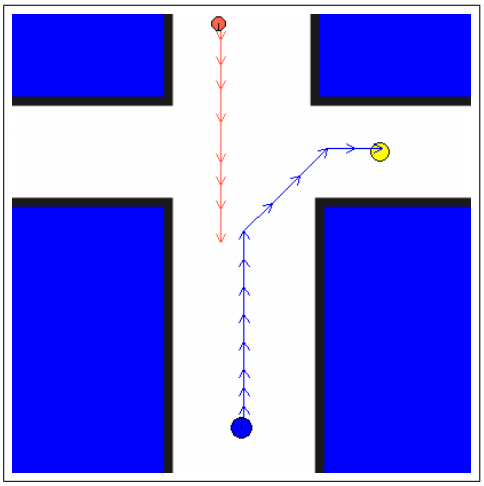


(b) Variable grid

Figure B.1: Path planned for a goal requiring the robot to turn right down a hallway.

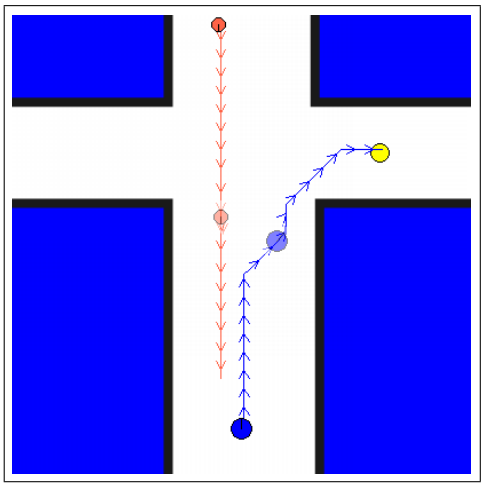


(a) Constant grid, with closest encounter marked

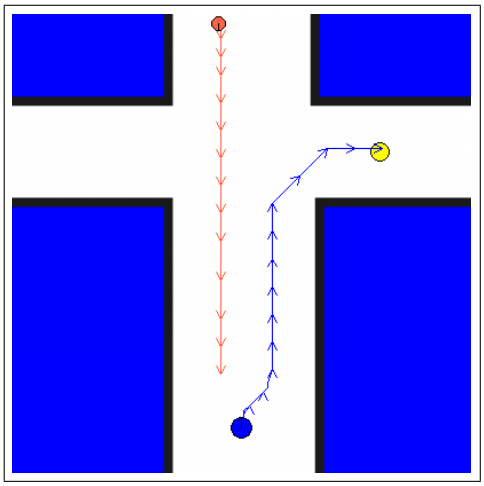


(b) Variable grid

Figure B.2: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the left of the hallway at a speed of 0.3 m/s. Figure (a) depicts the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

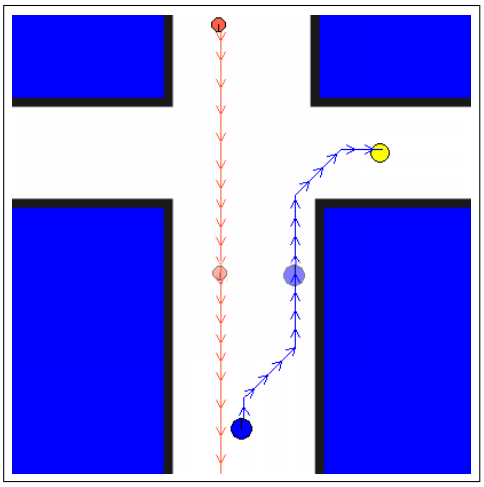


(a) Constant grid, with closest encounter marked

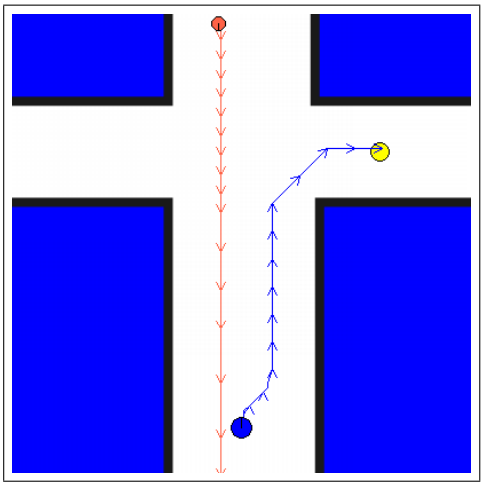


(b) Variable grid

Figure B.3: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the left of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

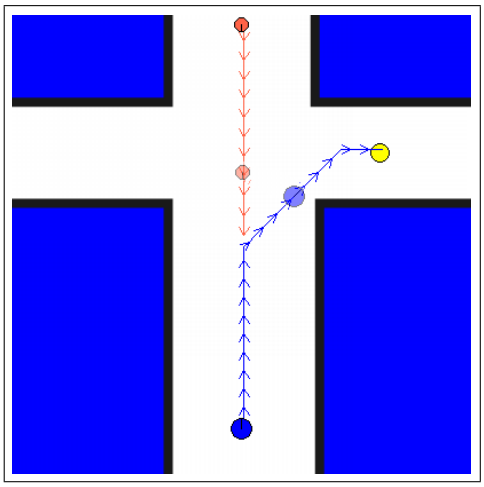


(a) Constant grid, with closest encounter marked

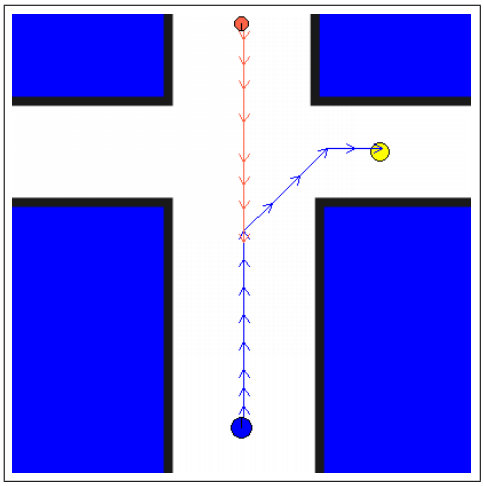


(b) Variable grid

Figure B.4: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the left of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. The robot moves further to the right than in Figure B.3 due to the larger personal space of the faster-moving person.

