PEOPLE AWARE MOBILE ROBOT NAVIGATION

A Thesis
Presented to
The Academic Faculty

by

Akansel Cosgun

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
College of Computing

Georgia Institute of Technology
August 2010

PEOPLE AWARE MOBILE ROBOT NAVIGATION

Approved by:

Professor Ignatius Arrogant,
Committee Chair
College of Computing
Georgia Institute of Technology

Professor Henrik Christensen, Advisor
College of Computing
Georgia Institute of Technology

Professor General Reference
School of Mathematics
Georgia Institute of Technology

Professor Ivory Insular
Department of Computer Science and Operations Research
North Dakota State University

Professor Earl Grey
College of Computing
Georgia Institute of Technology

Professor John Smith
College of Computing
Georgia Institute of Technology

Professor Jane Doe
Another Department With a Long Name
Another Institution

Date Approved: 1 July 2010

To myself,

Perry H. Disdainful,

the only person worthy of my company.

PREFACE

Theses have elements. Isn't that nice?

ACKNOWLEDGEMENTS

I want to thank people

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Why should I provide a summary? Just read the thesis.

CHAPTER I

INTRODUCTION

Introduction

1.1 Background

1.1.1 Social Spaces

According to Lam [43], mobile robots should obey certain rules while navigating in human environments. These rules include: not colliding anybody, not entering the personal space of a human unless the task is to approach the human and waiting if robot unwillingly enters the personal space of a human. Humans are already good at obeying such social conventions. Therefore most works on robot navigation in human environments is linked to human-human spatial interactions. One of the first studies in such interactions is conducted by Hall [25]. This study presents the proxemics theory, which categorizes the distance between people in four classes. These distances, named intimate, personal, social and public, provide spatial limits to different types of interactions. Kendon [31]'s F-formation is based upon observations that people often group themselves in a spatial formation, e.g. in clusters, lines and circles. Some works adopted Hall distances and Kendon's formations for human-robot interaction. Huttenrauch [29] found that personal distance between a robot and a person varied in the range of 0.45 to 1.2 meters and but claimed that works of Hall and Kendon should be adapted to suit the dynamics of HRI. Avrunin [3] aims to learn acceptable distances from human-human experiments in an approaching scenario.

CHAPTER II

MAP ANNOTATION

Map Annotation

In mobile robotics, the standard practice for mapping and localization is described as follows: When the robot is first taken to a new environment, it has to map the environment. There has been extensive research on Simultaneous Mapping and Localization (SLAM) literature. The usual output is a binary 2D grid map where 1s represents an obstacle and a 0s represent free space. Once the map is created, the robot can localize itself in the map while in operation. Every time the robot is restarted, it has to start with an initial estimation of its location. Although there are global localization methods developed in the community, the usual practice is that the robotics expert manually provides an approximate initial location of the robot, then the localization method corrects the localization estimation as the robot moves in the environment.

2.1 Related Work

Related Work

2.2 Semantic Maps

Semantic Maps

2.2.1 Waypoints

2.2.2 Planar Landmarks

2.2.3 Objects

2.3 User Interface

User Interface

2.4 Pointing Gestures for Human-Robot Interaction

Pointing Gestures

CHAPTER III

NAVIGATION AMONG PEOPLE

Autonomous navigation is one of the most fundamental tasks for a mobile robot. For a mobile robot with adequate actuation and sensing, collision-free navigation is considered a solved problem. There are many algorithms that achieve point-to-point autonomous navigation thanks to the advances in the motion planning community. Many of these algorithms are optimized to find the least-cost path, or the shortest path. However, when there are humans in the environment, such algorithms suddenly become inefficient or insufficient. For example, while it is acceptable for a robot to get inches close to a wall, doing so to a human is socially unacceptable and potentially dangerous. Similarly, sudden appearance of a robot can surprise or shock humans. There are many other social scenarios where the shortest path may not be optimal.

In addition to sub-optimality, these approaches may be incomplete in the sense that they can not find a solution even though there is a feasible one. This is because shortest-path navigation algorithms treat every object in the environment as an obstacle. This assumption does not hold when intelligent agents are present in the environment. Therefore navigation should differentiate humans and obstacles for more intelligent robot behavior.

Another aspect to spatial interaction between humans and robots is the dynamics of the robot motion. For example, people may feel uncomfortable and unsafe when they are in close proximity to high-speed agents or objects. Therefore, for a robot in a human environment, while it may be acceptable to speed up in dedicated regions, its speed should be limited in places where there is a significant possibility of encountering a human.

In this Chapter, we first provide a background and present the most common approach in contemporary autonomous navigation methods in Section 3.1. Second, we provide provide relevant works on navigation among people in Section 3.2. Third, in Section 3.3, we present how the goal points for navigation are determined. We then present our people-aware navigation method in Section 3.4. Lastly, we touch to the subject of introducing speed limits for all robots in a human environment in Section 3.5.

3.1 State-of-the-Art Approach in Autonomous Navigation

There are two prerequisites that enables autonomous navigation:

1. The map of the environment, usually in the form of a discrete grid, that represents static objects in the environment

2. A way to localize the robot in the map using sensory information as it moves in the environment

Robot navigation involves finding a collision-free path from a start pose (x_0, y_0, θ_0) to a goal pose (x_g, y_g, θ_g). In real-time operation, (x_0, y_0, θ_0) is the robot's current pose as the robot tries to reach to the goal pose from where it currently is. θ_g is optional as the goal of the robot could be to reach the goal position regardless of its orientation. The goal position is provided from an external process, and we will touch upon how the goal positions are calculated in Section 3.3.

A common approach to path planning is to divide the path planning into two parts: *global* and *local*. Global planning aims to find a path from the start position to the goal position. The global path is a set of consecutive positions that connect the start to goal position. A global path is usually found with a search algorithm executed on a graph of points. The search heuristics is dependent on specific global planners, and in most cases collision-free shorter paths are favored. The local planner is responsible

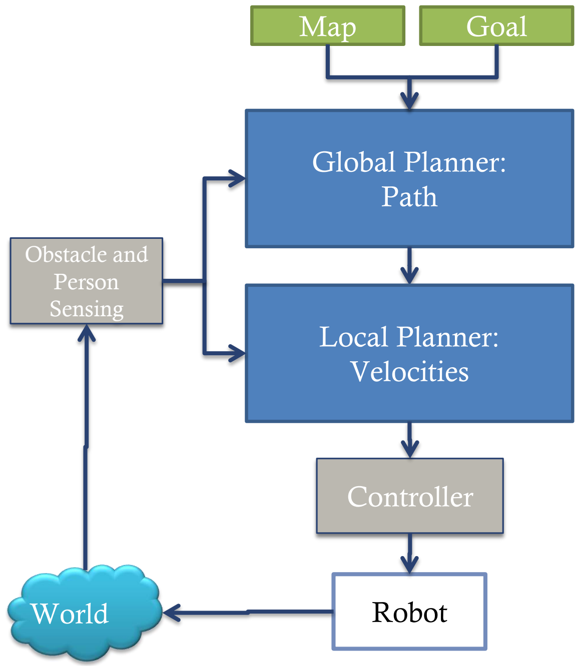


Figure 1: Caption

to execute the global path by calculating a trajectory and sending velocity commands to motor controllers. As the robot acts in the environment, its sensors sense the new state of the robot and people, and the new iteration begins. This cycle is shown in Figure 1.

A popular method to to implement the global and local planners is by using a *costmap* . A *costmap* not only has the same representation as the map, however collision-free positions can have non-zero costs. A lower cost cell is more favored to be in to a higher cost cell. After the calculation of all cells, the least-cost path is found that connects the start position to the goal position.

Note that this approach assumes the robot is able to execute any path provided to it. Holonomic robots can move in any direction, however non-holonomic robots has limitations in their movements. For example, two wheel robots can not move sideways. Two common approaches to solve this problem are: to implement trajectory planners that can handle imperfect control or to embed the robot's dynamics into sampling for global and local planning.

3.2 Related Work

In this section, we review the literature on robot navigation in human environments including socially acceptable navigation, learning behaviors from humans and cooperative navigation.

3.2.1 Socially Acceptable Path Planning

Socially acceptable robot navigation is considered in different applications such as free navigation [83], approaching people [75] and evacuating buildings [61]. Some works used the personal space concept in cost-based general path planners [83, 36]. Sisbot [83] models the social spaces as a ellipse-shaped Gaussian, and takes into account the safety, preferences and vision fields of humans for a robot that navigates from a location to another. Kirby [36] presents a path planner that takes into account social conventions such as tending to one side of the hallways. A potential field based trajectory planner for dynamic human environments is presented by Svenstrup [87]. Rios-Martinez [71] presents a RRT-based planner that considers not just safety but also the disturbance of humans. In simulation, if interaction within a group of people is detected, the robot can either not disturb the interaction or join the group. This approach is implemented on a wheelchair robot [94]. Althaus [1] presents a robot that can join a group of people and adjust to the formation reactively. The scenario where a robot encounters a human in a hallway is studied by Pacchierotti [65]. Parameters such as the distance between the human and the robot when the robot begins to deviate from its path and lateral distance that robot should be placed when it is passing the human are found from experiments. Recent work by Lu [48] showed that using gaze cues and social navigation makes robot-human hallway passing more efficient.

3.2.2 Learning Navigation from Human Behavior

Behaving human-like in robot navigation is usually favored in the literature [74]. One way to simulate human navigation behavior is to use social cost maps that capture social conventions [76, 49]. Contrary to the imitation approach, [7] tries to avoid predicted paths, with the goal to minimize the risk of interference. Kuderer [42] presents a tele-operated robot that computes the policy of a desired interactive navigation by learning from observations of pedestrians. Pellegrini [68] trains a dynamic social behavior, that account for social interactions, using pedestrian data.

3.2.3 Human Cooperation in Robot Navigation

Robots can exploit human cooperation in certain scenarios. In populated environments, one way to move with the crowd is to follow individuals that move towards the robot's goal [85, 57].

Some of the recent works in the literature claim that the robot motions should be predictable so the human observers can judge the motive and future behavior of the robot. Observational study in [46] claims that three features can increase the predictability of robot navigation: straight lines, stereotypical motions and usage of additional gestures. In a user study conducted by Gockley [23], humans observers watched two ways of person following. People found direction-following more natural than exact path following. Kruse [40] observes that when paths of two humans are crossed at a right angle, they adapt their velocity rather than the path. This behavior is implemented on a robot, resulting in more predictable motions.

Trautman [91] introduces the 'freezing problem', where traditional path planners fail to produce a feasible solution in crowded human environments. Muller [57] briefly mentions a 'shooing away' behavior, where the robot accelerates towards a human, hoping that he/she will get out of the way. Kruse [41] introduces an optimistic planner, which assumes that people will cooperate with robot movements. Their approach relies on assigning a non-infinite cost if a robot enters to a human's personal space, however the plan fails if humans doesn't move as expected because of the lack a local planner.

3.3 Goal Points for Navigation

As presented in Section TODO, our interactive system allows a user to annotate landmarks. After completing the *HomeTour*, the robot can navigate to or towards the labeled entities. A user can enter a navigation destination to the robot in three distinct ways: via labeled waypoints, planar landmarks or objects.

3.3.1 Labeled Waypoints:

If a waypoint is labeled and saved, the robot attaches that label to the explicit coordinates, namely position and orientation. Therefore, if the robot is instructed to navigate to a labeled waypoint, then the goal is readily the pose of the waypoint.

3.3.2 Labeled Planar Landmarks:

If the label is attached to planar landmark, or a set of planar landmarks, we use the following methodology depending on the number of landmarks attached to the corresponding label:

3.3.2.1 Only a single plane has the corresponding label:

We assume that the robot should navigate to the closest edge of the plane, so we select the closest vertex on the landmark's boundary to the robot's current position. This point is projected down to the ground plane, as our robot navigates on the floor. We calculate a line between this point and the robot's current pose, and navigate to a point on this line a meter away from the point, and facing this point. This results in the robot navigating to near the desired landmark, and facing it. This method is suitable for both horizontal planes such as tables, or vertical planes such as doors.

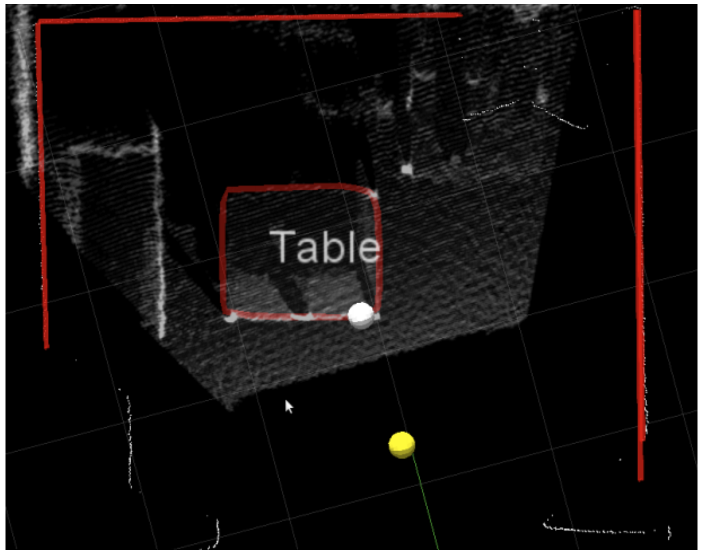


Figure 2: Top down point cloud view of a room. A planar landmark with label *Table* has previously been annotated by a user. The convex hull for the planar landmark is shown in red lines. When asked to navigate to *Table*, the robot calculates a goal pose, which is shown as the yellow point.

An example for calculating a goal for a single labeled planar landmark is shown in Figure 2.

where a the goal point corresponding to the singular label *Table*.

3.3.2.2 Multiple planes are attached to the same label:

We assume that the requested label corresponds to a region of space such as a room or corridor. In this case, we project the points of all planes with this label to the ground plane, and compute the convex hull. For the purposes of navigating to this label, we simply navigate to the centroid of the hull. While navigating to a labeled region is a simple task, this labeled region could also be helpful in the context of more complex tasks, such as specifying a finite region of the map for an object search task. An example for calculating a goal for a two labeled planar landmarks is shown in Figure 3.

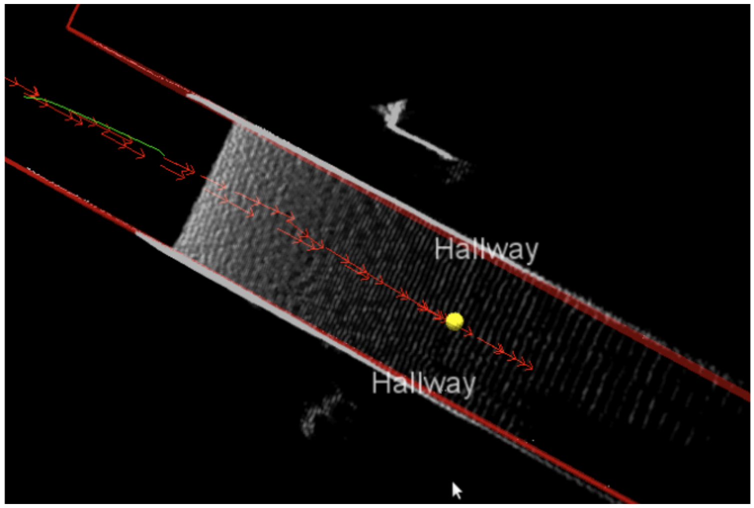


Figure 3: Top down point cloud view of a hallway. The user has previously annotated two planar landmarks with the same label, *Hallway*. When asked to navigate to *Hallway*, the robot calculates a goal pose, which is shown as the yellow point.

3.3.3 Labeled Objects:

As discussed in Section TODO, we first perform planar surface detection before detecting tabletop objects. When the robot is asked to navigate to a labeled object, the planar surface that the object lies on is given as the goal landmark. The robot calculates the goal position as described in the single labeled landmark case in Section 3.3.2.1.

3.4 People Aware Navigation

A extensively reviewed in Section 3.2, people-aware navigation algorithms aim to generate human-friendly paths that consider the safety and comfort of people. A common assumption for point-to-point people aware navigation is that humans are independent agents and that robot's motions have no effect on people's motions. However, humans navigate by constantly anticipating other people's reactions. Similarly, mere presence of a robot in motion is likely to influence how nearby humans would move.