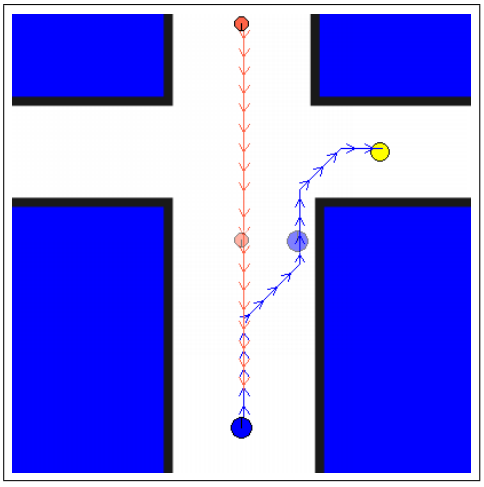


(a) Constant grid, with closest encounter marked

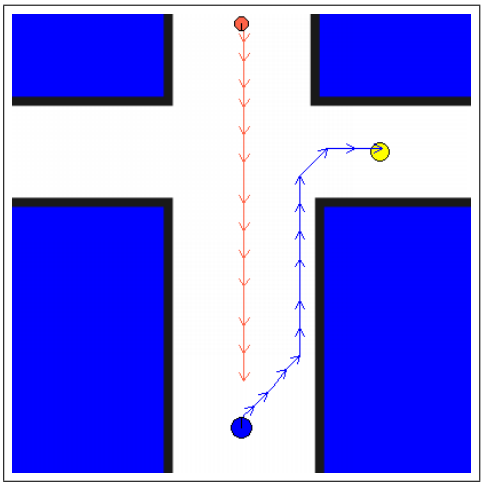


(b) Variable grid

Figure B.5: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the center of the hallway at a speed of 0.3 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

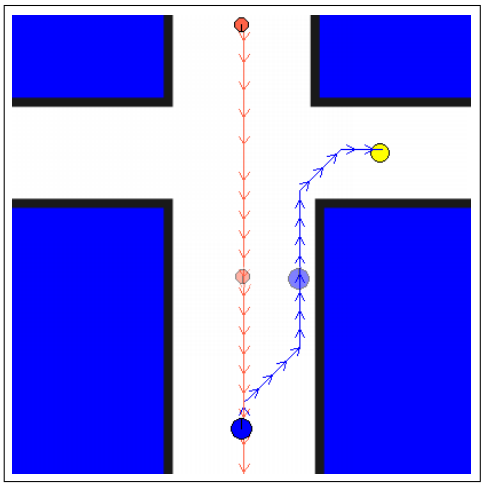


(a) Constant grid, with closest encounter marked

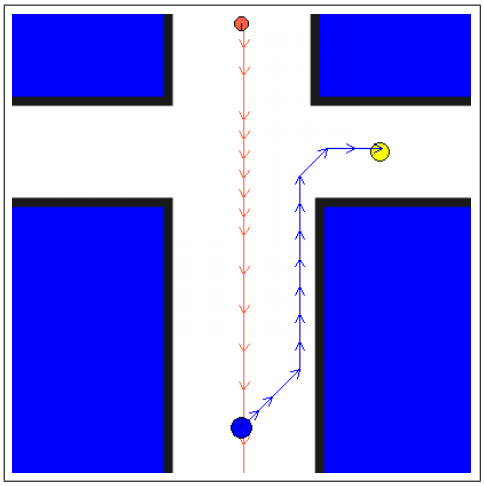


(b) Variable grid

Figure B.6: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the center of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. The robot moves close to the wall to avoid the person.

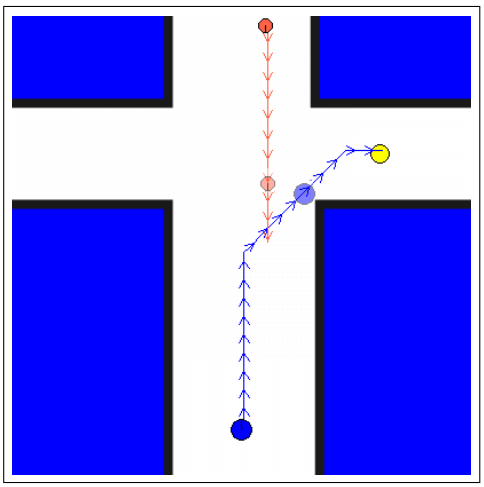


(a) Constant grid, with closest encounter marked

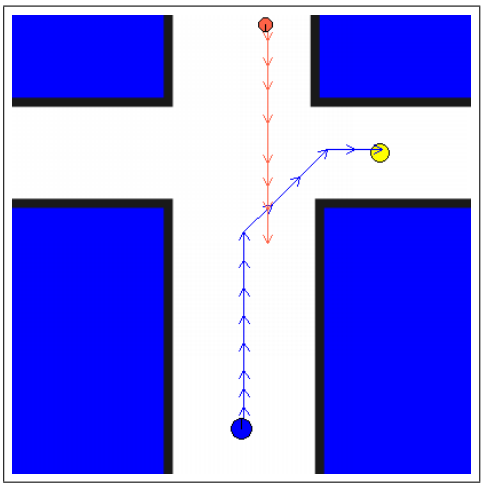


(b) Variable grid

Figure B.7: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the center of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. The robot moves close to the wall to avoid the person, turning much sooner in the path than in Figure B.6.

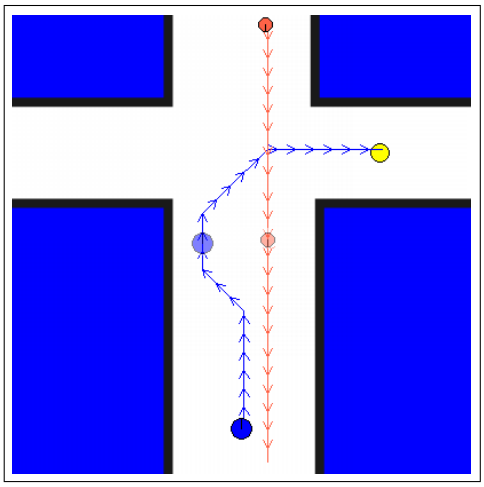


(a) Constant grid, with closest encounter marked

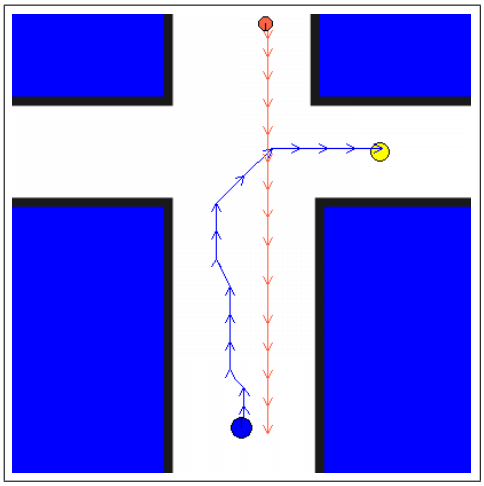


(b) Variable grid

Figure B.8: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the right of the hallway at a speed of 0.3 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. The robot turns in front of the person, but comes extremely close to the corner of the walls.

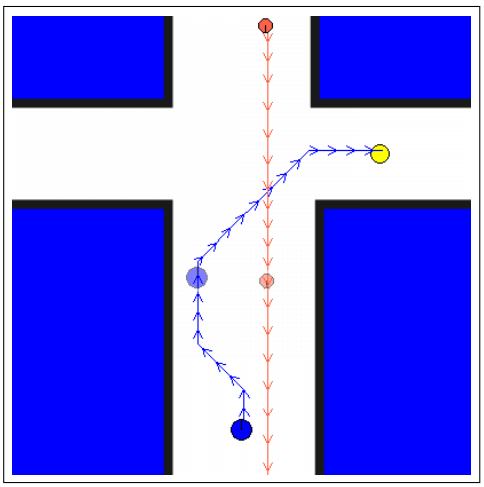


(a) Constant grid, with closest encounter marked

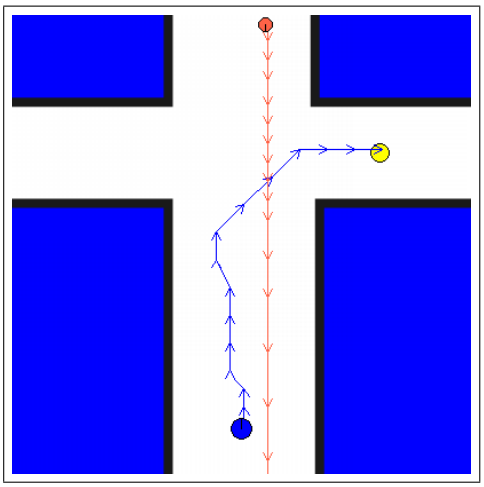


(b) Variable grid

Figure B.9: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the right of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. The robot moves to the left of the hallway rather than travel closely to both the person and the right wall.