Robots can potentially use this implicit cooperation between moving embodied agents. For example, consider a robot that is outside a room and given a goal pose

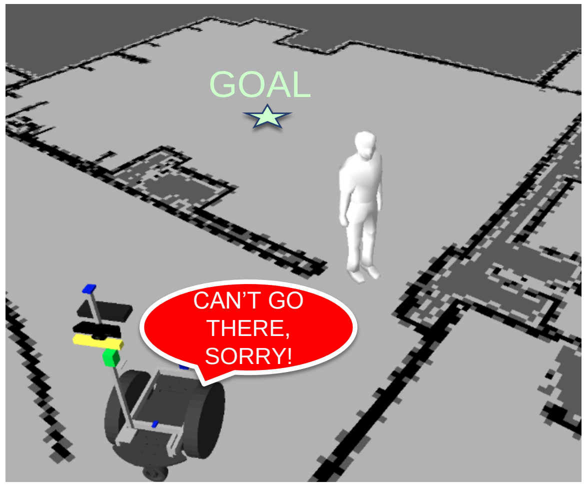


Figure 4: Standard path planners fail to produce a solution to the 'room problem'. Our people-aware planner anticipates that the human can give way to the robot if it approaches towards its goal.

in the room. There is a person standing at the door and blocking the path. Such an example is illustrated in Figure 4. Standard path planners, as well as planners that consider dynamic objects fail to produce a solution to this problem. The role of physical embodiment in human-robot interaction is significant [96], however it is commonly ignored in robot navigation. A people-aware planner should anticipate that the human may give way to the robot if it expresses its intent to go inside the room. Extending this idea, by using anticipation a robot can reduce its time of travel and behave more human-like in general cases.

In this section, we propose a people-aware navigation planner that considers reactions of humans to robot motion. Our planner first first finds the least-cost map in the costmap that considers safety and disturbance of people. The costmap definition is discussed in Section 3.4.1. Then the path is refined by simulating people's reaction to robot's motion using Social Forces Model [26]. The path refinement will be discussed in Section 3.4.1.1. In dynamic simulation, robots and humans repulse each other, and additional forces helps to stay away from obstacles and conserve formation in groups. Paths are re-planned when the world state changes or humans does not move as anticipated. In Section 3.4.2, we discuss our local planner. We then discuss the implementation of the system in Section 3.4.3, demonstrate two example scenarios in simulation in Section 3.4.3.1 and two on the real system in Section 3.4.3.2.

3.4.1 Global Planner

The global planner takes the start and goal positions and a 2D grid map as input and aims to find a set of waypoints that connects the start and goal cells. The output path has the minimum cost with regards to a cost function with 3 parameters: path length, safety and disturbance. We use A* search with Euclidean heuristics on a 8-connected grid map to find the minimum cost path. The configuration space obstacles are found by inflating the map obstacles for as much as the radius of the robot with the assumption that the robot is circular.

Path length cost: Each action a of the robot (moving to one of the 8 adjacent cells) has a non-negative action cost $Cost_a(x_i, y_i, a)$. If the destination cell is occupied by a configuration space obstacle, then the action cost is infinite. Otherwise, it is the distance in meters. The action cost is thus defined as:

$$Cost_a(x_i, y_i, a) = \begin{cases} u & \text{if } a = \text{N, E, S, W} \\ u\sqrt{2} & \text{if } a = \text{NW, NE, SW, SE} \\ \infty & \text{if } Cell(x_{i+1}, y_{i+1}) \text{ in obstacle} \end{cases} \tag{1}$$

where N,NW,.. are the grid cell expansion directions and u is the grid cell size. The resulting path length cost of a path P is then the sum of all action costs:

$$Cost_{path}(P) = \sum_{a \in P} Cost_a(x_i, y_i, a) \tag{2}$$

Safety cost: The notion of safety is the absolute need of any human-robot interaction scenario. This cost is a human centered 2D Gaussian form of cost distribution and aims to keep a distance between the robot and the humans in the environment. While some approaches used un-isotropic cost functions to account for human orientation, we use a isotropic Gaussian for its simplicity. Each cell coordinate around a human contains a cost inversely proportional to the distance. Since the safety loses its importance when the robot is sufficiently far away from the human, safety cost becomes zero after a threshold distance. If there are multiple people in an environment, the safety cost of a cell takes its value from the closest human.

$$Cost_{safety}(x, y) = \begin{cases} u \max_{h \in H}(\mathcal{N}(\mu_h, \sum)) & \text{if } d < d_{max} \\ 0 & \text{if } d \geq d_{max} \end{cases} \tag{3}$$

where d is the distance to the closest human, H is all humans, $\mu_h = (|x - h.x|, |y - h.y|)$ and $\sum = 0.5I_2$ is a fixed covariance matrix. The multiplication by the grid cell size compensates for the grid map resolution. Otherwise, for example, if a very fine map was used, safety cost would dominate the path length and disturbance costs, which are independent of the map resolution.

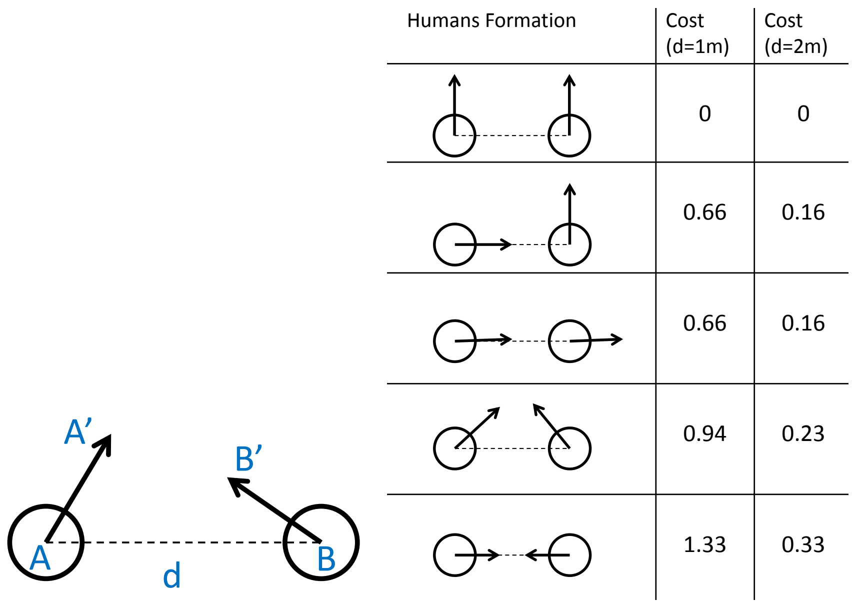
Disturbance cost: This cost is aimed to represent the cases where the robot potentially disturbs the interaction of a group of humans. For example, if two people are facing each other and talking, then the robot should not cross between them. We model this with a disturbance cost that is introduced if a path crosses between two people. We do not detect if there actually is conversation between the people, but estimate the disturbance cost using body poses of agents. This cost increases if body orientations of two people are facing each other and is inversely proportional on the distance between the two humans.

For each step taken in the grid, we check if the line segment from the current position to the projected position intersects a line segment between all pairs of humans. To illustrate, let's assume the robot crosses the line between human A and human B in Figure 5(a).

The disturbance cost is calculated as:

$$\begin{aligned} Cost_{dist}(x, y, a) &= \max(0, f(d).(\vec{AA'}.\vec{AB} + \vec{BB'}.\vec{BA})) \\ f(d) &= \frac{1}{d} - \frac{1}{d_{max}} \end{aligned} \tag{4}$$

where all the vectors are normalized and d_{max} is the maximum distance between the humans that returns a disturbance cost. Figure 5(b) illustrates several examples of disturbance costs with $d_{max} = 3$ meters.



(a) (b)

Figure 5: Disturbance costs in different human-human configurations and distances. A path that crosses the dashed lines incurs the disturbance cost calculated on the right side.

Total Cost: The total cost of a path P is computed with a weighted average of path length, safety and disturbance costs. We use A* search to find the least-cost path.

$$Cost_{Total}(P) = Cost_{path} + w_s.Cost_{safety} + w_d.Cost_{dist} \tag{5}$$

3.4.1.1 Path Refinement using Social Forces

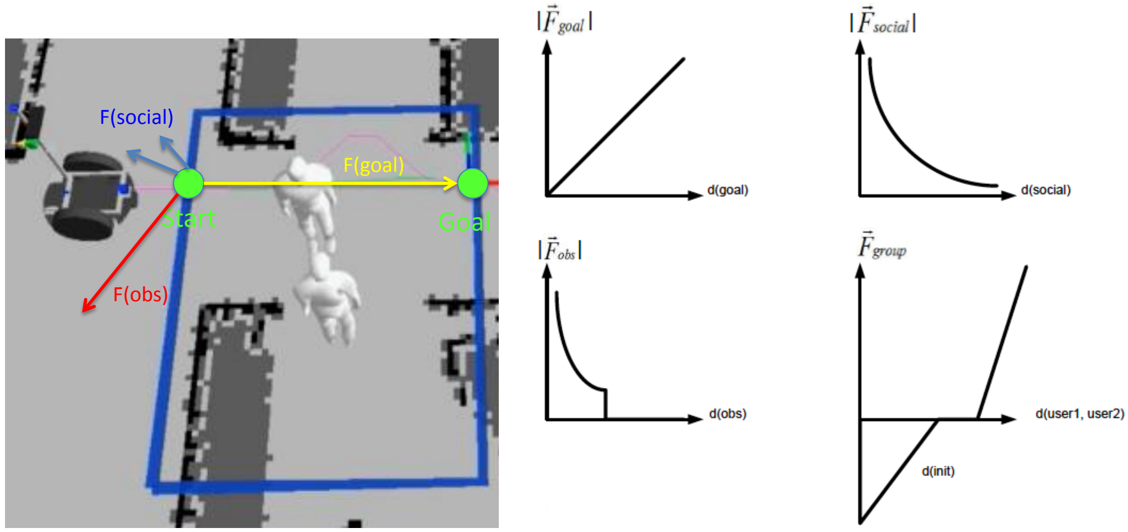
In this section, we describe the path refinement process that is applied to the global path. The initial geometric path generated by the global planner is not smooth, therefore robot motion might not be easy to interpret for human observers. The path refinement processes the global plan and simulates the parts of the path where group of humans are closeby. We use Social Forces Model (SFM) [26] to simulate the motions of both humans and the robot. Interaction between people are modeled as attractive and repulsive forces in SFM, similar to the Potential Field Method [33] for robot navigation. The forces are recomputed iteratively and the resulting simulated paths replaces the corresponding path sections in the global plan.

First, groups of people are found by clustering with respect to their positions. Simple euclidean distance thresholding is used for clustering. In our current implementation, a group region is defined as a rectangle, although other shapes are also possible. The path refinement process receives the global plan and finds out where it enters and exits each group region if it intersects the region. Goal of the dynamic iterative simulation is to find a sub-plan between those two points. Forces apply to all agents, including the robot and humans. We define 4 forces acting on the agents:

- F_{goal} : attraction towards a sub-goal
- F_{social} : repulsion from other agents
- F_{obs} : repulsion from nearest obstacle
- F_{group} : attraction or repulsion towards group members

The forces acting on the robot at the first iteration of forces simulation are illustrated on the robot in Figure 6(a). The force magnitudes with respect to distances between entities are plotted in Figure 6(b).

Starting from the first group region that intersects the static plan, the following procedure is applied within every group region: At every iteration, first the resultant force vector acting on the robot is found. Then the planner takes a step in the direction of the F vector for a fixed step size. Then each of the humans in the group takes a step towards the resultant force that is acting on them. The planner continues the iterations until a solution is found. If a solution is found, the calculated sub-plan replaces the static plan in this group region. Potential fields are known to stuck to



(a) (b)

Figure 6: a) Social forces acting on the robot, including F_{goal},F_{social},F_{obs}, are shown at the first iteration of the dynamic planner. Note that $F_{group} = 0$ as the robot does not belong to a group. The group force (not shown) exists, however, for the humans as they are in the same group region. b) Social forces with respect to the distance towards the corresponding entity.

local minima [39], and the planner might go into infinite loop. We stop the planner after a number of iterations and accept the static plan in the corresponding group region if that happens.

We assume that humans have a cognitive model of the robot, by thinking that the robot has a limited Field of View (FOV). When the robot has gone past a human (out of the FOV), then we make the repulsion force $F_{social} = 0$. We think that humans behave that way: as someone walks past, there are no social constraints resulting from that individual any more.

3.4.2 Local Planner

The local planner is responsible for finding the trajectory that the robot is capable of executing. It accepts a geometric global path as input and computes the linear and angular velocity necessary to follow the dynamic path. We adopt a local planner inspired by Dynamic Window Approach (DWA) by Fox [17]. In the original DWA approach, only circular trajectories are considered, defined by pairs (v, w) of linear and angular velocities. An objective function, consisting of target heading, clearance from obstacles and velocity of the robot is maximized by sampling admissible velocities.

Our approach also samples admissible velocities, but the optimization criteria we use consists only of the Euclidean distance to a sub-goal point chosen on the path that is ahead of the robot. The velocity pair that resulted in the closest proximity to the sub-goal is chosen and sent to robot controllers. At every control iteration, the sub-goal is chosen as the first point ahead of the robot that is further than a distance threshold. We found that a threshold of 0.25 meters was sufficient to choose the sub-goal. After the local planner calculates the output velocities, they are applied to the robot and the iterative process continues until the the robot reaches the goal. Since the goal is a singular point, it is impossible for the robot to be exactly at the goal. Therefore, a tolerance around the goal point, defined as a circle around the goal is defined.

Given the robot's current pose and an applied velocity, the DWA approach requires to have a motion model for the robot. The motion model projects the what the robot pose would be, if a velocity pair is applied to it for a time period. The robot we used for our implementation is a non-holonomic two wheeled robot. While one can use the general motion equations derived in [17], we used linear approximated motion equations for our robot in Equation 6. Given a robot pose $q^t = (x^t, y^t, \theta^t)$ at time t and an input velocity (v, w), the projected robot pose at time $t + \Delta t$ is:

$$q^{t+\Delta t} = f_{motion}(q^t, v, w, \Delta t) = \left\{ \begin{matrix} x^t - \frac{v}{w} sin(\theta^t) + \frac{v}{w} sin(\theta^t + w\Delta t) \\ y^t + \frac{v}{w} cos(\theta^t) - \frac{v}{w} cos(\theta^t + w\Delta t) \\ \theta^t + w\Delta t \end{matrix} \right\} \tag{6}$$

3.4.3 Results

In this section, we provide qualitative results both in simulation and on the real robot. We used a non-holonomic drive robotic platform, Segway RMP-200, for the real experiments. We used our laser-based torso tracking method presented in Section 4.2.2.

3.4.3.1 Simulation

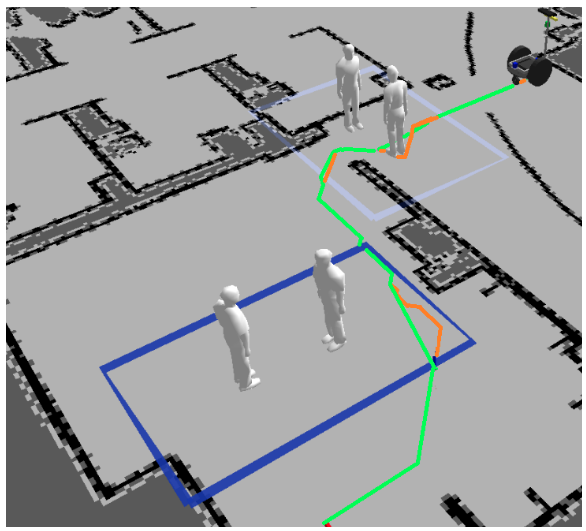
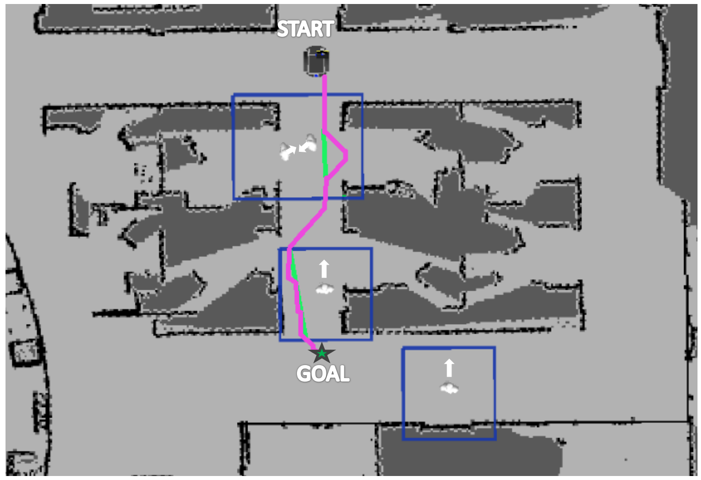
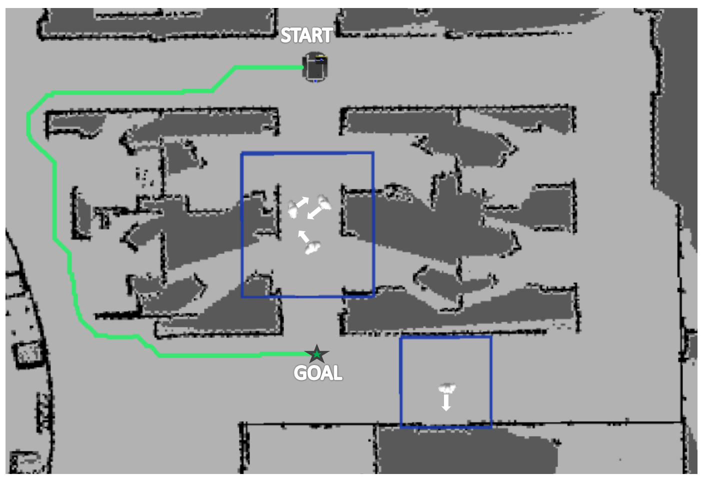


Figure 7: "Room Problem". The robot is outside a room and the goal is inside the room. Traditional planners can not solve the problem because two people are blocking the doorway. Our planner generates a tentative path, with the initial global plan shown in green and the dynamic refinements are shown in orange.

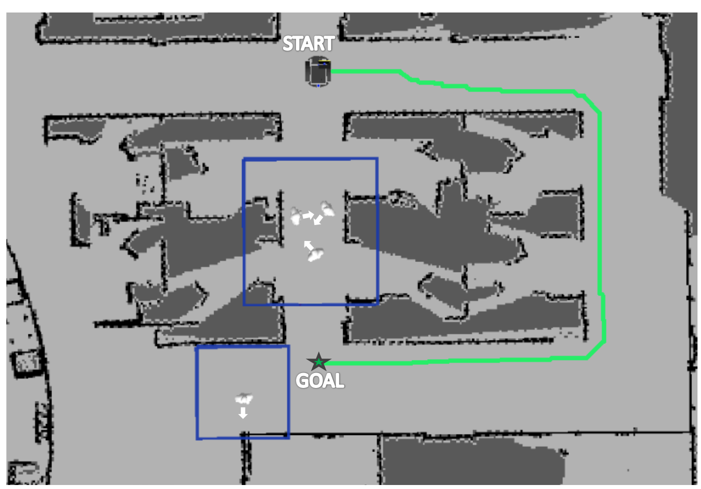
Room Problem: In this scenario, the robot is outside the room and a point inside the room is given as the goal (Figure 7). Traditional planners can not return a solution in this scenario because there is not enough space for the robot to navigate inside. There are two people standing at the doorway and there are two more standing people inside. The static plan and dynamic plans are shown in green and orange, respectively. This path is planned for the current time but makes assumptions about future positions of humans. Note that the dynamic planner modifies only the parts of plan inside group regions (blue rectangles). In the first group region (doorway), the



(a)



(b)



(c)

Figure 8: Paths differ drastically with the poses and grouping of humans. a) The robot takes shortest route, traveling in the vicinity of a group of two and another individual. b) third individual joins the group. Robot takes a longer path that doesn't have humans on path. c) fourth person changes his position, leading the robot to take the longest route.

static plan involves going between the humans. Dynamic simulation suggests that people will get closer to each other if the robot drives towards the side. In the second group region, since two humans are oriented to each other, going between them would add a high disturbance cost, therefore the static plan avoids going between them. Safety costs encourages staying far from the humans, but not too far because a longer path would increase the path length cost. The robot is further led to stay closer to the room boundaries in the dynamic planner due to the repulsive forces from both humans.

Office Environment: Goal of the robot is to navigate to a goal position in an office environment with 4 standing people (Figure 8). In this scenario, we show how the planned path is drastically changing with the poses of humans even though the start and goal position of the robot doesn't change. There are 3 main ways the robot can navigate to its goal: left, center or right corridor.

In the first configuration in Figure 8(a), two people are grouped together as they are looking at each other and likely conversing. The robot decides to take the center corridor, first slightly disturbing the speaking duo, then switches sides in the corridor and reaches its goal. In the figure, the dynamic path (pink line) is overlaid on the static path (green line).

In the second configuration in Figure 8(b), The third person at the center corridor joins the conversation. Now we have 2 group regions (rectangles) in the scene. Since passing through a group of 3 people would introduce a high disturbance cost in addition to the safety cost, the robot decides to take a longer route (left corridor). Since this path does not intersect any group regions, no dynamic simulation is done.

In the third configuration in Figure 8(c), the group of three hasn't moved, but the fourth person person has changed its position. In this case, if the left corridor is taken again, an additional safety cost would be incurred. Therefore the robot decides to take the longest route (right corridor). Again, since the robot travels far from humans, no dynamic simulation is done.

3.4.3.2 Real Robot

We demonstrate our anticipatory navigation planner on the real system in two environments: hallway and kitchen. Each scenario is run twice under different human positions and behaviors in order to show how the planner responds.

Hallway passing: In this scenario (Figure 9), robot's goal is to navigate to the end of the hallway. In the first run, humans move as the robot anticipates. In the second run, humans do not move as anticipated, and the robot adjusts its path. Each step is described in the caption of the figure. In both cases, the initial plan is to disturb the interaction by going between the two. This is because the safety cost for getting close to one of the humans was more dominant than the disturbance cost.

Narrow corridor: In this scenario (Figure 10), robot's goal is to drive towards the exit door. There are 3 people nearby the robot. The robot can either take the shorter route that is the direct path, or take a longer path that is to the left of the table. Each important step is described in the caption of the figure. The first run shows that the robot may plan hoping to influence the human. The second run shows that the robot may take a longer route if the disturbance and safety costs are going to be large.

3.5 Speed Limits for Safe Navigation

In this chapter so far, we studied navigation planning that aims to generate human-friendly paths and safety of people must be at the utmost importance. In order to ensure the safety of people, we introduced **Safety Cost** in Section 3.4.1 as well as repulsion forces in Section 3.4.1.1. However, a seemingly safe path generated by our approach can still lead to an accident because it is may not be possible for the robot to track every human in the environment all the time. For example, when the robot is turning a corner, there is a risk that the robot suddenly encounters a person.

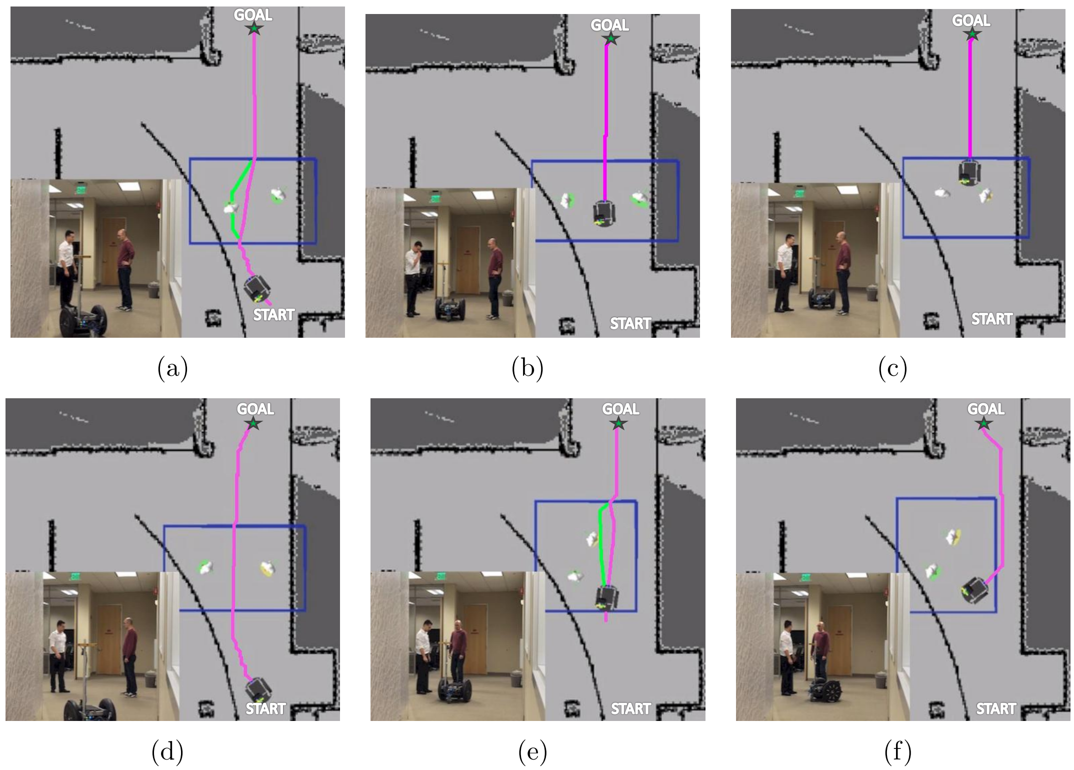
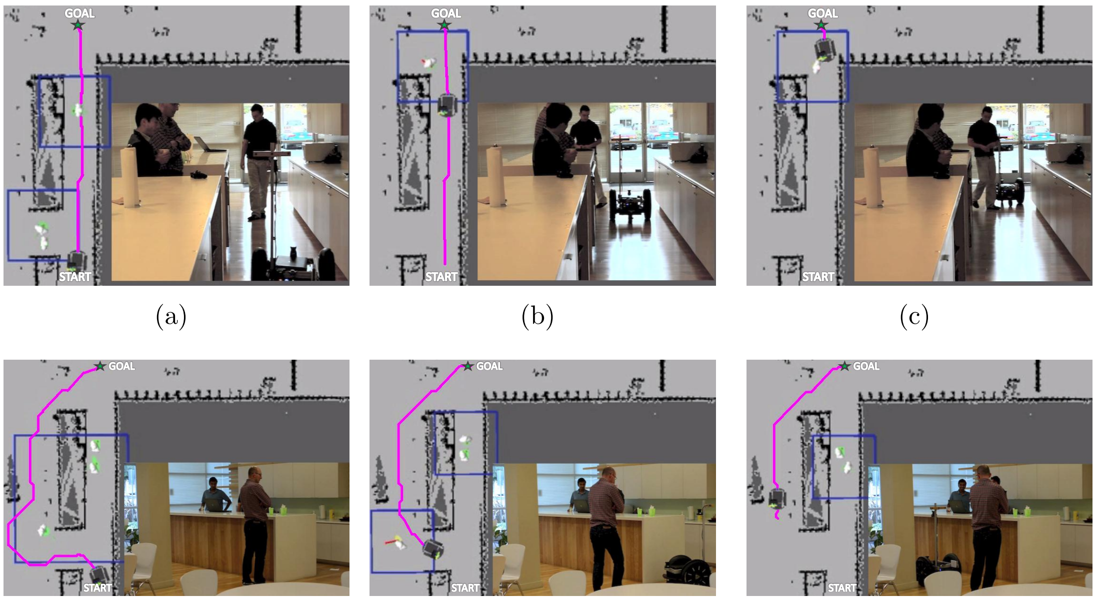


Figure 9: The Hallway scenario. 2 runs are shown in first and second rows.The static plan (green line) and dynamic plan refinement (pink line) are shown. First run: a) Navigation starts. The dynamic planner anticipates that people will give way to the robot when it starts to move towards them. b) Humans notice the robot, and give way by increasing the separation between them. c) The robot continues towards its goal and humans regroup. Second run: d) both the static and dynamic plan involves going in between humans again e) human on the right gets closer to the other person. Since a human made significant movement, dynamic planner re-plans. Plan no longer involves going in between. f) static planner periodic re-plan triggers, leading to robot to stick to the wall to the right.



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

Figure 10: The Kitchen scenario. In the first run, there are two people blocking the path to the left and one person at the narrow corridor. a) robot decides to take the shorter route, because it would disturb one person instead of two. There is not enough space to pass, and dynamic planner assumes the person would get out of the bottleneck to give way. b) human behaves as robot anticipated and gets out of the narrow passage. robot slows down because it enters the human region. c) person gets back to his original position, robot reaches the goal. In the second run: d) there are two people at the narrow corridor and one person on the left. The robot decides to take the longer route and pass the third person from left. The safety cost from the two others would be too high if the robot took the direct route. e) the person steps back as he recognizes the robot. since the person has moved, the dynamic planner re-plans and decides to pass from right. f) after the robot passes the person, it proceeds to its goal.

Since the robot have finite deceleration, a collision may be unavoidable. The speeds that mobile robots should exhibit is also dependent on the context. For instance, a robot should move slowly and carefully in a hospital room. On the other hand, the robot may navigate faster in a long office corridor. The speed limits should better be provided by experts or the building owners, who govern how fast the robots should navigate in a particular environment.

In this section, we introduce speed maps that sets the speed limits for mobile robots in an environment. We claim that usage of such speed maps make the robots safer and potentially more efficient. The speed map designed for the second floor of the College of Computing building at Georgia Tech is shown in Figure 11.

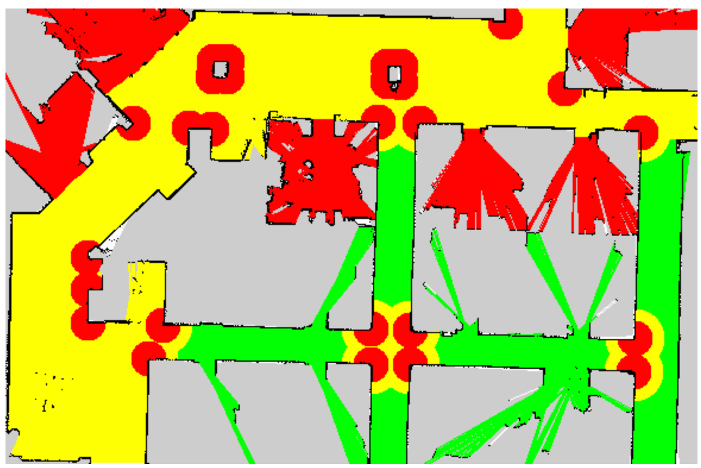


Figure 11: Designed speed limits for. The robot has to be relatively slow in red zones, can have moderate speed in yellow zones and is allowed to move relatively faster in green zones.

The free spaces in this map are divided into three zones:

1. Green zone: The robot is allowed to travel at relatively faster speeds.
2. Yellow zone: Human interaction is possible, top speed is less than the maximum allowed
3. Red zone: Human encounter is probable, top speed is severely limited.

This speed map is designed by hand using the following rules: Spaces corresponding to rooms and cubicles are covered as Red Zones. Blind corners are covered with a Red Zone close to corner and Yellow zone enclosing the Red Zone. Corridors are covered as Green Zones and the rest is covered as Yellow Zones.

The speed map depicted in Figure 11 is designed by hand, however the speed map generation can be automatized using automatic room categorization and additional processing. Room segmentation has been proposed in an interactive fashion by [15], as well as automatically, especially for creating topological maps [56].

3.5.1 Results

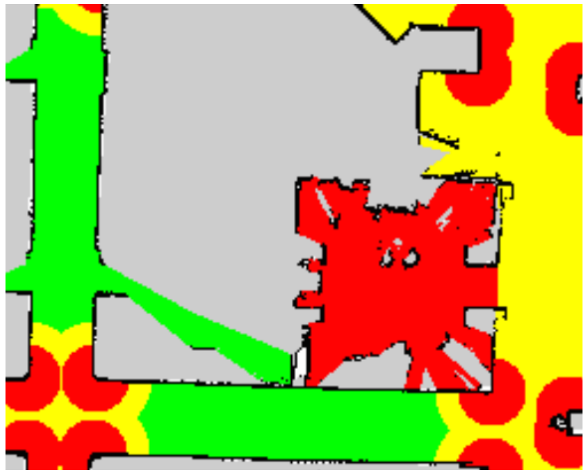
In this section, we evaluate the effect of applying speed limits in a navigation scenario. The robot is in an office environment and it has to turn a corner to reach the its goal. There is a person standing right around the corner and the person is not visible to the robot until it gets fairly close to the person. We had two conditions of speed limits:

- Condition 1: Fixed maximum linear speed of $v_{max} = 1.0m/s$
- Condition 2: Variable speed limits are used according to the speed map: $v_{max}(green) = 1.5m/s$, $v_{max}(yellow) = 0.5m/s$, $v_{max}(red) = 0.15m/s$

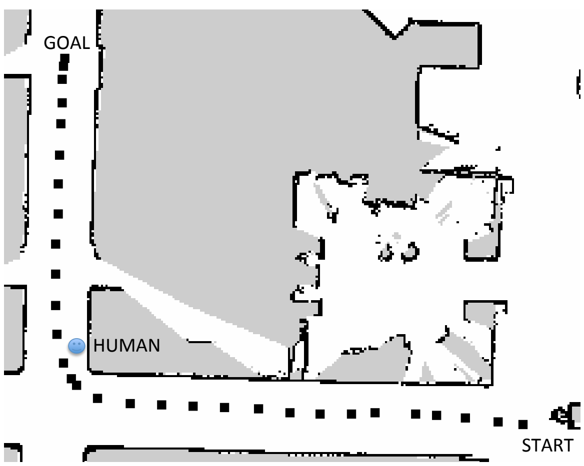
In both conditions, standard ROS Navigation is used, which found the lowest cost path on a costmap consisting of costs from nearby obstacles. The robot detected the person using our multimodal person detection and tracking method described in Chapter 4. We analyze the distance/velocity of the robot as it encountered the human and measure the time to reach to the goal as evaluation metrics.

The section of the speed map including the corner is shown in Figure 12(a). The trajectory with a fixed maximum speed is shown in Figure 12(b). Note that the robot got dangerously close to the human while it turned the corner in Condition 1 and it was traveling with about $0.3m/s$ when it detected the person. The trajectory using the proposed speed limits is shown in Figure 12(c). With this approach, the robot slowed down as it approached the corner, therefore allowing earlier detection of the human. The robot gave more space to the human in this case, the speed of the robot when it encountered the robot was $0.15 m/s$, half the speed of fixed max velocity case. Considering the speed of the robot and distance to the human at the time of encounter, we can say that the robot was safer.