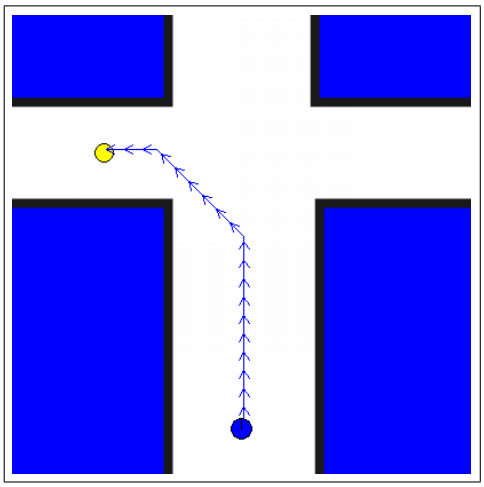


(a) Constant grid, with closest encounter marked

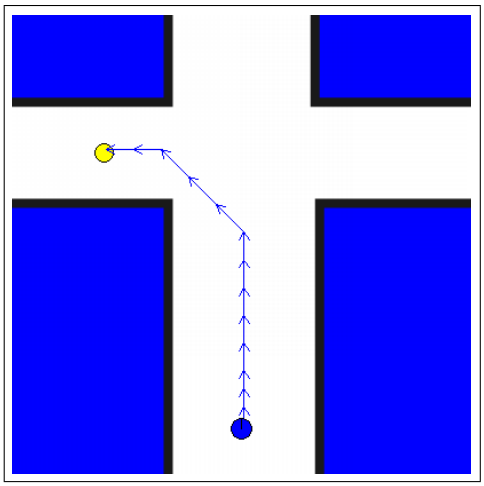


(b) Variable grid

Figure B.10: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's right. One person (orange circle) is traveling down the right of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. As with Figure B.9, the robot passes on the left, but moves out of the person's way sooner.

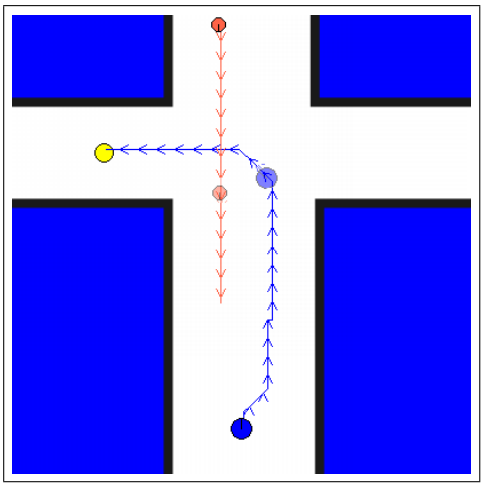


(a) Constant grid

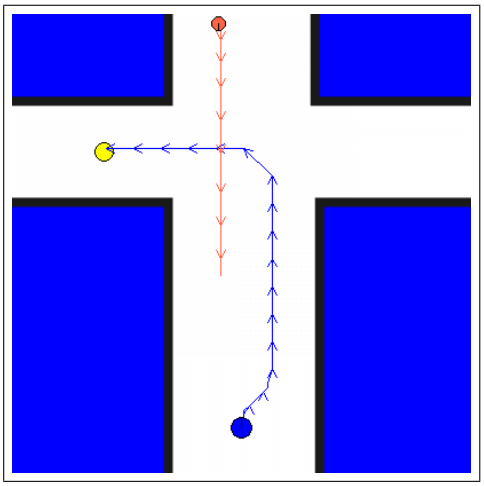


(b) Variable grid

Figure B.11: Path planned for a goal requiring the robot to turn left down a hallway.

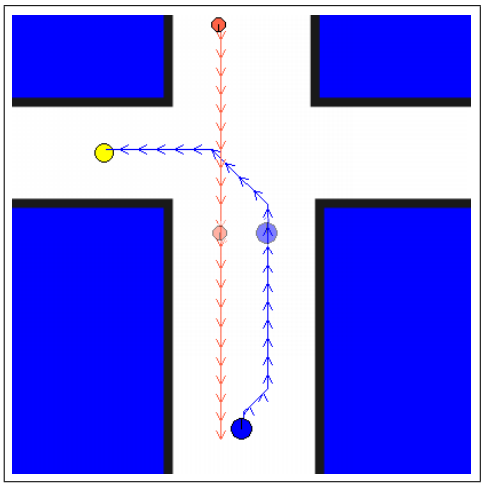


(a) Constant grid, with closest encounter marked

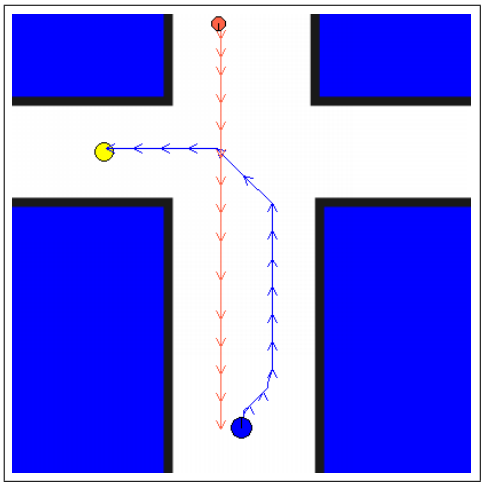


(b) Variable grid

Figure B.12: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the left of the hallway at a speed of 0.3 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

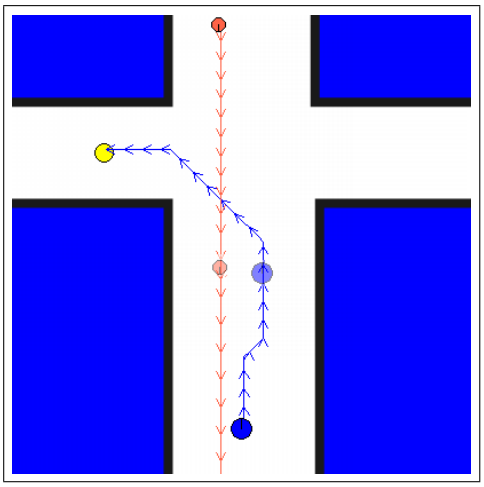


(a) Constant grid, with closest encounter marked

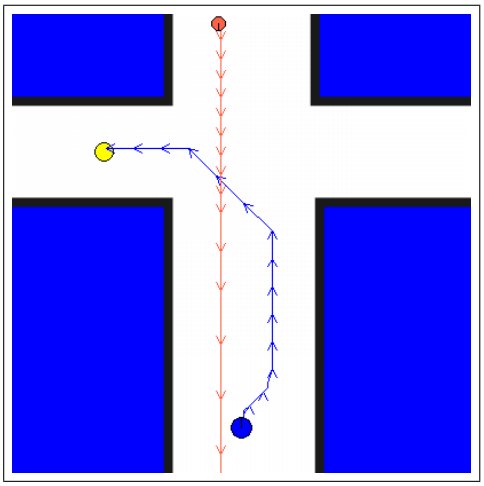


(b) Variable grid

Figure B.13: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the left of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