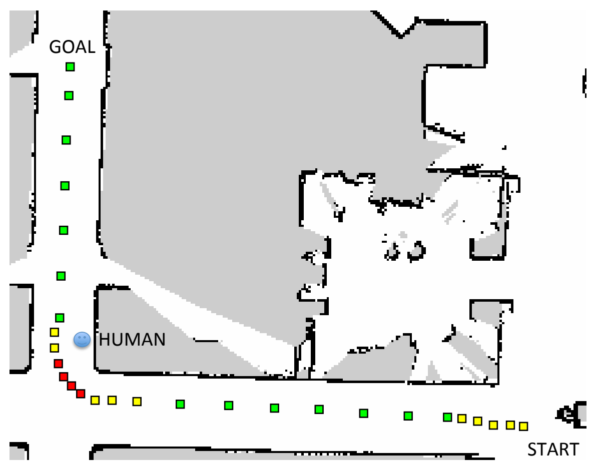
The second metric we measured was the time to reach the goal. The robot was more efficient with the proposed approach, reaching the goal in $28s$ compared to $29.1s$.



(a)



(b)



(c)

Figure 12: Comparison study of using a maximum top speed versus using location-dependent speed limits. Robot is given a fixed goal location. Right around the corner, there is a bystander human, who is not visible to the robot until the robot makes the turn. Points annotate robot position measured at fixed time intervals. a) Speed map of a corridor intersection at the second floor of College of Computing at Georgia Tech. b) Robot's top speed is fixed at $1.0m/s$. Note that the distance between robot positions are mostly constant. The robot gets very close to the bystander because it is moving relatively fast when it turned the corner. c) The robot is allowed to move with $1.5m/s$ in green, $0.5m/s$ in yellow and $0.15m/s$ in red zones. Colors of the sampled points on the path show the associated speed zone. It can be seen by looking at robot's positions that this approach was more gracious turning the corner and respecting human's personal space.

CHAPTER IV

MULTIMODAL PERSON DETECTION AND TRACKING

The ability to robustly track a person is an important prerequisite for human-robot interaction. To realize any task that involves humans, the challenge is the detection and tracking of humans in the vicinity of the robot considering the robot's movements, sensing capabilities and occlusions. The scope of how much information is needed from the human perception module depends on the objective of the application. First, the robot should determine if there are people nearby. If the robot senses people around, the robot should find out *where* they are. Representing people as points (x,y) in maps is common practice for navigation planning. If the task requires the robot to face a person, then the orientation θ needs be detected. The robot further can determine *who* the detected person is. Identification of humans is necessary for enabling non-generic service. Finally, the robot should interpret *what* the person is doing by analyzing the motion features and through gesture analysis. Tracking body parts of humans over time give significant information about human activity.

We focus on tracking people who are either walking or standing, as these are the two most common human poses around a mobile robot. Many full-body or body part detectors have been developed in the literature, reviewed in Section 4.1. Full-body detectors are not suitable for mobile robot navigation applications because of their inability of capturing the entire body with on-board sensors when people are close to the robot. We aim to robustly track a person 360° around the robot. However, most sensors have a limited field of view and using only a single detector can lead to a system with a single point of failure. Therefore, we think a multimodal detection system is better suited for on-board people tracking for our use cases.

Laser scanners are the natural sensor of choice as state-of-the-art mobile robots are already equipped with an ankle-height laser scanner that is mainly used for navigation. The laser scanners we used on our robot are Hokuyo UTM 30-LX, which has 270° Field of View (FOV), 0.25° angular resolution, $40Hz$ refresh rate and $30m$ maximum range. We are only interested in detections in close range (less than $5m$). In that range interval, and the accuracy of each laser reading is $\pm 3cm$, which is sufficient for our use cases. The relatively higher accuracy and resolution are the two advantages of laser scanners over cameras and RGB-D cameras. Cameras, on the other hand, have the advantage of providing richer information, which can be used to extract body parts. We use a combination of detectors using either a laser scanner and RGB-D camera for robustness and better coverage, described in Section 4.2. Representing people as a points in the map is sufficient for mobile robot navigation and each detector produces a point as a person hypothesis. We use a real-time probabilistic tracking framework that relies on the fusion of the multiple person detections, described in Section 4.3. For certain applications, identifying specific users allows the robot to go beyond generic capabilities. We present our face recognition method in Section 4.4.

4.1 Related Work

Person detection was first addressed by the computer vision community as an object detection problem. Early research on person detection using vision is surveyed by Moeslund [54]. Face detection is a common method for detecting people, with the work of Viola and Jones [95] being the most popular one. See Zhang [101] for a survey on contemporary approaches on vision based face detection. Another popular topic has been pedestrian detection in crowded scenes Leibe [45] and Tuzel [93].

In 2000's, laser scanners became the de-facto sensor for localization and mapping. Laser scanners are usually placed slightly above floor for obstacle avoidance, therefore leg detection is common practice. Early works by Montemerlo [55] and Schulz [77] focused on tracking multiple legs using particle filters. Legs are typically distinguished in laser scans using geometric features such as arcs [97] and boosting can be used to train a classifier on a multitude of features [2]. Topp [90] demonstrates that leg tracking in cluttered environments is prone to false positives. For more robust tracking, some efforts fused information from multiple lasers such as Carballo's work [11], which uses a second laser scanner at torso level. Glas [22] uses a network of laser sensors at torso height in hall-type environments to track the position and body orientation of multiple people. Several works used different modalities of sensors to further improve the robustness. Kleinehagenbrock [38] and Bellotto [6] combine leg detection and face tracking in a multi-modal tracking framework. Other examples include combining sound localization and vision [8] and combining RFID tracking and vision [21].

Laser-based person methods pertains tracking of humans in 2D, projected to floor plane. Tracking of the body parts has long been a topic of interest in vision [5, 79]. With the recent introduction of 3D sensors such as the Velodyne, Swissranger and Kinect, more robust tracking became possible. Spinello [84] trains geometrical features at different height levels in the 3D point cloud for pedestrian detection. Ganapathi [18] estimates body part locations with a probabilistic model. One of the well-known skeleton tracking algorithms is the Microsoft Kinect SDK by Shotton [78], which trains decision forests using simple depth features in a vast database. This software is not suitable to work on a mobile robot as it is designed to work on a stationary sensor. In the robotics community, there are efforts to develop skeleton trackers that work on mobile robots and in unstructured scenes [10].

Face recognition is a widely used application as surveyed by Phillips [69]. One of the pioneers in face recognition uses a set of patch masks for features that doesn't necessarily correspond to eyes, ears or noses [92]. [102] combines PCA (Principal

Component Analysis) and LDA (Linear Discriminant Analysis) to improve the generalization capability when only a few samples are available.

There has been some work to identify humans using 3D data, such as the head-to-shoulder signature [37] and body motion characteristics [58]. Biometric person identification techniques, such speaker recognition [35], 3D ear shape [98] and multi-modal cues [19] have potential to be more accurate than face recognition. However, these approaches are better suited to work in controlled environments.

4.2 Person Detection

In this section, we present our person detectors, namely leg detection (Section 4.2.1) and torso detection (Section 4.2.2). We also use an implementation of an upper body detector by Mitzel [52], which uses a template and the depth information of a RGB-D camera to identify upper bodies (shoulders and head), designed to work for close range human detection using head mounted cameras.

4.2.1 Leg Detection

A front-facing laser scanner at ankle height is used for leg detection. The output of a laser scanner at each iteration is an array of range measurements, represented in the polar coordinate system. We first convert the range data to Cartesian coordinate system:

$$x_i = \sum_{\phi=\phi_{start}}^{\phi_{end}} r_i \cos(\phi)$$

$$y_i = \sum_{\phi=\phi_{start}}^{\phi_{end}} r_i \sin(\phi)$$

Then we apply segmentation, Segmentation produces clusters of consecutive scan points, which due to their proximity, have a high likelihood of belonging the same object. Two adjacent distance measurements are considered to be in the same segment if the Euclidean distance between them is below a threshold value. Starting from the start of the range array, a new segment is started if $|r_i - r_{i+1}| < d_{cluster}$. Although some approaches use a variable segmentation threshold that is a function of the range, we use a fixed clustering threshold $d_{cluster} = 0.1m$. The segmentation process results in a set of segments $\mathbf{S}$. A set of geometric features are extracted from the laser segment.

In a laser scan, legs can appear in different patterns [90]. We look only single leg and person-wide blob patterns as these two cover all the ways legs can be seen in a laser scan. Depending on the application, we accept either only the single leg pattern or both of the patterns (explained in Section **??**.

There are a number of geometric features that can be extracted from a laser segment, as delineated by Arras [2]. We use three geometric features that is used to detect a leg: segment width, circularity, and Inscribed Angle Variance (IAV):

1. Segment Width: Measures the Euclidean distance between the first and last point of a segment S_i

2. Segment Circularity: This measure is a simple measure to assess if the segment shape resembles a circle. The circularity criterion we used is the ratio of the perpendicular distance from the middle point to the line segment that connects start and end points, to the segment width. For example, in a perfect half circle in Figure 13, the circularity criterion is $|\overline{P_0 P_n}/d_{mid} = 0.5$. In case of a laser scan, as can be seen in Figure 14, we again consider the ratio of d_{mid} to segment width. For this calculation we only consider the middle point as it provides a simple heuristic on circularity.

3. Inscribed Angle Variance (IAV): This feature is originally proposed by Xavier [97], in order to detect circles. We adopt IAV in order to detect legs, which are not necessarily circle-shaped, especially for the person-wide blob pattern. As an example, inscribed angles on a circle is shown in Figure 15. As a geometric

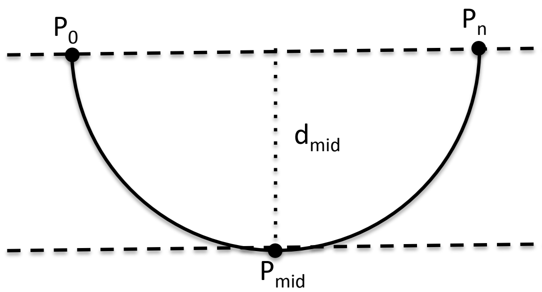


Figure 13: Circularity criterion in a perfect circle is: $|P_0P_n|d_{mid} = 0.5$

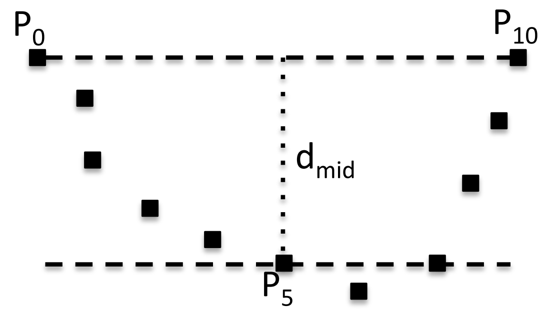


Figure 14: Circularity criterion in a this laser segment is: $|P_0P_{10}|/d_{mid}$

property of the circle, $\angle P_0P_1P_4$ and $\angle P_0P_2P_4$ are equal angles. IAV for a given set of points is the average of all inscribed angles:

$$IAV_S = \sum_{P=P_1}^{P_{n-1}} \angle P_0PP_n$$

For a perfect circle, $IAV_S = 90°$. For shapes that are not perfect circles but are similar to circles, IAV feature should be consistent. Laser segments from a leg usually resemble a circle, therefore we use IAV as one of the features for leg detection.

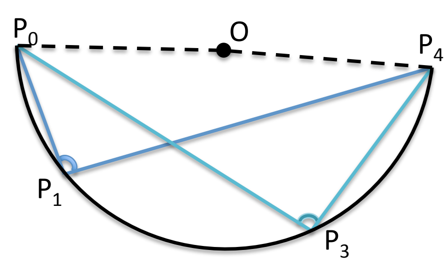


Figure 15: Inscribed angles of an arc are shown in the figure. Inscribed Angle Variance (IAV) is calculated by taking the average of all inscribed angles on a laser segment.

In order to be able to use these values, we first found the nominal feature values for an average human leg. We captured the laser scan data while the robot followed a person through an office environment. The following method used for this experiment will be discussed in detail in Section 5.2. For the training set, two people's legs were

Segment type	Width(m)		Circularity		IAV($radians$)	
	μ	σ	μ	σ	μ	σ
Single Leg	0.13	0.03	0.25	0.15	2.23	0.4
Personwide blob	0.33	0.07	0.14	0.09	2.61	0.16
Other	0.22	0.12	0.1	0.11	2.71	0.38

Table 1: Table shows average and standard deviations of geometric leg features calculated in our dataset.

recorded with different clothing (shorts, baggy pants and trousers) to account for variance in the leg parameters. About 17×10^3 Single Leg patterns and 0.6×10^3 person-wide blobs were manually labeled in the data. In addition, 120×10^3 segments were labeled as 'other' or 'not a leg'. The average and variance of the aforementioned geometric features for single leg, personwide blob, as well as other segments are given in Table 1.

For every segment S_i in a test laser scan, we first extract the geometric features f_1^i, f_2^i, f_3^i. We then calculate the weighted Mahalanobis distance to the average leg parameters for the each leg pattern:

$$D_{mah}^i = \sum_{j=1}^{n_{features}} w_j \frac{(f_j^i - \mu_j)^2}{\sigma_j^2} \tag{7}$$

where w_j are the weights for each feature, mu_j and $sigma_j$ are pulled from Table 1. The resulting Mahalanobis distance is then compared with a detection threshold. If $Dmah_{leg}^i < Threshold_{leg}$, the segment S_i is considered a detection. $Threshold_{leg}$ defines how many standard deviations away from the average features are allowed. In our implementation, we empirically set the feature weights as: $\mathbf{W}_{leg} = (0.35, 0.26, 0.39)$, in the feature order given in Table 1. For normal operation, we set $Threshold_{leg} = 1.5$, which accounts for about %95 of the detections. If only one person is being tracked, we use a higher threshold. The reason behind will be explained in Section 4.3.

4.2.1.1 Associating Leg Segments

After single leg patterns are detected, we try match the leg segments. We extend our leg detection approach to determine which leg segments are connected. Note that this method applies if there is a RGB-D camera pointing to the lower body of the human. For each leg segment pair, if both of them are within the FOV of the RGB-D sensor, we use our algorithm to determine whether there is a connectivity between two candidate leg segments. If a connectivity is found, then the leg segments pair is qualified to be a leg segment pair representing a person. See Figure 16 as an example result. Figure 17 shows the flow chart of the association algorithm.

Figure 16: Two person detections are seen in this figure. Our leg segment association algorithm propagates pixels vertically from candidate leg segments and connects leg pairs.

First, the centroids each of the two candidate leg segments are found. These points are projected onto the depth image acquired from the RGB-D camera. At each iteration, each leg segment, our algorithm first propagates horizontally to both directions in the depth image, then the center pixel is located and it propagates 1 pixel vertically ($+z$ direction). If there are no connectivity after a number of iterations, then we conclude that the candidate leg pair does not represent a person. If there

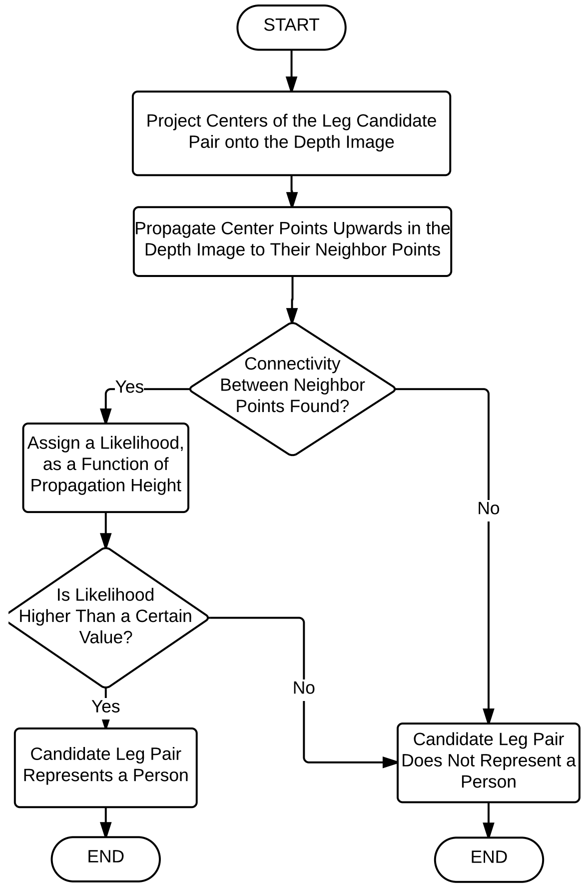


Figure 17: Flow chart for determining if two leg segment candidates belong to a person.

is a connectivity at some point, we then assign a likelihood score to the pair as a function of the vertical propagation height. If this score is higher than a threshold, then the algorithm concludes that the leg candidate segments represent a person. The propagation scoring eliminates most of the false positives due to sensor noise and non-human shapes.

4.2.2 Torso Detection

In this section, we describe our torso detection approach. For this detector, we used another Hokuyo UTM 30-LX laser scanner, placed at torso height ($1.27m$). Our approach relies on fitting an ellipse to laser segments and determining the detection result by interpreting the axis lengths (Figure 18). Our torso detector allows us to detect the orientation of the person unlike the laser-based leg detectors, therefore this detector is also suitable for applications that relies on extracting the orientation of the person from a single laser scan.

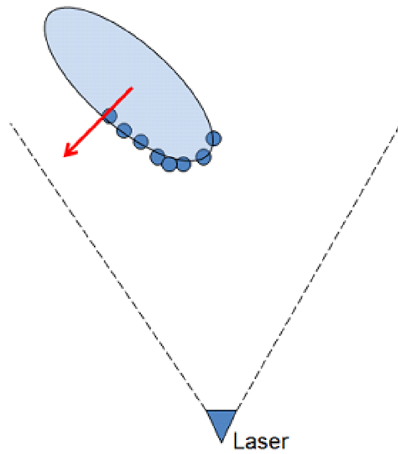


Figure 18: Our torso detector fits and ellipse to the human torso and estimate its position and orientation.

The first step to detect torsos in a laser scan is to segment the laser scan. We use the same segmentation technique used for leg detection, explained in Section 4.2.1. We then fit an ellipse to each laser segment. We use a numerical ellipse fitting method that solves the problem with a generalized eigensystem, introduced Fitzgibbon [16]. This ellipse fitting method is robust, efficient and ellipse-specific, so that even very noisy sensor data will always return an ellipse. Compared to iterative methods, it is computationally very efficient, therefore the speed of the calculations is limited to the laser scan refresh rate.

The ellipse fitting algorithm provides us with the centroid and orientation of the ellipse as well as the minor and major axis lengths. To disambiguate the front/back of a person, we assume that people are facing the sensor when they are first detected. While this is a significant limitation our current system, one can potentially utilize face detection as will be described in Section TODO to estimate if the person is facing towards the robot or not.

To detect a torso in a laser segment, we use the minor and major axis lengths,

Torso Features	μ	σ
Width(m)	0.44	0.12
Circularity	0.32	0.18
IAV($radians$)	2.57	0.38
Major axis length(m)	0.39	0.08
Minor axis length(m)	0.17	0.06

Table 2: Table shows average and standard deviations of geometric features for a human torso in laser scans.

as well as the three geometric features introduced in Section 4.2.1. We collected 450 laser scans in total while a person stood in front of the sensor and made a one full turn around himself. We calculated the mean and standard deviation of the all five features, which is given in Table 3. For a given laser segment, we find the weighted Mahalanobis distance in Equation 7 to the averaged parameters. If $Dmah^i_{torso} < Threshold_{torso}$, the segment is considered a detection. The feature weight constants we used was $\mathbf{W}_{torso} = (0.19, 0.09, 0.35, 0.24, 0.13)$, in respective order given in Table 2. These values were empirically determined, although one can do more sophisticated analysis for optimal weights.

Figure 19 shows how the torso detection rate changes for a given Mahalanobis Distance Threshold in our dataset. What is not displayed in the plot is that higher torso detection rate also means higher rates of false positives. For normal operation, we set $Threshold_{torso} = 1.25$, which accounts for about %90 detection rate. If the tracker is dedicated to track only a single person, then we use a higher threshold: $Threshold_{torso} = 2.5$. The reasoning behind this threshold selection will be discussed in Section 4.3.

4.2.2.1 Evaluation of Torso Detection

In order to evaluate the accuracy of the position and orientation estimations of our torso detection method, we collected torso data from 23 people. Subjects were instructed to stand on 4 targets at different distances with 8 different orientations on

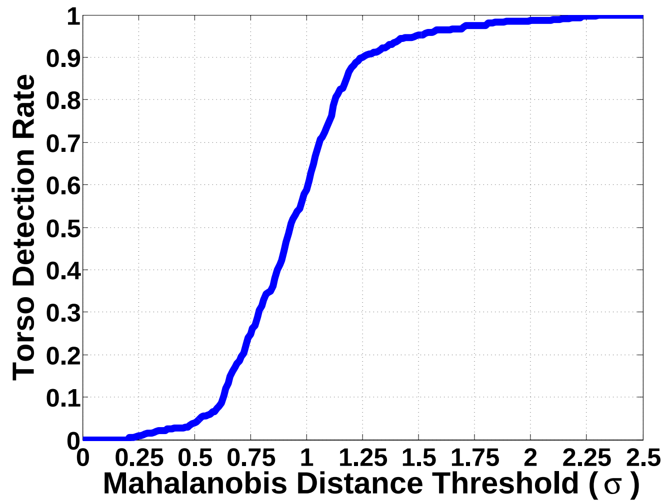


Figure 19: Torso detection rate vs weighed Mahalanobis Distance Threshold in our dataset

each target. Experimental setup from the sensor's view is shown in Figure 20. For each pose at every target, we logged the position and orientation estimation of the torso detector and compared it with ground truth, which is fixed.

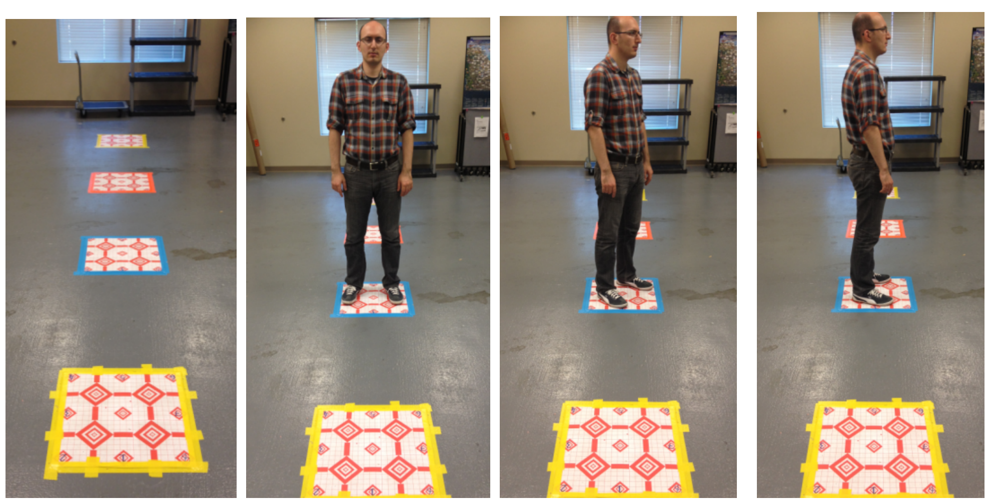


Figure 20: Experimental setup for the evaluation study of the Torso Detector.

Table 3 shows the angular error at every target distance and human orientation with respect to the laser scanner.

The average positional error was about $5cm$ regardless of the distance and the

Distance To Laser	N	NE	E	SE	S	SW	W	NW	ALL
1.0m	4°	12°	22°	13°	5°	7°	26°	17°	13°
2.5m	5°	16°	19°	10°	3°	6°	14°	17°	11°
4.0m	4°	10°	30°	16°	7°	11°	21°	17°	15°
5.5m	5°	11°	41°	18°	10°	6°	38°	23°	19°
ALL	4°	12°	27°	14°	6°	7°	24°	18°	14.5°

Table 3: Average orientation error of the torso detector with respect to distance from sensor and body pose in a study with 23 people

orientation of the human. The average orientation error throughout all the experiments was 14.5°. Error in orientation, however, varied greatly by pose of the person with respect to the laser scanner. Average error in orientation differed slightly with respect to the distance from the sensor and was the least with 11° when the humans were 2.5m away from the sensor. We attribute to the fact that when humans closer than 2.5m to the laser scanner, it captures more of the arms, which makes the fitted ellipse slightly worse. The orientation of the human with respect to the sensor had a significant effect on orientation error. Least error was achieved when people faced the sensor (4°) or the opposite way (6°). On the other hand, average orientation error was $24° - 27°$ when humans are perpendicular to the sensor, because a large portion of the torso is not visible to the laser scanner in that configuration.

4.3 Person State Estimation

The position and velocity of the person can not be determined by direct observation due to measurement noise and false detections. Therefore there is a need for a filtering algorithm in order to estimate the state of a person. Using a state predictor for human movement has two advantages. First, the predicted trajectories are smoother than raw detections. Smooth tracking helps the robot maintain consistent trajectories for high-level applications such as Person Following (Section 5). Second, it provides a posterior estimation that can be used for data association when there is a lack of matching detections. This allows the tracker to handle temporary occlusions. We use a discrete Kalman Filter [30] to predict the position of a person. There are other types of filtering techniques available in the literature, such as Particle Filters [32]. Since the results of the person state estimator is used by time-critical higher level applications, the tracker should come up with an estimate in real time. Therefore the choice of using Kalman Filters was motivated by its computational efficiency. Efficient person state estimation also increases the safety of the robot, as the robot can react faster if there are people in close proximity.

Accoding to Hicheur [27], humans tend to maintain a constant speed when they are walking straight and reduce speed while turning. We used constant velocity model which assumes people will maintain their speed. Even though this assumption is not always true, it provides a simple model without sacrificing too much from tracking performance.

The Kalman filter estimates a process as a predictor-corrector cycle using feedback control. The process has two cycling states: time update and measurement update as shown in Figure. Time update projects the state forward by using the current state and error covariance. Measurement update is responsible for the feedback and corrects the previous estimate.

The Kalman Filter is governed by two linear stochastic difference equations:

$$s_k = As_{k-1} + Bu_{k-1} + w \tag{8}$$

$$z_k = Hs_k + v \tag{9}$$

Where s_k represents the process state at time step k, A is the state propagation matrix, B relates the optional control input u, z_k is a measurement, H is the measurement observation matrix. w and v represent the process and measurement noises, respectively, drawn from normal probability distributions with zero mean $N(0, Q)$ and $N(0, R)$.

We define the state of a person s_k at time step k as:

$$s_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} \tag{10}$$

where (x_k, y_k) is the position and $(\dot{x}_k, \dot{y}_k)$ is the velocity of the person in Cartesian Coordinates. With the constant velocity model, the time update equations are:

$$x_k = x_{k-1} + \dot{x}_{k-1}\Delta t_k + w \tag{11}$$

$$y_k = y_{k-1} + \dot{y}_{k-1}\Delta t_k + w \tag{12}$$

$$\dot{x}_k = \dot{x}_{k-1} \tag{13}$$

$$\dot{y}_k = \dot{y}_{k-1} \tag{14}$$

resulting in the following Kalman Filter matrices:

$$A = \begin{bmatrix} 1 & 0 & \Delta t_k & 0 \\ 0 & 1 & 0 & \Delta t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{15}$$

where Δt_k is the time difference from the previous detection. A track is lost if there are no detections for a fixed amount of time. At every time update of a filter, if Δt_k is larger than a fixed threshold, the track is killed.

The reason B vector is zero is that we track people in the world frame and robot motion is already accounted for with robot localization. For this reason, we assume there are no control inputs to our system. The noise matrices we used are:

$$Q = qI_4 \qquad\qquad R = rI_2 \tag{16}$$

where we used $q = 0.02$ and $r = 1.0$ in practice.

Our approach is multimodal in the sense that asynchronous measurements are accepted from different sources as long as they provide a positional estimate in the respective sensor frames. Using the latest localization information, this position is converted to the world frame and then fed as a measurement to the active filters. We apply an additional layer of filtering to every detection before it is considered a measurement. We check if a new detection is in collision with the static map, and it if is in collision, we reject that particular detection. The check against the static map is fast and helps reduce false positives in practice. We use Nearest Neighbor (NN) data association [4], which is a reasonable compromise between performance and computational cost.

Depending on the task, a single person or multiple people must be tracked. We examine each case below:

- **Single target tracking:** For some tasks, such as person following, dedicated tracking of a single specific user is required and tracking bystanders is not required for task success. In this case, our goal is to keep tracking the specific user, so we significantly relax the detection thresholds of the detectors. Even though doing so results in more spurious detections, we do not start more than a single track. This approach improves the tracking performance of a single person.

- **Multi-target tracking:** When the robot is navigating to a goal point with human bystanders, tracking multiple people at the same time is necessary. Moreover, losing track of a bystander would not be very detrimental to task success. We keep a separate Kalman filter for each tracked person. If a detection is matched to multiple filters, only the closest filter is associated with the detection and the other filters are considered to have no detections for that time step.

4.4 Face Recognition

For certain interactive navigation tasks such as finding a specific person, a robot needs to have person recognition capability. Our person recognition approach uses face recognition and optionally shirt color features. We detect faces in RGB images using the popular face detector by Viola and Jones [95]. We use the Eigenface method by Turk and Penland [92] for face recognition. Our approach allows new faces to be trained on-the-fly.

With the $Eigenface$ approach, face are represented in a lower-dimensional space. Sirovich and Kirby [82] showed that dimension reduction method Principal Component Analysis (PCA) can be used on face images to form a set of basis features. The main idea of PCA for faces is to find vectors that best account for variation of face images in all training images. These vectors are called $eigenvectors$. Then a face space is constructed called $eigenfaces$ and the images are projected onto this space. Our approach of face recognition works as follows:

Figure 21: Example results of our person recognition method is shown in the image. We use $Eigenfaces$ face recognition method and optionally shirt color recognition.

1. A person unknown to the system comes up to the robot and initiates training.

2. Robot asks the person to turn his face one side to another, and takes M face and shirt images of this person.

3. Eigenfaces from the entire training set is calculated, and every known face is projected to the corresponding M-dimensional weight $facespace$.

4. After training is completed, face recognition is reactivated.

5. A distance value from face recognition and optionally from shirt color recognition is received and it is thresholded for a decision. An example recognition result is in Figure 21.

Using the UI of the robot, a user can start training and adjust the information in the person database. The person data is managed by a SQLite database hosted locally on the robot.

Shirt color recognizer can be used when there is little time between the training and recognition. Activating the shirt recognition should improve recognition and reduce false positive detections. We assume a rectangular region below the face captures the shirt (1.5 times below the the face rectangle size). The distribute the histogram into bins using normalized RGB color space because of its relative robustness to lighting. For detection, we calculate the distance between the training histogram to the test histogram using Earth Mover Distance [73]. The color histogram is adaptively updated at every high confidence detection in order to account for illumination changes. The overall person score is calculated by a weighted average of face and shirt distance.

CHAPTER V

PERSON FOLLOWING

In this chapter, we focus on one aspect of human-robot interaction, namely person following in an indoor environment. There are many scenarios in which a person following robot can be useful. For example, a robot can carry luggage of travelers in airports, or groceries in a supermarket. Person following is also the enabling capability for interactive acquisition of the *Home Tour* scenario discussed in TODO. The robot needs to know how to follow a person before building an environment representation and providing services to the user. There are two properties a person following behavior should achieve: robust following and social awareness.

Typically, a service robot operates in a dynamic and populated environment, therefore the robot must be able to keep track of a single person even when they are temporarily occluded. Multimodal person tracking that is presented helps the robot to have better estimates of a user's position. As discussed in Chapter 4, for the person following task, the robot has to track a designated user, and the detection thresholds of detectors are relaxed for robust tracking at the expense of more false positive detections.

The robot not only has to keep appropriate distance to the user, but it also has to recognize *what* the user is trying to accomplish and move accordingly. For example, during the home tour scenario, when the user stops, the robot should predict that the user is going to annotate a landmark, and it should come beside the user instead of standing behind. Moreover, the robot should be smarter when passing doors or following a person who is cutting a corner. In order to be able to handle these scenarios, the robot should act beyond purely reactive following behavior. It is desirable for the robot to anticipate what action the user is likely going to take and act accordingly.

In this chapter, after referring to previous studies on person following robots in Section 5.1, we present the elementary person following method in 5.2. After that, in Section 5.3 we demonstrate situation awareness of a person following robot in three commonly encountered scenarios: door passing, user activity awareness and handling corners. In Section 5.4, we present an application of person following to telepresence robots.

5.1 Related Work

A robot that follows a person is a widely studied scenario in robotics. A relevant body of work is pursuit evasion [12], however the target is trying to evade the follower. In person following robots, we assume that the target is cooperating with the follower.

In one of the earliest works in this area by Sidenbladh [80], robot keeps the person centered in the camera image using a P controller. Prassler [70] also offers a fully reactive approach using the *Velocity Obstacles* concept, which uses the velocity of the target to find allowable velocities of the robot that guarantees avoiding collision if both the target and the robot move with constant speed. The approach is applied on a wheelchair, even though social constraints were not considered. The robot does not necessarily follow a person directly from behind. Gockley [**?**] observed how people walk together. It is reported that partners who were conversing tend to look forward with occasional glances to each other. Ohya [62] presents a following method to escort a target on the side while avoiding obstacles. It was assumed that the target would move with the same acceleration and velocity. Murakami [59] presents a method to first estimate the sub-goal of the leading person and then following as if the robot knows the goal.