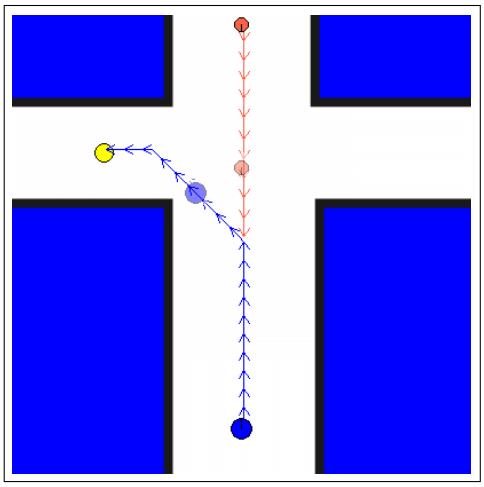


(a) Constant grid, with closest encounter marked

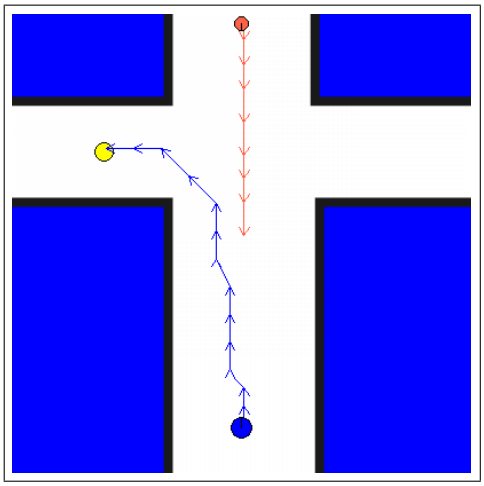


(b) Variable grid

Figure B.14: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the left of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

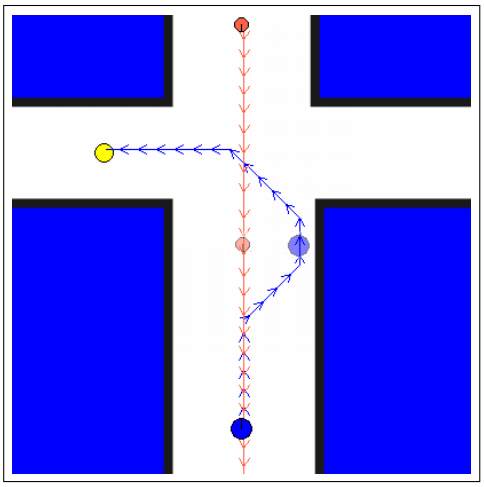


(a) Constant grid, with closest encounter marked

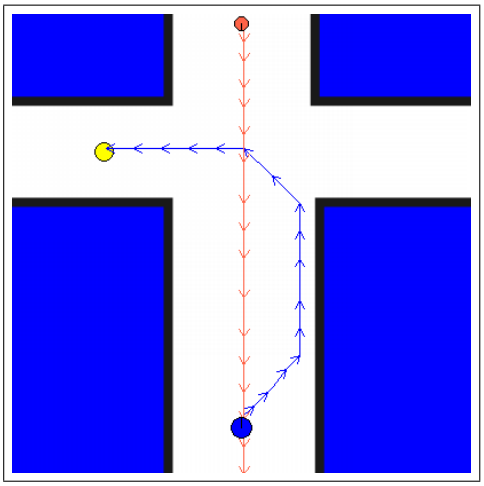


(b) Variable grid

Figure B.15: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the center of the hallway at a speed of 0.3 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. Because the person is moving slowly, the robot is able to cut across to the left of the hallway before they pass each other.

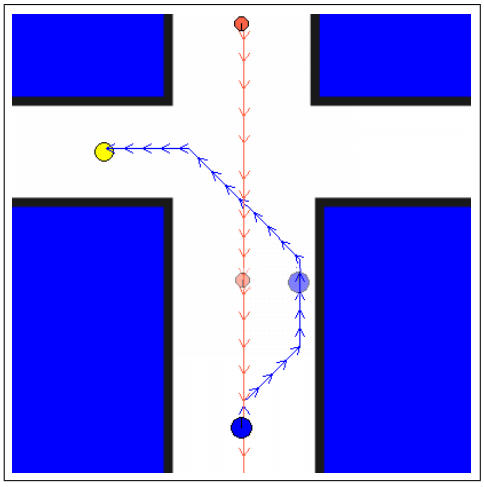


(a) Constant grid, with closest encounter marked

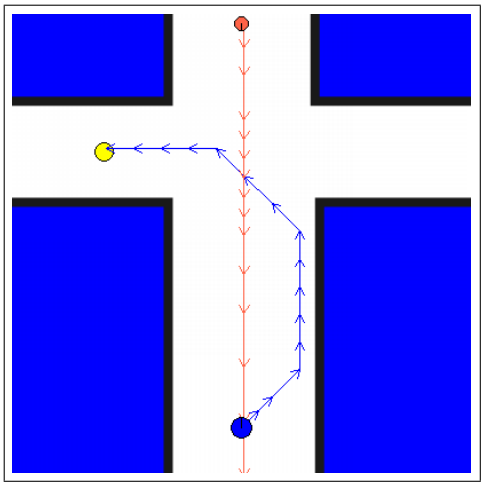


(b) Variable grid

Figure B.16: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the center of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. Because the person is moving faster than in Figure B.15, the robot instead takes a longer path on the right of the hallway.

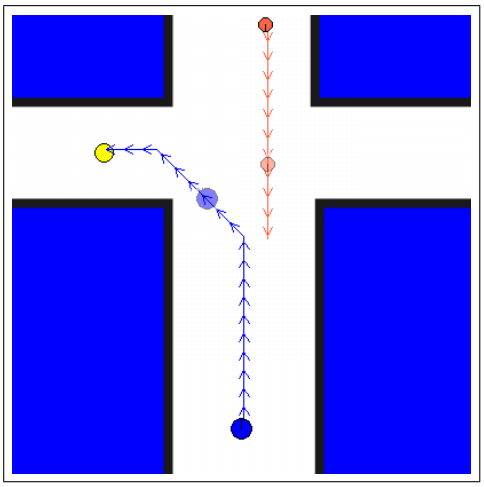


(a) Constant grid, with closest encounter marked

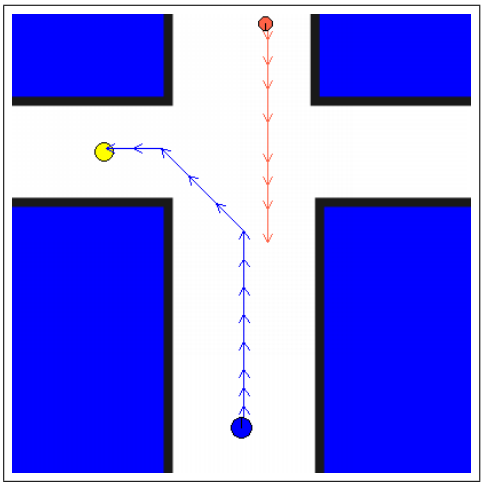


(b) Variable grid

Figure B.17: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the center of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. As with Figure B.16, the robot moves to the right to pass the fast-moving person.

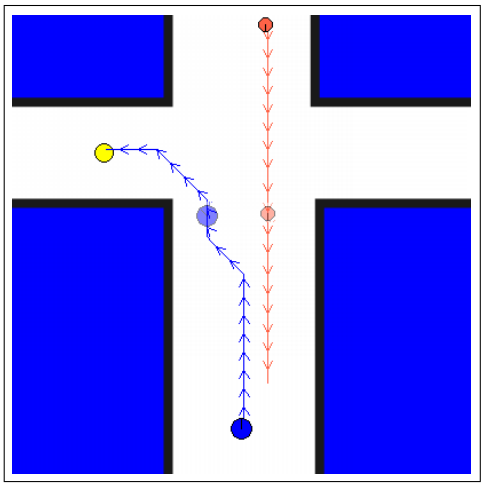


(a) Constant grid, with closest encounter marked

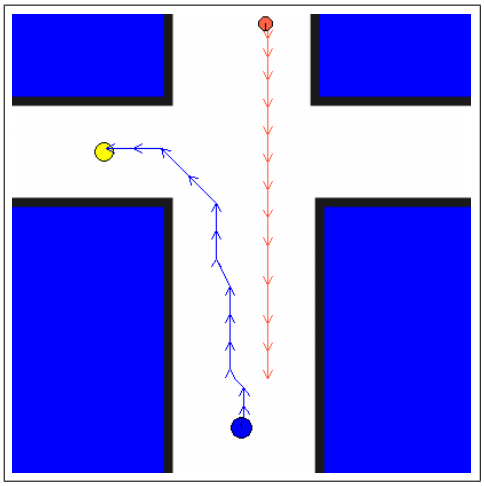


(b) Variable grid

Figure B.18: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the right of the hallway at a speed of 0.3 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

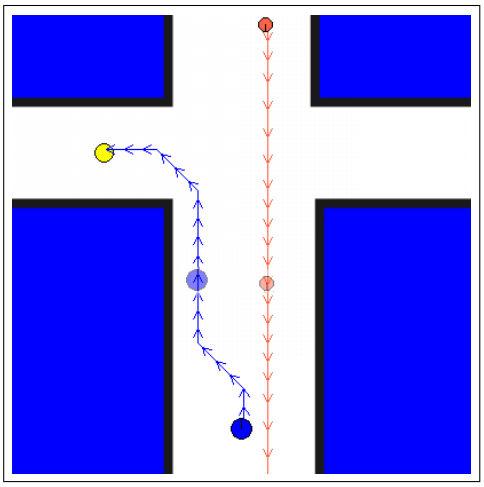


(a) Constant grid, with closest encounter marked

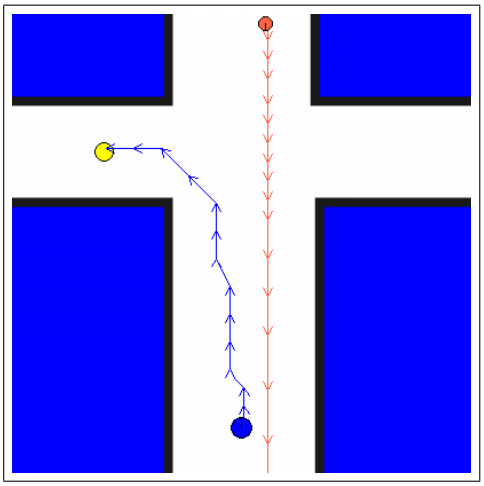


(b) Variable grid

Figure B.19: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the right of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. Unlike Figure B.16, the robot moves left rather than squeeze between the person and the wall on the right.

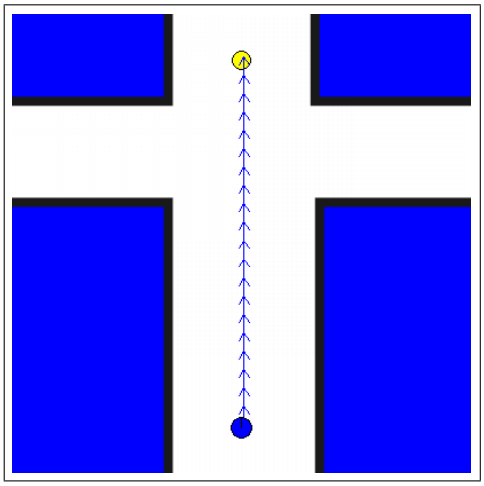


(a) Constant grid, with closest encounter marked

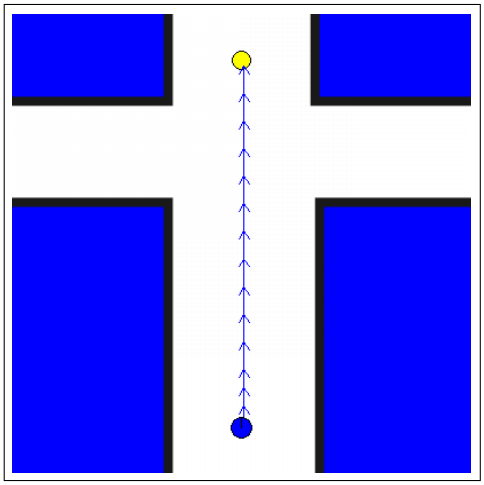


(b) Variable grid

Figure B.20: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) on the robot's left. One person (orange circle) is traveling down the right of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. Unlike Figure B.17, the robot moves left rather than squeeze between the person and the wall on the right.

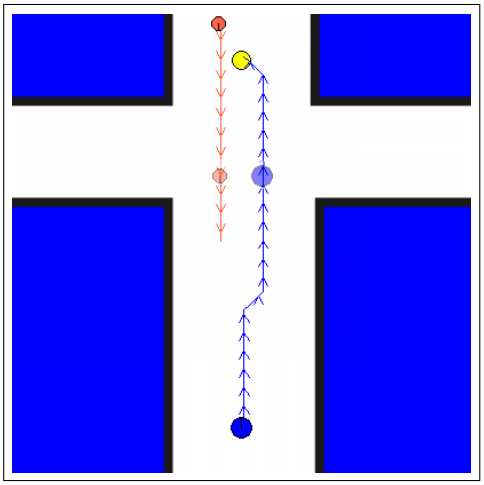


(a) Constant grid

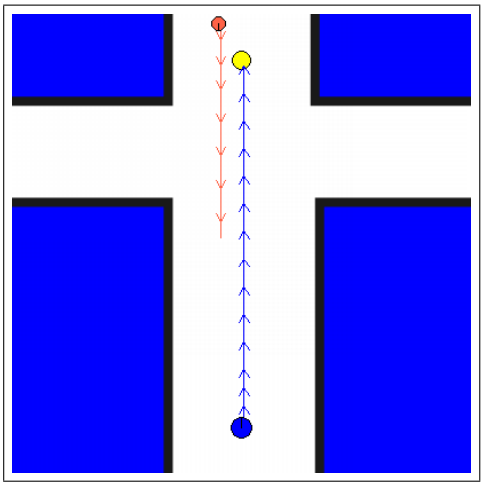


(b) Variable grid

Figure B.21: Path planned for a goal requiring the robot to drive straight past an intersection in the hallway.