Park [67] models the problem as a control problem and offers an algorithm based on Model Predictive Control. [53] employs randomized tree expansion and biases the calculated paths towards a sub-goal, which is the current position of the person. Hoeller [28] adopts the virtual targets idea and selects a goal position in a circular region around the person. Stein [86] proposes choosing and following a leader to handle navigation in crowds.

Some of the relevant works considered the social side of the interaction. Gockley compared two elementary following methods: direction following, where the robot always attempts to drive towards the tracked person, and path following, where the robot follows the path the person took. It was shown that direction following behavior was perceived as more human-like and natural than path following. Yuan's [99] system switches following behavior between parallel following, direction and path following depending on the layout of the obstacles. Zender [100] emphasizes on situation awareness for following and studies cases of handling doors and corridors. To handle the doors, the robot increases its following distance and that leads the robot to wait for a while. Following in a corridor is handled with an approach similar to Pacchierotti [65], and the robot's speed is increased. [47] presents a system that is capable of responding to verbal and non-verbal gestures and follow a person. When following a group of people, a common method is to choose and follow a leader. Granata [24] presents behaviors such as going towards, following and searching a user. Ota [63] touches upon the recovery functions whenever the robot loses tracks of the leading person.

5.2 Basic Person Following

In this section, we describe our basic person following method. To keep track of the person

When the following behavior is initiated from a higher level process, first the person to be followed must be tracked. The robot looks for the closest person in the vicinity of the robot (within $2m$). If no person is detected for some time, then the command is invalidated. We use the person detection and tracking system described in Section 4.

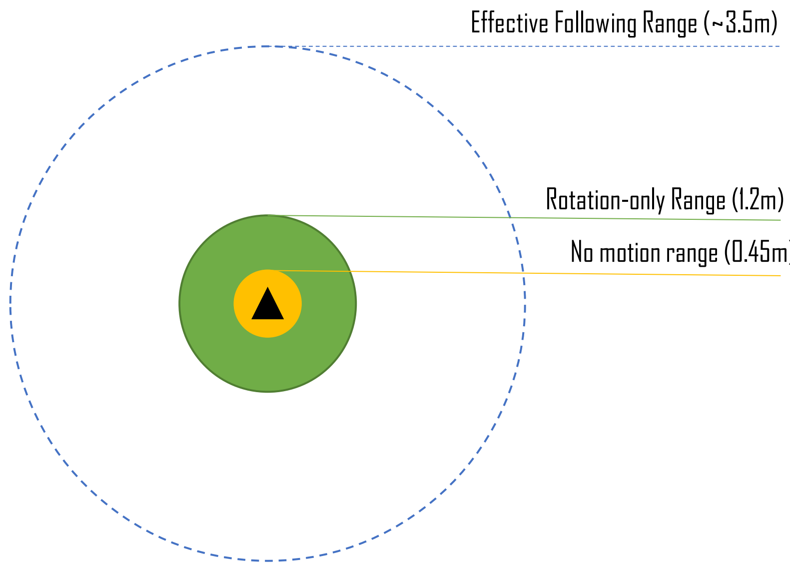


Figure 22: Overhead view of relevant ranges for person following. Robot is represented as the triangle in the middle.

In the basic person following mode, the robot has three different strategies depending on the distance towards the followed person. The distance to the user is calculated as the distance from the center of the robot base to the person's current location estimation. We used Hall's characterization of personal spaces in order to determine the distance limits. See Section TODO for a review of Hall's work. The three distinct zones and corresponding robot behavior are given as follows:

Intimate Zone $[0 - 0.45m]$**:** In this short interval, the person is very close to the robot, therefore any motion of the robot may be potentially unsafe. Therefore the robot comes to a complete halt in this zone ($v = 0, w = 0$). This behavior also allows the user to safely interact with the on-board User Interface.

Personal Zone $[0.45 - 1.2m]$**:** When the robot is in the personal zone of the user, the robot stops and only rotates towards the followed person ($v = 0, w = w_{rotation}$). The rotational velocity is determined with a P controller, with the error term defined as the difference between the current orientation and the tracked person's orientation. The rotational velocity is capped at a fixed value, so that the person feels comfortable with the rotation of the robot.

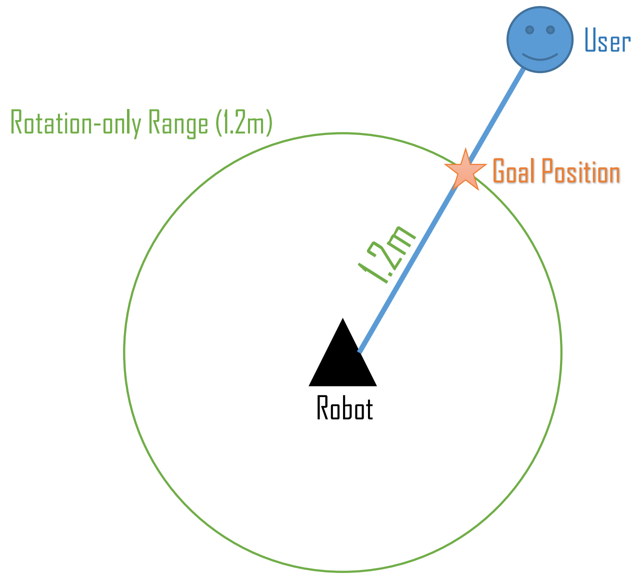


Figure 23: An illustration of how the goal position is calculated when the user is in the social space $[1.2m - 3.5m]$.

Social Zone $[1.2m - 3.5m]$**:** In this range interval, the robot executes the main following behavior. At every time step, a goal pose that is $1.2m$ away from the and headed towards the user is calculated, see Figure 23 for an illustration. A collision-free path is found and the robot executes this path until a new measurement from the tracked person is received. The path is found using Dynamic Window Approach (DWA) local planing method. We use the ROS implementation of DWA for the basic following behavior.

Sometimes it is inevitable that the person tracking system loses the target, particularly when the person is consistently faster than the robot or the person goes outside the range of the sensors (further than $\sim 3.5m$ in our case). When this happens, the robot will attempt to go to its last calculated goal position and look for the person. By this means, the robot attempts to keep up with the lost person as far as possible with the hope that the person will re-appear in the vicinity of the last seen position. After the robot reaches this goal, it stops and waits for an amount of time. If the user is saved to the database, or the robot already knows that he/she is in the database, then the face recognition system described in Section 4.4 is activated. Otherwise, the robot continues following the closest person that appears in this position. If no person is detected within a fixed amount of time, 5 seconds in our implementation, then the robot declares that the person is lost.

5.3 Situation Aware Person Following

5.3.1 Door Passing

5.3.2 User Activity Awareness

5.3.3 Corners

5.4 Application To Telepresence Robots

A telepresence robot can be described as *Skype on wheels*, where a remote user teleconferences while having the control of the movement of a robotic system in a physical environment. Telepresence robots constitute a promising area in the consumer robotics industry as evidenced by multiple start-up companies working on telepresence products. However, currently all the telepresence robots that are available in the market are controlled by manual driving - usually via the keyboard or a joystick. In this section, we present an implementation of person following on a telepresence robot and a user study that evaluates effect of having person following capability on a telepresence robot.

Telepresence robots are a level above video conferencing since the robot is used as the communication medium and the remote user can now control the movement. Therefore, the spatial interaction between people and a telepresence robot in social situations is worth investigating. One of those situations is moving with a group of people. In an effort to analyze the spatial and verbal interaction, we focus on engagement with one person where the remote user interacts with the person while following him/her in a corridor. This is situation is very likely to happen in office environments, for example when the remote user is having a discussion with a co-worker while walking to his office after a meeting. As telepresence robots become more common, there will be need to have the functionality of autonomous following of a person so that the remote user doesn't have to worry about controlling the robot.

We evaluate our system by conducting a user study, where there are two following conditions:

1. Manual Person Following: Robot is controlled with an Xbox controller

2. Autonomous Person Following: Initiated by clicking on a user in RGB-D image

The aim of the user study is to measure how remote users like using the autonomous following feature compared to the manual. For the study, the remote user has a task that consists of listening to a passage the followed person reads and answering related questions after the interaction. We also observe subjects' experiences using the system, get useful feedback and pinpoint future challenges that can be helpful designing new applications for telepresence robots.

5.4.1 Robot Platform

The system described in this paper is implemented on an experimental telepresence robot shown in Figure 24. The robot has a differential drive base and can be used for about 8 hours with full charge. For the experiments in this paper, the speed of the robot was limited to 0.55 m/s. A laser scanner with 360° field of view, which was taken from Neato XV-11 vacuum cleaning robot, was mounted horizontally at 0.3m height. The system runs on Windows 7 and Microsoft Robotics Developer Studio (MRDS) as its distributed computing environment. The remote user connects to the robot via wireless internet and communicates with others using Skype. On the

Figure 24: The telepresence robot platform we used for our experiments.

remote end, . The robot is also equipped with an omni-directional microphone and a high-end speakerphone.

There are two operation modes for the robot: Teleoperation and Autonomous Person Following. A Xbox 360 Wireless Controller is used to remotely teleoperate the robot. A wide-angle camera is placed on top of the monitor and tilted slightly downward to help the remote user to see the floor, robot base and people's faces at the same time. A Kinect Sensor is also placed above the monitor. Person following is initiated through the user interface shown in Figure 25.

A modified version of the local planner used in Section 3.4.2 is used for person following. A utility function consisting multiple factors, including the respective position to the person, is optimized over multiple steps using Breadth-First Search. Details of the planning method can be found in [14].

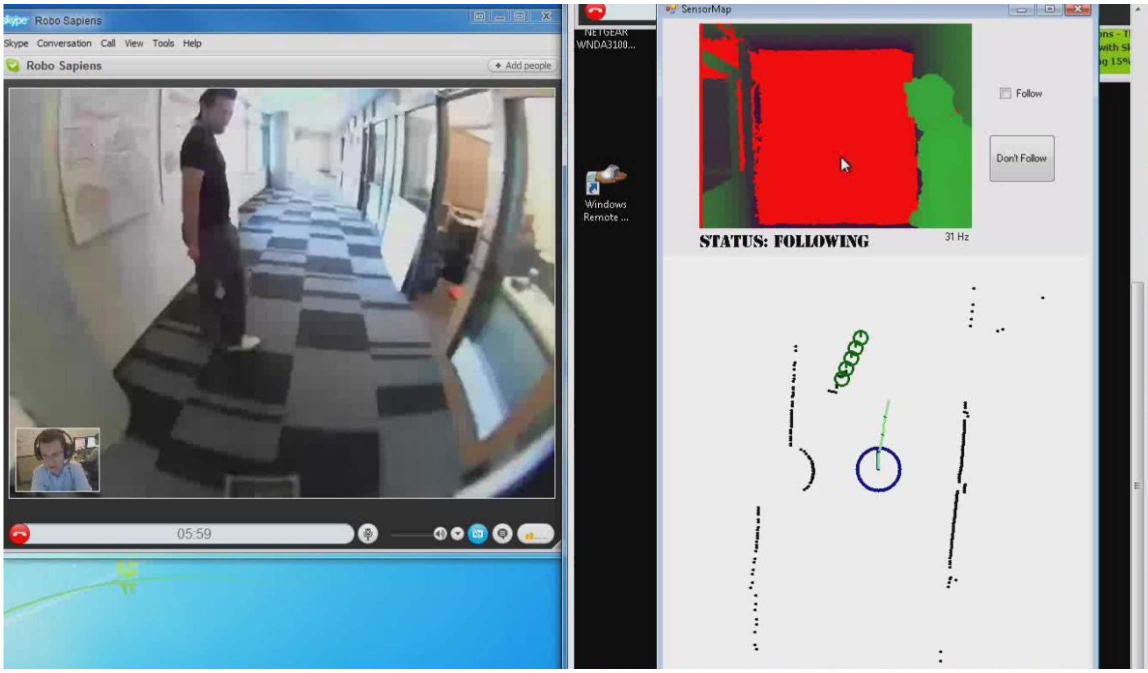


Figure 25: User Interface of the robot for the remote user.

5.4.2 User Study

In this study, remote user is the subject and the followed person is the experimenter. To investigate the effectiveness of using autonomous person following for an interaction task, we ran a controlled experiment and varied manual vs. autonomous following within subjects.

Design: The experiments were conducted in working hours and bypassers were allowed to walk across the experiment area or talk. The subjects were given the task of following the experimenter through the course for a lap and listen to the passage he is reading. In the first run, the subject used the autonomous following or teleoperation method to follow a person and complete the lap. In the second run, the subject used the other method. At the end of each run, the subject was asked to complete a 4-question quiz about the passage. The passages and quiz questions were taken from Test of English as a Foreign Language (TOEFL) listening section examples. One passage was about "behaviorism" and the other one was about "manila hemps", and passages were chosen so that they are at a similar difficulty level. The time it takes to read a passage corresponded approximately to the same time a lap is completed. We also asked numbered 7 point Likert scale questions, administered after each run, about how *Understandable* the experimenter was, *Easiness of UI*, if the robot exhibited *Natural Motions*, how *Safe* the remote user felt, if the subject was able to *Pay Attention* to the passage, how *Fast* the robot was and how much *Fun* the subject had. At the end of both runs, the user was asked which method he/she will prefer over the other for this type of a scenario. The exact questionnaire was show in Section/Appendix TODO.

Participants: 10 volunteers participated in the study (6 male and 4 female between the ages of 25-48). Participants consisted of 4 researchers and 6 interns at Microsoft Research. 5 of the participants had little knowledge, 4 had average knowledge and 1 had above average knowledge on robotics. The participants weren't gamers: 4 participants never played console games, 4 played rarely, 1 sometimes played and 1 often played. 6 of the participants often used video conferencing software, while 2 sometimes and 2 rarely used. 9 of the participants were not native English speakers and all of them had taken the TOEFL before. Participants were recruited through personal relations and were given a small gift (valued at approximately US$10) for their help.

Procedure: The participants were first greeted by the experimenter and instructed to complete a pre-task questionnaire regarding their background. The robot was shown to the participant and basic information about its capabilities was told. The experimenter explained the task while walking with the participant in the corridor and showing the course to be followed. Participants were told that they should stay close to the experimenter while he is walking and there will be a quiz regarding the passage afterwards. The participant was informed that there are 2 operation modes:

manual and autonomous person following.

Before the experiment started, the participant went through training for about 15 minutes. First, the participant learned the basic controls for the Xbox controller when he/she was nearby the robot. Then the participant was taken to the remote station, which was in a room about 20 meters away from the corridor area. The participant was informed about the UI and was shown how the autonomous following can be activated. Then a test run was executed, where the remote user followed the experimenter via teleoperation and had a conversation.

After the training, the actual run was executed using either the manual or autonomous method. When the lap was completed, first the passage quiz, then the survey questions were answered by the subject. Then the second experiment using the other method was executed, and the second passage quiz and survey questions were given to the subject. As the last question, the subject was asked to state his/her method of preference. Lastly, the participants were debriefed about the study and engaged in a discussion. We switched the starting method for every other experiment in order not to bias the subjects' opinions about one particular method.

Measures: We had three measurement criteria to compare manual vs autonomous following: 1) Number of correct answers to passage quizzes: Assuming the standardized TOEFL exercises were of same difficulty, we ran a paired $t-test$ on two groups of autonomous and manual. 2) Survey questions: We ran a paired $t-test$ using 7-point Likert Scale on each of the seven questions. 3) Preferred Method: We looked at which method subjects chose over the other one.

Results: Out of 4 quiz questions, the correct answers for autonomous group ($\mu = 2.9$, $\sigma = 0.9$) were more than the manual group ($\mu = 2.2$, $\sigma = 1.2$) but the statistical difference was not statistically significant ($t(9) = 1.48$, $p = 0.17$ on $t-test$).

Table 4 summarizes the survey results. For *Understandable* and *Fun*, the scores

Question	Autonomous		Manual Drive		$t - test$	
	μ	σ	μ	σ	p	t
1. Understandable	4.0	1.5	3.6	1.7	0.47	0.73
2. Easy UI	6.5	0.9	5.0	2.2	0.06	2.13
3. Natural Motion	5.4	1.0	3.5	1.9	0.03	2.52
4. Safe	5.1	1.7	2.3	1.4	0.01	3.09
5. Pay Attention	5.3	1.8	3.4	1.5	0.02	2.63
6. Fast	3.9	0.3	4.3	0.8	0.10	-1.8
7. Fun	5.3	1.5	5.1	1.7	0.66	0.45

Table 4: Survey results of the user study for person following for telepresence robots. Table displays survey question average and standard deviations for the two conditions: Autonomous Person Following and Manual Person Following.

slightly favored autonomous method but the difference wasn't statistically significant. Manual method User Interface (gaming controller) was found to be easy to use ($\mu = 5.0$, $\sigma = 2.2$), but the UI for autonomous method (clicking) was found to be marginally easier ($\mu = 6.5$, $\sigma = 0.9$), ($t(9) = 2.13$, $p = 0.06$). The motions of the robot was found to be significantly more *Natural* to have a conversation for autonomous ($\mu = 5.4$, $\sigma = 1.0$) than manual ($\mu = 3.5$, $\sigma = 1.9$), ($t(9) = 2.52$, $p = 0.03$). Participants thought they were able to *Pay more Attention* to the passage the experimenter is reading when the robot was following the him autonomously ($\mu = 5.3$, $\sigma = 1.8$) compared to manual control ($\mu = 3.4$, $\sigma = 1.5$) and the statistical difference was significant ($t(9) = 2.63$, $p = 0.02$). Participants have found the autonomous method ($\mu = 5.1$, $\sigma = 1.7$) much safer than manual method ($\mu = 2.3$, $\sigma = 1.4$) and there was a significant difference between two groups ($t(9) = 3.09$, $p = 0.01$). The speed of the robot was found to be neither fast nor slow for both methods ($\mu = 3.9$, $\sigma = 0.3$) and ($\mu = 4.3$, $\sigma = 0.8$).