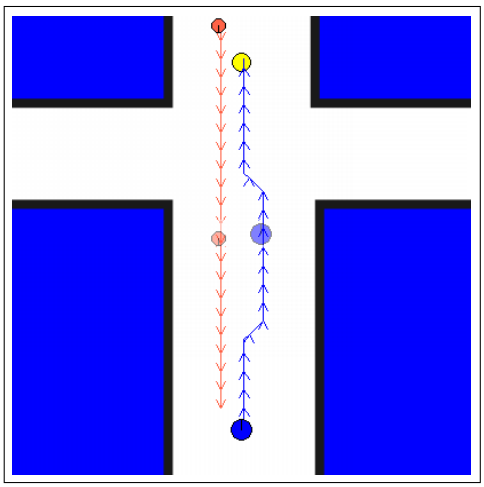


(a) Constant grid, with closest encounter marked

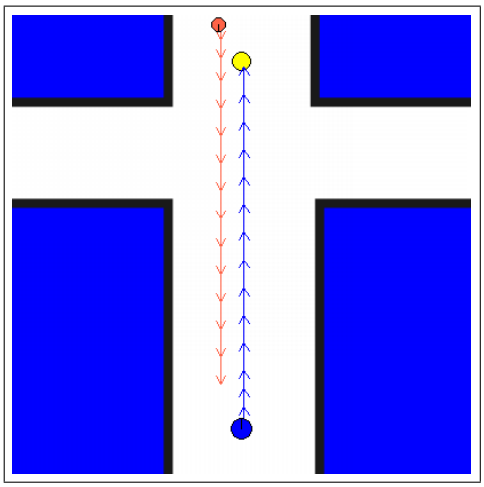


(b) Variable grid

Figure B.22: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the left of the hallway at a speed of 0.3 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. On the variable grid, the robot's path does not turn at all due to the size of the cells.

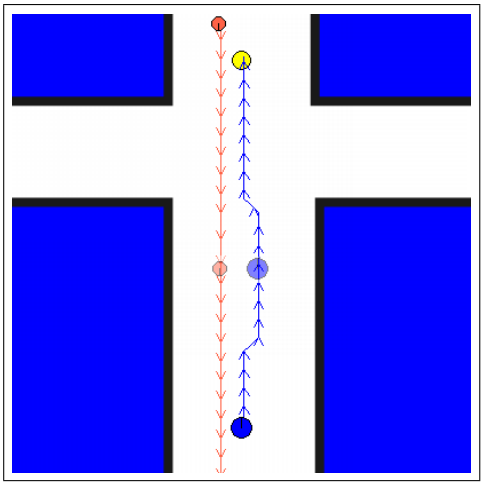


(a) Constant grid, with closest encounter marked

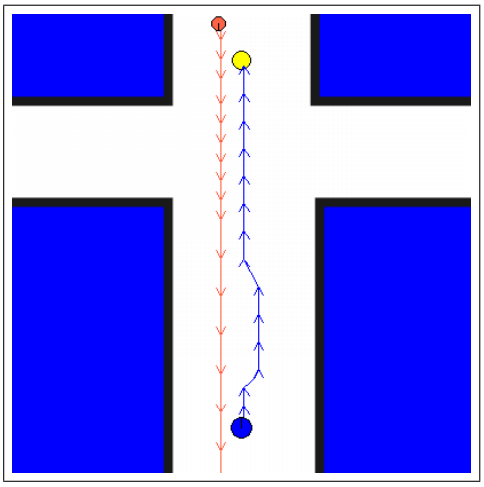


(b) Variable grid

Figure B.23: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the left of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. On the variable grid, the robot's path does not turn at all due to the size of the cells.

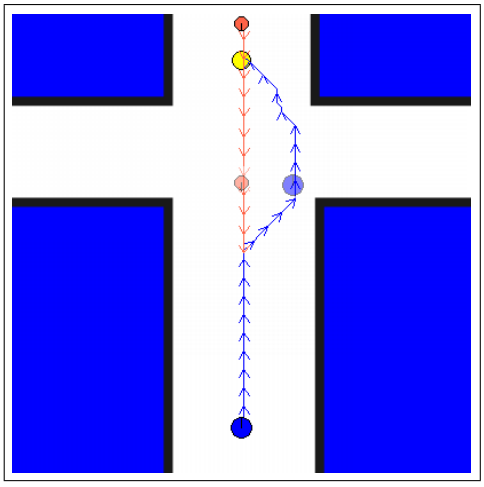


(a) Constant grid, with closest encounter marked

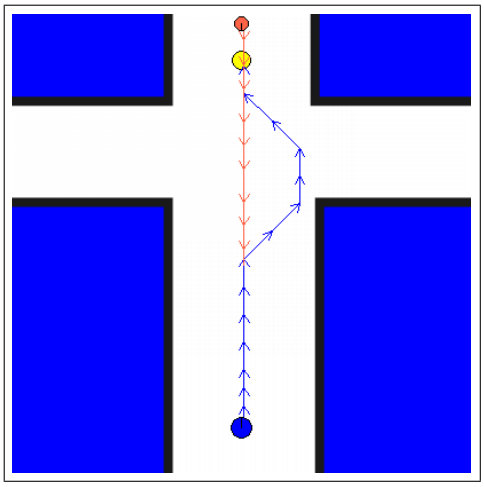


(b) Variable grid

Figure B.24: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the left of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

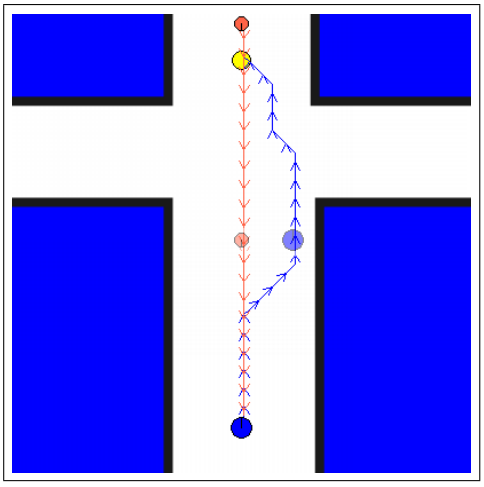


(a) Constant grid, with closest encounter marked

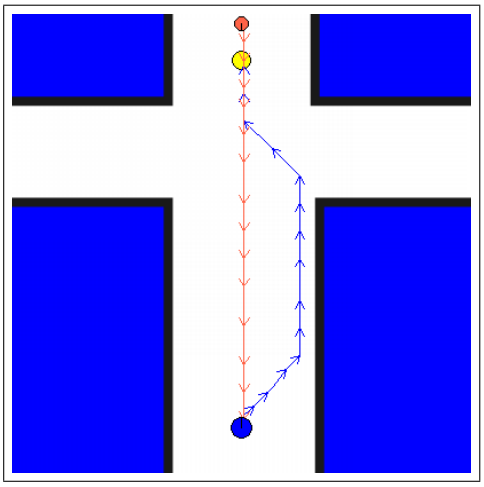


(b) Variable grid

Figure B.25: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the center of the hallway at a speed of 0.3 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