All 10 subjects chose autonomous person following over teleoperation for this task.

5.4.3 Design Implications

Our user study showed that a person following behavior is desirable for telepresence robots when there is interaction. The follow-up discussions also agreed with the survey results, as one subject (R10) stated: *"It just gives me more focus and concentration."* Below, we list our observations and implications for future research and design for telepresence robots:

Motor Noise: Even though the motors on the robot were relatively quiet, 8 out of 10 participants expressed that the motor noise made communication harder. This justifies the close scores we collected in the survey question asking if the subject was able to understand what the experimenter was saying. (R8) was disturbed by the noise: *"When I was driving, it was always this constant sound. It was worse for the autonomous one. It was constantly adjusting and compensating for the movement."* On the other hand, (R5) found the motor noise useful: *"I actually like it because it gives me the feedback whether I'm driving faster or slower. It also gives me a little bit feeling of life."* Thus, although excessive motor noise should be avoided, some noise might be useful.

Wireless Connection: Second most cited problem for video conferencing was the video quality and time lags. (R8) clearly expressed why it was hard to walk with the experimenter using the manual method: *"The frame rate drops all of a sudden and you have no choice but to stop."* Another subject (R9) made use of the displayed sensor data when the video conferencing quality went bad: *"Because of the lag, I just switched to the Kinect (depth image) and the overhead view (laser)."* This was possible because the wide angle camera image was coming from Skype whereas sensor displays were received from the Windows Remote Assistance. Clearly, a big challenge for telepresence systems is to deal with wireless connection problems.

Natural Interaction: Even though the participants thought the motions of the robot were natural to have a conversation ($\mu = 5.4$, $\sigma = 1.0$), some didn't feel it was a natural way to communicate. As seen in Figure 24, the screen displaying the remote user's face is flat and it introduced problems when the robot was traveling on the side of the person. (R5), when asked about walking side by side: *"..we don't* *have face-to-face. It is not really a conversation."* This raises design considerations on how the remote user's face is brought out. One of the subjects (R5) discovered that the microphone characteristics are different than human hearing: *"I don't have a distance sense if the experimenter is further away or close. If you have the fading audio, then I'll immediately notice."* Whether a telepresence robot should exhibit the same characteristics of human perception or not is an open question and needs further investigation.

Assisted Teleoperation: Telepresence robots should possess a layer to assists the remote user to avoid obstacles and collisions. *Safety* ratings for the manual method were very low ($\mu = 2.3$, $\sigma = 1.4$) and (R8) expressed the concern: *"I was especially worried about running into the experimenter."* This suggests that scenarios involving interaction would demand more attention of the remote users. The teleoperation should also be intuitive and be similar to driving modalities that people are already used to. (R4) stated: *"I was thinking about Manual mode compared to driving a car."* before suggesting *".. maybe something like a cruise control might be good."*

Gaming Experience: Since the robot was controlled by a gaming console controller, some participants likened the manual mode to gaming. (R9) said: *"Manual is like playing video games."* and (R5) said: *"I don't play video games so controlling those consoles is not natural to me."* Thus, it is possible that gamers are less likely to have trouble driving the robot. This observation is also made by Takayama [88].

Long Term Interaction: None of the subjects participated in our study had used a telepresence robot before. (R6) justified the inability to use the manual method: *"Maybe if I have some more practice for about several hours of driving the robot, I can use manual as well as autonomous."* (R8) on having fun using teleoperation: *"It was fun because it was the first time I did it but I can imagine that over time, I'll get bored of it."* The *Fun* question in the survey received similar scores for autonomous and manual, possibly because using a telepresence robot was a new experience for the subjects. Studies regarding long term interaction for telepresence robots can yield interesting results, as in [44].

Error recovery: When the person was lost during following, the UI displayed a text that the person was lost so that the remote user can re-initiate the following by clicking on the person. None of the subjects complained about the robot losing the person. When asked explicitly about the robot losing the experimenter, (R10) answered: *"That's not a big deal in comparison to me driving the robot."* Therefore, applications developed for telepresence robots can take advantage of the human being in the loop and does not have to be error-free for deployment.

5.4.4 Discussion

User studies showed that autonomous person following is a desired capability for a telepresence robot and it was favored over direct teleoperation for an accompanying task. Autonomous following was found to be safer, easier to use and helped the remote users to pay more attention to the conversation instead of the robot control. From the experience earned from user studies, there are still interesting challenges to explore in terms of human-robot interaction for telepresence robots.

CHAPTER VI

PERSON GUIDANCE

when the robot has to guide a person from his/her current position to another place, it should support the person's activities and guide him/her in the way he/she wants to be guided

6.1 Related Work

The earliest works in guide robots focused on the implementation and long-term deployment of tour guide robots in public places such as museums. Burgard [9] presents the robot Rhino, that was deployed to a museum for 47 hours. The Minerva robot was an improved model over Rhino [89], and was deployed to a museum with an order of magnitude larger floor space. This robot was in operation for two weeks, and it was able to have short-term interaction with people , i.e. head motion, facial expressions. Siegwart [81] presents a robot that was deployed to an exhibition for 6 months. Nourbakhsh [60] presents a project where four guide robots were deployed to museums for a period of five years. The authors remark that it is indeed possible to deploy guide robots public places, unsupervised. All these tour-guide robots had various degrees of autonomy and was received with enthusiasm. However, in all of these works it was apparent that there is a need for research in the area of Human-Robot Interaction.

Pacchierotti [64] demonstrates an office guide robot, but the main focus is on passing people in corridors. Clodic [13] presents another robot deployed in a museum. It was reported that a continuous interaction all along the guiding mission is fundamental to keep visitor's interest. Martin [50] studies the scenario of guiding a visitor in an office environment and focuses on robust person tracking. Pandey

[66] focuses on the leave-taking of the guided person. The robot predicts the intent of the discontinuation of the task and either breaks the mission or searches for the user depending on the waiting time. Martinez-Garcia [51] focuses on the scenario of guiding a group of people with multiple robots at the same time. Garrell [20] works on a similar problem, where the task of the two robots is to control group of people and guide them. Another relevant scenario is the evacuation scenario, in which there is a danger and robots guide people to the safe a location [34, 72].

6.2 Guide Robot

In this section, we describe our method of person guidance. After a request to guide a person is received from a higher level process, the robot first plans a path using our navigation planner in Section TODO, or standard ROS Navigation. The robot continues on its path as long while constantly monitoring the distance between itself and the guided person. The robot adjusts its speed according to this distance. A straightforward method would be the stop-and-wait method, meaning the robot would continue on its path normally if the person is within a distance threshold, and would stop as long as the robot is outside the radius. However, this method of guidance is not socially appreciated. Robot should consider the distance to the human and incorporate this information in its control strategy.

We use a variable speed profile so that the robot can better keep up with the person and the motions of the robot is smoother. We define a speed function that is a function of the distance between the robot and the user. This speed profile is shown in Figure 26. The robot moves at a low speed v_{safe} if the human is dangerously close. The speed is peaked at distance d_{peak} and the robot stops if the distance is larger than d_{guide}, which may indicate that the human is not interested in being guided. Note that this speed profile is subject to the static speed limits, i.e. v_{peak} is capped by the static speed map limit.

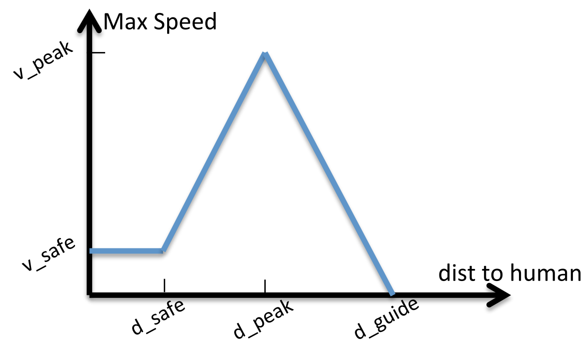


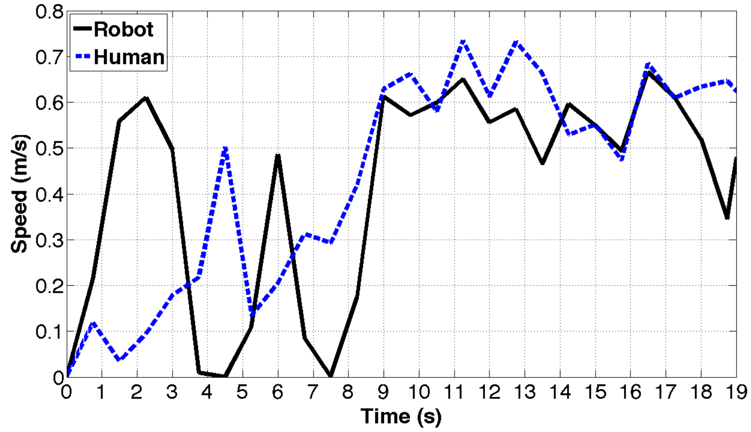
Figure 26: Speed profile of a person guiding robot as a function of the distance to the user.

In the second scenario, the robot has the same goal point and guiding a person. In the first condition, we use ROS Navigation but robot stops if the distance to the human is over a threshold. In second condition, we use our method of dynamic speed adjustment for guidance. In this scenario, there is no person standing around the corner. We measured the instantaneous speeds of the robot and the human.

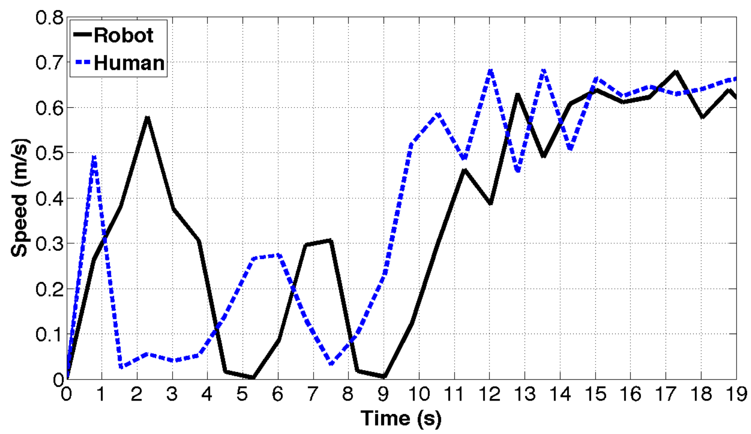
6.2.1 Results

The robot is given a fixed goal to guide a person, who is tracked with a torso-level laser scanner, by fitting an ellipse to the torso. We compared the velocity profile in Figure 27 with $d_{guide} = 1.7$m, $d_{peak} = 0.9$m, $d_{safe} = 0.1$m, $v_{safe} = 0.1$m/s, $v_{peak} = 1.0$m/s. We compared our approach with the simple strategy: If the human is closer than d_{guide}, then the navigation continues with a fixed max speed. Otherwise robot stops and waits.

In the experiment, when guiding was enabled, the human first waited until the robot stopped at d_{guide}. Then he took a step and waited for a second time, and then started following the robot. The comparison of robot speeds is given in Figure 27(a) for fixed max speed and Figure 27(b) with the speed profile. Between $t = 0$ and $t = 9s$, the accelerations are much more rapid for the fixed max speed case. Robots that exhibit high accelerations will likely be perceived as unsafe, therefore our approach exhibits a more socially acceptable behavior. Moreover, after the person



(a)



(b)

Figure 27: Comparison of robot and human speeds with respect to time. a) Standard ROS Navigation b) Our approach: accelerations are less steeper than a), which employs the dynamic speed adjustment for guidance.

started following ($t > 9s$), our approach is better at mimicking the speed of the human.

6.3 Application To Blind Users

In this section, we present our person guidance system specifically tailored for guiding blind users. Our approach consists of planning a path for the user and applying vibrations via a haptic belt to keep the user on the path.

System graph, persception ellipse

6.3.1 Tactile Belt

In the previous section, we assumed that the guided person can detect where the robot is and follow him/her. With a blind user, this assumption does not hold. Therefore we need a mechanism to give directions to the user. Readily available options for assistive interfaces are limited to Braille or devices that presents content with speech synthesis. These ways of presenting information have difficulty dealing with representing spatial information. We also think visually impaired individuals would prefer a non-speech interface because they mostly rely on their sense of hearing in daily life. We therefore use a tactile belt for navigation guidance, because it can represent directions and rotations, be worn discreetly and does not occupy the hearing sense.

The belt has 8 pancake vibration motors, linearly spaced around the waist, and the motors can be controlled asynchronously via an Arduino board. We used two distinct vibration patterns to control the person's movements:

1. Directional Movement: When the guided person should move in a direction

2. Rotational Movement: When the guided person should turn around self

HW/SW/Signals

6.3.2 Velocity to Vibration Mapping

6.3.3 Results

CHAPTER VII

CONCLUSION

Conclusion

APPENDIX A

QR CODE BASED LOCATION INITIALIZATION

QR Code Based Location Initialization

APPENDIX B

ASSISTED REMOTE CONTROL

Assisted Remote Control

APPENDIX C

VIBRATION PATTERN ANALYSIS FOR HAPTIC BELTS

Vibration Pattern Analysis for Haptic Belts

REFERENCES

[1] Althaus, P., Ishiguro, H., Kanda, T., Miyashita, T., and Christensen, H. I., "Navigation for human-robot interaction tasks," in *Robotics and Automation, 2004 IEEE International Conference on*, vol. 2, pp. 1894–1900, IEEE, 2004.

[2] Arras, K. O., Mozos, O. M., and Burgard, W., "Using boosted features for the detection of people in 2d range data," in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3402–3407, IEEE, 2007.

[3] Avrunin, E. and Simmons, R., "Using human approach paths to improve social navigation," in *8th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 73–74, IEEE, 2013.

[4] Bar-Shalom, Y. and Li, X.-R., *Multitarget-multisensor tracking: principles and techniques*, vol. 19. YBS Storrs, Conn., 1995.

[5] Baumberg, A. and Hogg, D., "Learning deformable models for tracking the human body," in *Motion-Based Recognition*, pp. 39–60, Springer, 1997.

[6] Bellotto, N. and Hu, H., "Multisensor-based human detection and tracking for mobile service robots," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 39, no. 1, pp. 167–181, 2009.

[7] Bennewitz, M., Burgard, W., Cielniak, G., and Thrun, S., "Learning motion patterns of people for compliant robot motion," *The International Journal of Robotics Research*, vol. 24, no. 1, pp. 31–48, 2005.

[8] Bernardin, K. and Stiefelhagen, R., "Audio-visual multi-person tracking and identification for smart environments," in *Proceedings of the 15th international conference on Multimedia*, pp. 661–670, ACM, 2007.

[9] Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and Thrun, S., "The interactive museum tour-guide robot," in *AAAI/IAAI*, pp. 11–18, 1998.

[10] Buys, K., Cagniart, C., Baksheev, A., De Laet, T., De Schutter, J., and Pantofaru, C., "An adaptable system for rgb-d based human body detection and pose estimation," *Journal of Visual Communication and Image Representation*, 2013.

[11] Carballo, A., Ohya, A., and Yuta, S., "Fusion of double layered multiple laser range finders for people detection from a mobile robot," in *Multisensor*

Fusion and Integration for Intelligent Systems, 2008. MFI 2008. IEEE International Conference on, pp. 677–682, IEEE, 2008.

[12] Chung, T. H., Hollinger, G. A., and Isler, V., "Search and pursuit-evasion in mobile robotics," *Autonomous robots*, vol. 31, no. 4, pp. 299–316, 2011.

[13] Clodic, A., Fleury, S., Alami, R., Chatila, R., Bailly, G., Brethes, L., Cottret, M., Danes, P., Dollat, X., Elisei, F., and others, "Rackham: An interactive robot-guide," in *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pp. 502–509, IEEE, 2006.

[14] Cosgun, A., Florencio, D. A., and Christensen, H. I., "Autonomous person following for telepresence robots," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4335–4342, IEEE, 2013.

[15] Diosi, A., Taylor, G., and Kleeman, L., "Interactive slam using laser and advanced sonar," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 1103–1108, IEEE, 2005.

[16] Fitzgibbon, A., Pilu, M., and Fisher, R. B., "Direct least square fitting of ellipses," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 5, pp. 476–480, 1999.