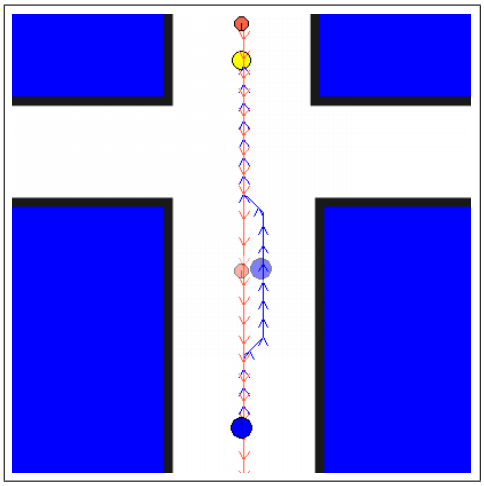


(a) Constant grid, with closest encounter marked

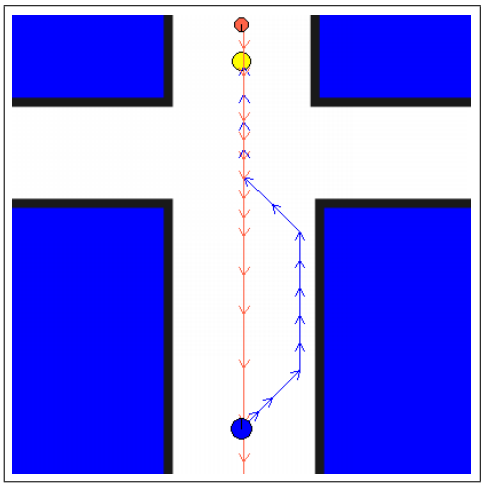


(b) Variable grid

Figure B.26: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the center of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid.

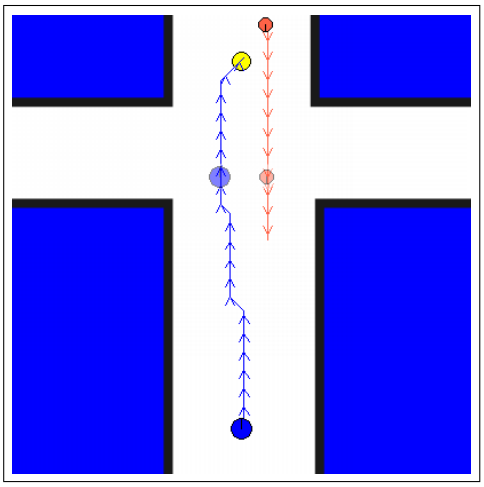


(a) Constant grid, with closest encounter marked

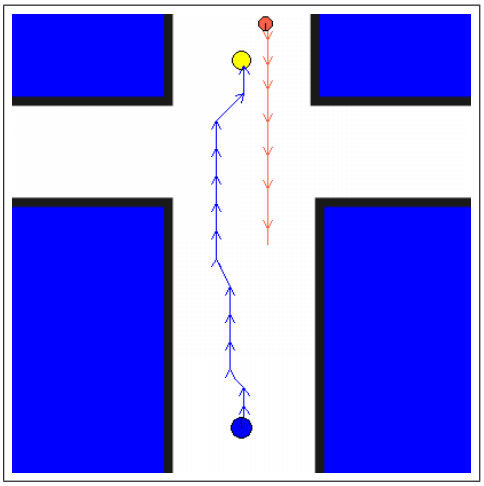


(b) Variable grid

Figure B.27: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the center of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. On the constant grid, the robot passes extremely close to the person; because the person is moving quickly, the robot trades off a briefly high cost from personal space with taking a short path. In contrast, on the variable grid with reduced action space, the robot must incur high inertia costs to avoid hitting the person, and thus also accepts the longer path rather than incur personal space costs.

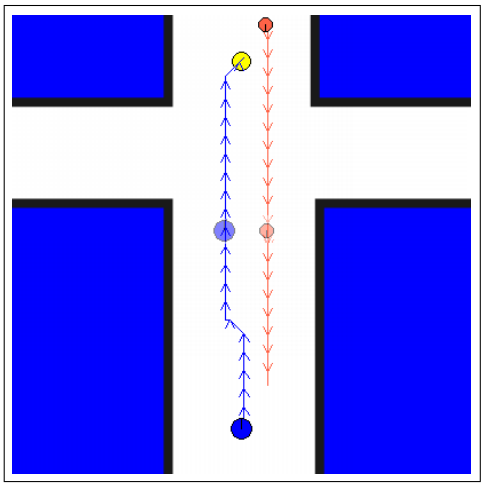


(a) Constant grid, with closest encounter marked

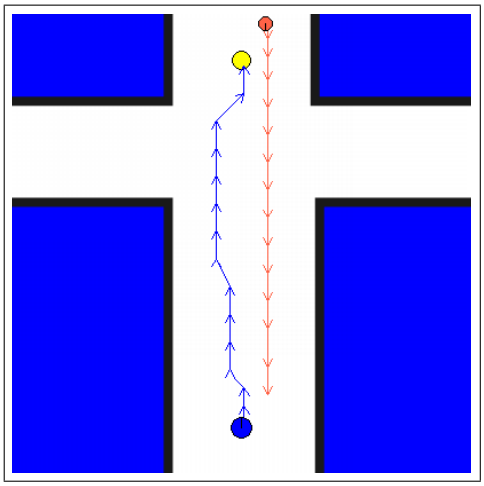


(b) Variable grid

Figure B.28: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the right of the hallway at a speed of 0.3 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. The robot moves left rather than travel close to both the person and the wall on the right.

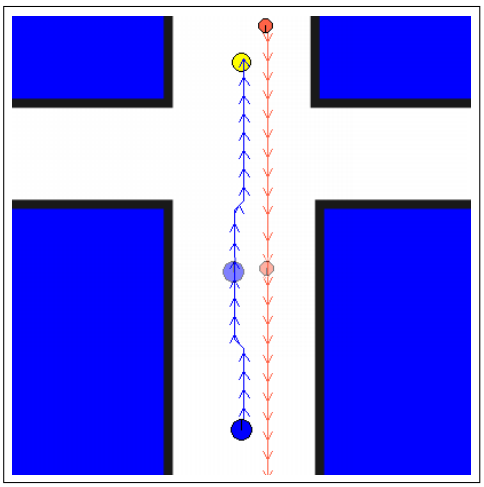


(a) Constant grid, with closest encounter marked

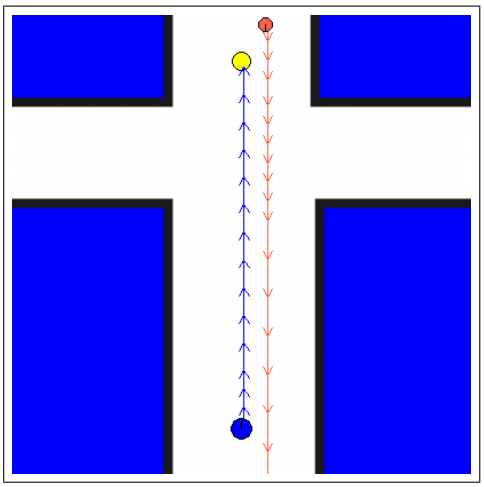


(b) Variable grid

Figure B.29: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the right of the hallway at a speed of 0.5 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. The robot moves left rather than travel close to both the person and the wall on the right.



(a) Constant grid, with closest encounter marked



(b) Variable grid

Figure B.30: Statically planned paths for the robot (blue circle at bottom) traveling at 0.5 m/s to a goal (yellow circle) straight ahead of the robot. One person (orange circle) is traveling down the right of the hallway at a speed of 0.7 m/s. Figure (a) depicts the the path planned on a constant grid, with the closest point between the robot and person marked. Figure (b) shows the whole path planned on a variable grid. As with Figure B.23, on the variable grid, the robot's path does not turn at all due to the size of the cells.

Appendix C

Cross-Cultural Social Differences

This thesis focuses on implementing human social conventions on a robot. In particular, it focuses on implementing the social conventions typically found in the United States, where personal space is large, people tend to walk on the right side of hallways, and cutting in front of others is often acceptable. While some mention has been made when describing the constraints to discuss necessary changes for use in other cultures (see Section 4.2), this appendix seeks to address, in a more general way, what kinds of social conventions differ across cultures. Focus is made on the differences between North American and Arabic social conventions primarily due to connections with the Carnegie Mellon University campus in Qatar.

C.1 Social Conventions

Interactions between people are governed by culturally-specific social conventions. These conventions regulate how close people stand near each other (termed "proxemics"), how they look at each other, how they gesture, and how they speak to each other.

C.1.1 Proxemics

The study of how people arrange themselves in space when interacting with each other was coined "proxemics" by Hall (1966). For North American cultures, Hall defined four proxemic zones: the intimate zone, which typically involves close contact; the personal zone, roughly an arm's length away, where people stand for face-to-face interaction; the social zone, which is further out and used for business interactions; and finally the public zone, which begins roughly 4 meters away from a person and is used for public speaking. Personal space, in particular, serves

as both protection for sensory information during an interaction (e.g., how much one sees and smells of the interaction partner), as well as communication between interactors, helping to define the type of interaction as well as the relationship between the participants (Aiello and Thompson, 1980).

For different cultures, these zones of interaction are often of different sizes. In particular, a distinction can be made between "contact" versus "non-contact" cultures (Hall, 1966; Watson, 1970; Aiello and Thompson, 1980). Non-contact cultures, which include North American, Asian, Indian, Northern and Western European, typically maintain interaction distances similar to those given above. In contrast, in contact cultures, such as Arabic, Latin, and Southern European, people typically interact at a much closer distance. Contact cultures tend to have a much greater tolerance for crowding in public spaces than do non-contact cultures; where North Americans may instinctively form lines in crowded shops, Arabs may push close in a large crowd (Feghali, 1997).

In many cultures, these interpersonal distances also relate to how well the interactors know each other. For example, in both Arab and North American cultures, people tend to keep strangers further away than friends (Sanders et al., 1985), and strangers sitting on public benches tend to keep similar distances across different cultures (Mazur, 1977). Gender can also influence interpersonal distances; some research has found that Arab females tend to keep male friends further away than female friends (Sanders et al., 1985).

If people of different cultures attempt to interact using their own proxemic behavior, "proxemic interference" may occur. This may lead to both discomfort and misunderstanding (Watson, 1970).

C.1.2 Gaze and Orientation

Different cultures not only maintain different interpersonal distances, but also tend to employ different body orientations. While North Americans tend to stand at angles to each other and look away while talking, Arab interactions involve direct body orientation and direct eye contact (Watson and Graves, 1966; Feghali, 1997). In such contact cultures, the American practice of avoiding direct eye contact is considered impolite (Watson, 1970).

C.1.3 Gestures

People of all cultures tend to gesture while talking. Some cultural-specific gestures, such as the American "OK" sign, can be offensive in other cultures (Safadi and Valentine, 1990). However, some types of gestures appear to be similar across

various cultures, such as head movements for each item in a list, and gestures (including head movements) for pointing (McClave et al., 2007).

C.1.4 Speech

An important part of speech is language. While English certainly has different dialects (consider American English versus British English), English speakers are generally able to understand each other. However, this is not necessarily the case with Arabic dialects, which have wide variability and speakers of one Arabic dialect may not be able to understand speakers of another (Feghali, 1997).

Beyond the language used, different cultures rely on different assumptions about another person's knowledge and expectations of a conversation. This is referred to as the context of the speech: high context relies on physical context (such as knowledge assumed to be internalized by the interactors), while low context is much more explicit in the spoken message. Arabic society is high context while Western societies are low context (Feghali, 1997).

Furthermore, the quality of speech, such as volume and rate, varies across cultures. In particular, Arabs tend to speak quickly and loudly as compared to North Americans (Watson and Graves, 1966; Feghali, 1997).

C.2 Implications for Robots

These differences in cultural conventions must be addressed when designing robots to interact with people of multiple cultures or even simply a culture other than the one in which the robot is designed. The following implications should be used as general guidelines for designing such robots. However, it is important to note that no conclusive evidence exists for the assumption that a robot should observe the same conventions as people; people may have very different expectations for robots, and a robot that attempts to behave in a human-like manner may not be ideal. Our work, however, does indicate that a robot that follows physical social conventions is easily understood by people.

A robot that speaks must be able to speak in the same language as the people to whom it is speaking. While this may seem obvious, the robot may need to adaptively change its primarily language in countries where many different dialects are spoken. A robot in an Arabic culture should speak faster and louder than a robot interacting with Westerners. Similarly, a robot that gestures or understands gestures must use culturally-specific models of gesture meaning, as similar gestures may have radically different meanings in different cultures.

Mobile robots need to account for different treatment of space in different cultures. In contact cultures, such as Arabic society, the robot may need to approach people more closely and interact with a more direct body orientation than it would in non-contact cultures. However, this may raise safety concerns; a robot that is capable of physically harming a person should perhaps keep a greater distance than would be typical for the culture.

Due to differing conceptualizations of public space, robots situated in contact cultures may need to have better handling of large groups. While people in non-contact cultures may instinctively line up to interact with a social robot, people in contact cultures may be more likely to attempt to interact with the robot as a large group (Feghali, 1997).

C.3 Conclusions

Developers of social robots need to be aware of the culture for which their robots are intended. As this appendix has discussed, social conventions vary across cultures, and behaviors that are proper in one culture may be awkward or impolite in others.

However, little research has been done to date regarding social robots across cultures. One existing computational model simulates difference in proxemics and gaze for virtual agents to interact as Anglo American, Spanish, or Arabic (Jan et al., 2007), and may be a reasonable starting point for a more complete model for social robots.