[17] Fox, D., Burgard, W., and Thrun, S., "The dynamic window approach to collision avoidance," *Robotics & Automation Magazine, IEEE*, vol. 4, no. 1, pp. 23–33, 1997.

[18] Ganapathi, V., Plagemann, C., Koller, D., and Thrun, S., "Real time motion capture using a single time-of-flight camera," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 755–762, IEEE, 2010.

[19] Garcia-Salicetti, S., Beumier, C., Chollet, G., Dorizzi, B., les Jardins, J. L., Lunter, J., Ni, Y., and Petrovska-Delacrétaz, D., "Biomet: a multimodal person authentication database including face, voice, fingerprint, hand and signature modalities," in *Audio-and Video-Based Biometric Person Authentication*, pp. 845–853, Springer, 2003.

[20] Garrell, A. and Sanfeliu, A., "Local optimization of cooperative robot movements for guiding and regrouping people in a guiding mission," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 3294–3299, IEEE, 2010.

[21] Germa, T., Lerasle, F., Ouadah, N., and Cadenat, V., "Vision and rfid data fusion for tracking people in crowds by a mobile robot," *Computer Vision and Image Understanding*, vol. 114, no. 6, pp. 641–651, 2010.

[22] Glas, D. F., Miyashita, T., Ishiguro, H., and Hagita, N., "Laser-based tracking of human position and orientation using parametric shape modeling," *Advanced robotics*, vol. 23, no. 4, pp. 405–428, 2009.

[23] Gockley, R., Forlizzi, J., and Simmons, R., "Natural person-following behavior for social robots," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, pp. 17–24, ACM, 2007.

[24] Granata, C. and Bidaud, P., "A framework for the design of person following behaviors for social mobile robots," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 4652–4659, IEEE, 2012.

[25] Hall, E. T., *The hidden dimension.* Anchor Books, 1966.

[26] Helbing, D. and Molnar, P., "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[27] Hicheur, H., Vieilledent, S., Richardson, M., Flash, T., and Berthoz, A., "Velocity and curvature in human locomotion along complex curved paths: a comparison with hand movements," *Experimental brain research*, vol. 162, no. 2, pp. 145–154, 2005.

[28] Hoeller, F., Schulz, D., Moors, M., and Schneider, F. E., "Accompanying persons with a mobile robot using motion prediction and probabilistic roadmaps," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1260–1265, IEEE, 2007.

[29] Hüttenrauch, H., Eklundh, K. S., Green, A., and Topp, E. A., "Investigating spatial relationships in human-robot interaction," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 5052–5059, IEEE, 2006.

[30] Kalman, R. E., "A new approach to linear filtering and prediction problems," *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[31] Kendon, A., *Conducting interaction: Patterns of behavior in focused encounters*, vol. 7. CUP Archive, 1990.

[32] Khan, Z., Balch, T., and Dellaert, F., "An mcmc-based particle filter for tracking multiple interacting targets," in *Computer Vision-ECCV 2004*, pp. 279–290, Springer, 2004.

[33] Khatib, O., "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.

[34] Kim, Y.-D., Kim, Y.-G., Lee, S. H., Kang, J. H., and An, J., "Portable fire evacuation guide robot system," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2789–2794, IEEE, 2009.

[35] Kinnunen, T. and Li, H., "An overview of text-independent speaker recognition: From features to supervectors," *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.

[36] Kirby, R., Simmons, R., and Forlizzi, J., "Companion: A constraint-optimizing method for person-acceptable navigation," in *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pp. 607–612, IEEE, 2009.

[37] Kirchner, N., Alempijevic, A., and Virgona, A., "Head-to-shoulder signature for person recognition," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1226–1231, IEEE, 2012.

[38] Kleinehagenbrock, M., Lang, S., Fritsch, J., Lomker, F., Fink, G. A., and Sagerer, G., "Person tracking with a mobile robot based on multi-modal anchoring," in *Robot and Human Interactive Communication, 2002. Proceedings. 11th IEEE International Workshop on*, pp. 423–429, IEEE, 2002.

[39] Koren, Y. and Borenstein, J., "Potential field methods and their inherent limitations for mobile robot navigation," in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1398–1404, IEEE, 1991.

[40] Kruse, T., Basili, P., Glasauer, S., and Kirsch, A., "Legible robot navigation in the proximity of moving humans," in *Advanced Robotics and its Social Impacts (ARSO), 2012 IEEE Workshop on*, pp. 83–88, IEEE, 2012.

[41] Kruse, T., Kirsch, A., Sisbot, E. A., and Alami, R., "Exploiting human cooperation in human-centered robot navigation," in *RO-MAN, 2010 IEEE*, pp. 192–197, IEEE, 2010.

[42] Kuderer, M., Kretzschmar, H., and Burgard, W., "Teaching mobile robots to cooperatively navigate in populated environments," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 3138–3143, IEEE, 2013.

[43] Lam, C.-P., Chou, C.-T., Chiang, K.-H., and Fu, L.-C., "Human-centered robot navigation owards a harmoniously human–robot coexisting environment," *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 99–112, 2011.

[44] Lee, M. K. and Takayama, L., "Now, i have a body: Uses and social norms for mobile remote presence in the workplace," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 33–42, ACM, 2011.

[45] Leibe, B., Seemann, E., and Schiele, B., "Pedestrian detection in crowded scenes," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1, pp. 878–885, IEEE, 2005.

[46] Lichtenthäler, C. and Kirsch, A., "Towards Legible Robot Navigation - How to Increase the Intend Expressiveness of Robot Navigation Behavior," in *International Conference on Social Robotics - Workshop Embodied Communication of Goals and Intentions*, 2013.

[47] Loper, M. M., Koenig, N. P., Chernova, S. H., Jones, C. V., and Jenkins, O. C., "Mobile human-robot teaming with environmental tolerance," in *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, pp. 157–164, ACM, 2009.

[48] Lu, D. V. and Smart, W. D., "Towards more efficient navigation for robots and humans," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1707–1713, IEEE, 2013.

[49] Luber, M., Spinello, L., Silva, J., and Arras, K. O., "Socially-aware robot navigation: A learning approach," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 902–907, IEEE, 2012.

[50] Martin, C., Böhme, H.-J., and Gross, H.-M., "Conception and realization of a multi-sensory interactive mobile office guide," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, vol. 6, pp. 5368–5373, IEEE, 2004.

[51] Martínez-García, E., Akihisa, O., Yuta, S., and others, "Crowding and guiding groups of humans by teams of mobile robots," in *Advanced Robotics and its Social Impacts, 2005. IEEE Workshop on*, pp. 91–96, IEEE, 2005.

[52] Mitzel, D. and Leibe, B., "Close-range human detection for head-mounted cameras," in *British Machine Vision Conference (BMVC)*, 2012.

[53] Miura, J., Satake, J., Chiba, M., Ishikawa, Y., Kitajima, K., and Masuzawa, H., "Development of a person following robot and its experimental evaluation," in *Proceedings of the 11th International Conference on Intelligent Autonomous Systems, Ottawa, Canada*, pp. 89–98, 2010.

[54] Moeslund, T. B. and Granum, E., "A survey of computer vision-based human motion capture," *Computer Vision and Image Understanding*, vol. 81, no. 3, pp. 231–268, 2001.

[55] Montemerlo, M., Thrun, S., and Whittaker, W., "Conditional particle filters for simultaneous mobile robot localization and people-tracking," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1, pp. 695–701, IEEE, 2002.

[56] Mozos, O. M., Triebel, R., Jensfelt, P., Rottmann, A., and Burgard, W., "Supervised semantic labeling of places using information extracted from sensor data," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 391–402, 2007.

[57] MÜLLER, J., STACHNISS, C., ARRAS, K., and BURGARD, W., "Socially inspired motion planning for mobile robots in populated environments," in *Proc. of International Conference on Cognitive Systems*, 2008.

[58] MUNSELL, B. C., TEMLYAKOV, A., QU, C., and WANG, S., "Person identification using full-body motion and anthropometric biometrics from kinect videos," in *Computer Vision–ECCV 2012. Workshops and Demonstrations*, pp. 91–100, Springer, 2012.

[59] MURAKAMI, R., MORALES SAIKI, L. Y., SATAKE, S., KANDA, T., and ISHIGURO, H., "Destination unknown: walking side-by-side without knowing the goal," in *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pp. 471–478, ACM, 2014.

[60] NOURBAKHSH, I. R., KUNZ, C., and WILLEKE, T., "The mobot museum robot installations: A five year experiment," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 4, pp. 3636–3641, IEEE, 2003.

[61] OHKI, T., NAGATANI, K., and YOSHIDA, K., "Collision avoidance method for mobile robot considering motion and personal spaces of evacuees," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),*, pp. 1819–1824, IEEE, 2010.

[62] OHYA, A. and MUNEKATA, T., "Intelligent escort robot moving together with human-interaction in accompanying behavior," in *Proceedings 2002 FIRA Robot World Congress*, pp. 31–35, 2002.

[63] OTA, M., OGITSU, T., HISAHARA, H., TAKEMURA, H., ISHII, Y., and MIZOGUCHI, H., "Recovery function for human following robot losing target," in *Industrial Electronics Society, IECON 2013-39th Annual Conference of the IEEE*, pp. 4253–4257, IEEE, 2013.

[64] PACCHIEROTTI, E., CHRISTENSEN, H., JENSFELT, P., and OTHERS, "Design of an office-guide robot for social interaction studies," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 4965–4970, IEEE, 2006.

[65] PACCHIEROTTI, E., CHRISTENSEN, H. I., and JENSFELT, P., "Human-robot embodied interaction in hallway settings: a pilot user study," in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pp. 164–171, IEEE, 2005.

[66] PANDEY, A. K. and ALAMI, R., "A step towards a sociable robot guide which monitors and adapts to the person's activities," in *Advanced Robotics, 2009. ICAR 2009. International Conference on*, pp. 1–8, IEEE, 2009.

[67] Park, J. J. and Kuipers, B., "Autonomous person pacing and following with model predictive equilibrium point control," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pp. 1060–1067, IEEE, 2013.

[68] Pellegrini, S., Ess, A., Schindler, K., and Van Gool, L., "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 261–268, IEEE, 2009.

[69] Phillips, P. J., Flynn, P. J., Scruggs, T., Bowyer, K. W., Chang, J., Hoffman, K., Marques, J., Min, J., and Worek, W., "Overview of the face recognition grand challenge," in *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, vol. 1, pp. 947–954, IEEE, 2005.

[70] Prassler, E., Bank, D., and Kluge, B., "Motion coordination between a human and a robotic wheelchair," in *Robot and Human Interactive Communication, 2001. Proceedings. 10th IEEE International Workshop on*, pp. 412–417, IEEE, 2001.

[71] Rios-Martinez, J., Spalanzani, A., and Laugier, C., "Understanding human interaction for probabilistic autonomous navigation using risk-rrt approach," in *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pp. 2014–2019, IEEE, 2011.

[72] Robinette, P. and Howard, A. M., "Incorporating a model of human panic behavior for robotic-based emergency evacuation," in *RO-MAN, 2011 IEEE*, pp. 47–52, IEEE, 2011.

[73] Rubner, Y., Tomasi, C., and Guibas, L. J., "A metric for distributions with applications to image databases," in *Computer Vision, 1998. Sixth International Conference on*, pp. 59–66, IEEE, 1998.

[74] Sasaki, T. and Hashimoto, H., "Human observation based mobile robot navigation in intelligent space," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp. 1044–1049, IEEE, 2006.

[75] Satake, S., Kanda, T., Glas, D. F., Imai, M., Ishiguro, H., and Hagita, N., "How to approach humans?-strategies for social robots to initiate interaction," in *Human-Robot Interaction (HRI), 2009 4th ACM/IEEE International Conference on*, pp. 109–116, IEEE, 2009.

[76] Scandolo, L. and Fraichard, T., "An anthropomorphic navigation scheme for dynamic scenarios," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 809–814, IEEE, 2011.

[77] Schulz, D., Burgard, W., Fox, D., and Cremers, A. B., "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 2, pp. 1665–1670, IEEE, 2001.

[78] Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., and Moore, R., "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[79] Sidenbladh, H., Black, M. J., and Fleet, D. J., "Stochastic tracking of 3d human figures using 2d image motion," in *Computer Vision?ECCV 2000*, pp. 702–718, Springer, 2000.

[80] Sidenbladh, H., Kragic, D., and Christensen, H. I., "A person following behaviour for a mobile robot," in *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, vol. 1, pp. 670–675, IEEE, 1999.

[81] Siegwart, R., Arras, K. O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., Jensen, B., Lorotte, A., Mayor, L., Meisser, M., and others, "Robox at expo. 02: A large-scale installation of personal robots," *Robotics and Autonomous Systems*, vol. 42, no. 3, pp. 203–222, 2003.

[82] Sirovich, L. and Kirby, M., "Low-dimensional procedure for the characterization of human faces," *JOSA A*, vol. 4, no. 3, pp. 519–524, 1987.

[83] Sisbot, E. A., Marin-Urias, L. F., Alami, R., and Simeon, T., "A human aware mobile robot motion planner," *Robotics, IEEE Transactions on*, vol. 23, no. 5, pp. 874–883, 2007.

[84] Spinello, L., Arras, K. O., Triebel, R., and Siegwart, R., "A layered approach to people detection in 3d range data," in *AAAI Conf. on Artif. Intell.(AAAI)*, 2010.

[85] Stein, P., Spalanzani, A., Santos, V., Laugier, C., and others, "Robot navigation taking advantage of moving agents," in *IROS Workshop on Assistance and Service robotics in a human environment*, 2012.

[86] Stein, P. S., Santos, V., Spalanzani, A., Laugier, C., and others, "Navigating in populated environments by following a leader," in *Ro-man 2013-International Symposium on Robot and Human Interactive Communication*, 2013.

[87] Svenstrup, M., Bak, T., and Andersen, H. J., "Trajectory planning for robots in dynamic human environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4293–4298, IEEE, 2010.

[88] Takayama, L., Marder-Eppstein, E., Harris, H., and Beer, J., "Assisted driving of a mobile remote presence system: System design and controlled user evaluation," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[89] Thrun, S., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Hahnel, D., Rosenberg, C., Roy, N., Schulte, J., and

others, "Minerva: A second-generation museum tour-guide robot," in *Robotics and automation, 1999. Proceedings. 1999 IEEE international conference on*, vol. 3, IEEE, 1999.

[90] Topp, E. A. and Christensen, H. I., "Tracking for following and passing persons," in *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pp. 2321–2327, IEEE, 2005.

[91] Trautman, P. and Krause, A., "Unfreezing the robot: Navigation in dense, interacting crowds," in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pp. 797–803, IEEE, 2010.

[92] Turk, M. A. and Pentland, A. P., "Face recognition using eigenfaces," in *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pp. 586–591, IEEE, 1991.

[93] Tuzel, O., Porikli, F., and Meer, P., "Human detection via classification on riemannian manifolds," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[94] Vasquez, D., Stein, P., Rios-Martinez, J., Escobedo, A., Spalanzani, A., Laugier, C., and others, "Human aware navigation for assistive robotics," in *ISER-13th International Symposium on Experimental Robotics-2012*, 2012.

[95] Viola, P. and Jones, M. J., "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[96] Wainer, J., Feil-Seifer, D. J., Shell, D. A., and Mataric, M. J., "The role of physical embodiment in human-robot interaction," in *15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pp. 117–122, IEEE, 2006.

[97] Xavier, J., Pacheco, M., Castro, D., Ruano, A., and Nunes, U., "Fast line, arc/circle and leg detection from laser scan data in a player driver," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp. 3930–3935, IEEE, 2005.

[98] Yan, P. and Bowyer, K. W., "Biometric recognition using 3d ear shape," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 8, pp. 1297–1308, 2007.

[99] Yuan, F., Hanheide, M., Sagerer, G., and others, "Spatial context-aware person-following for a domestic robot," 2008.

[100] Zender, H., Jensfelt, P., and Kruijff, G.-J. M., "Human-and situation-aware people following," in *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, pp. 1131–1136, IEEE, 2007.

[101] Zhang, C. and Zhang, Z., "A survey of recent advances in face detection," tech. rep., Tech. rep., Microsoft Research, 2010.

[102] Zhao, W., Krishnaswamy, A., Chellappa, R., Swets, D. L., and Weng, J., "Discriminant analysis of principal components for face recognition," in *Face Recognition*, pp. 73–85, Springer, 1998.

INDEX

VITA

Perry H. Disdainful was born in an insignificant town whose only claim to fame is that it produced such a fine specimen of a researcher.

People Aware Mobile Robot Navigation

Akansel Cosgun

81 Pages

Directed by Professor Henrik Christensen

This is the abstract that must be turned in as hard copy to the thesis office to meet the UMI requirements. It should *not* be included when submitting your ETD. Comment out the abstract environment before submitting. It is recommended that you simply copy and paste the text you put in the summary environment into this environment. The title, your name, the page count, and your advisor's name will all be generated automatically.