

# 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

<b>DEDICATION</b> . . . . .	iii
<b>PREFACE</b> . . . . .	iv
<b>ACKNOWLEDGEMENTS</b> . . . . .	v
<b>LIST OF TABLES</b> . . . . .	ix
<b>LIST OF FIGURES</b> . . . . .	x
<b>SUMMARY</b> . . . . .	xiii
<b>I INTRODUCTION</b> . . . . .	1
1.1 Background . . . . .	1
1.1.1 Social Spaces . . . . .	1
<b>II MAP ANNOTATION</b> . . . . .	2
2.1 Related Work . . . . .	2
2.2 Semantic Maps . . . . .	2
2.2.1 Waypoints . . . . .	3
2.2.2 Planar Landmarks . . . . .	3
2.2.3 Objects . . . . .	3
2.3 User Interface . . . . .	3
2.4 Pointing Gestures for Human-Robot Interaction . . . . .	3
<b>III NAVIGATION AMONG PEOPLE</b> . . . . .	4
3.1 State-of-the-Art Approach in Autonomous Navigation . . . . .	5
3.2 Related Work . . . . .	7
3.2.1 Socially Acceptable Path Planning . . . . .	7
3.2.2 Learning Navigation from Human Behavior . . . . .	8
3.2.3 Human Cooperation in Robot Navigation . . . . .	8
3.3 Goal Points for Navigation . . . . .	9
3.3.1 Labeled Waypoints: . . . . .	9

3.3.2	Labeled Planar Landmarks:	9
3.3.3	Labeled Objects:	11
3.4	People Aware Navigation	11
3.4.1	Global Planner	11
3.4.2	Local Planner	16
3.4.3	Results	17
3.5	Speed Limits for Safe Navigation	19
<b>IV</b>	<b>MULTIMODAL PERSON DETECTION AND TRACKING</b>	<b>23</b>
4.1	Related Work	24
4.2	Person Detection	26
4.2.1	Leg Detection	26
4.2.2	Torso Detection	31
4.3	Person State Estimation	35
4.4	Face Recognition	38
<b>V</b>	<b>PERSON FOLLOWING</b>	<b>41</b>
5.1	Related Work	41
5.2	Basic Person Following	41
5.3	Situation Aware Person Following	41
5.3.1	Door Passing	41
5.3.2	User Activity Awareness	41
5.3.3	Corners	41
5.4	Application To Telepresence Robots	41
<b>VI</b>	<b>PERSON GUIDANCE</b>	<b>42</b>
6.1	Related Work	42
6.2	Guide Robot	42
6.3	Application To Blind Users	42
<b>VII</b>	<b>CONCLUSION</b>	<b>43</b>

APPENDIX A — QR CODE BASED LOCATION INITIALIZATION . . . . .	44
APPENDIX B — ASSISTED REMOTE CONTROL . . . . .	45
APPENDIX C — VIBRATION PATTERN ANALYSIS FOR HAPTIC BELTS . . . . .	46
REFERENCES . . . . .	47
INDEX . . . . .	53
VITA . . . . .	54

## LIST OF TABLES

1	Table shows average and standard deviations of geometric leg features calculated in our dataset. . . . .	29
2	Table shows average and standard deviations of geometric features for a human torso in laser scans. . . . .	33
3	Average orientation error of the torso detector with respect to distance from sensor and body pose in a study with 23 people . . . . .	34

## LIST OF FIGURES

1	Caption . . . . .	6
2	Top down point cloud view of a room. A planar landmark with label <i>Table</i> has previously been annotated by a user. The convex hull for the planar landmark is shown in red lines. When asked to navigate to <i>Table</i> , the robot calculates a goal pose, which is shown as the yellow point. . . . .	10
3	Top down point cloud view of a hallway. The user has previously annotated two planar landmarks with the same label, <i>Hallway</i> . When asked to navigate to <i>Hallway</i> , the robot calculates a goal pose, which is shown as the yellow point. . . . .	11
4	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. . . . .	14
5	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. . . . .	15
6	”Room Problem” revisited. The resulting static plan and dynamic plans are shown in green and orange, respectively. . . . .	17
7	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. . .	20

8	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. The robot has entered the group region, therefore the velocity is halved. 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. . . . .	21
9	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. . . . .	22
10	Circularity criterion in a perfect circle is: $ P_0P_n d_{mid} = 0.5$ . . . . .	28
11	Circularity criterion in a this laser segment is: $ P_0P_{10} /d_{mid}$ . . . . .	28
12	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. . . . .	28
13	Two person detections are seen in this figure. Our leg segment association algorithm propagates pixels vertically from candidate leg segments and connects leg pairs. . . . .	30
14	Flow chart for determining if two leg segment candidates belong to a person. . . . .	31
15	Our torso detector fits and ellipse to the human torso and estimate its position and orientation. . . . .	32

16	Torso detection rate vs weighed Mahalanobis Distance Threshold in our dataset . . . . .	34
17	Experimental setup for the evaluation study of the Torso Detector. . . . .	35
18	Example results of our person recognition method is shown in the image. We use <i>Eigenfaces</i> face recognition method and optionally shirt color recognition. . . . .	39

## **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 [31], 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 [18]. 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 [22]’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. Hüttenrauch [20] 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 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.

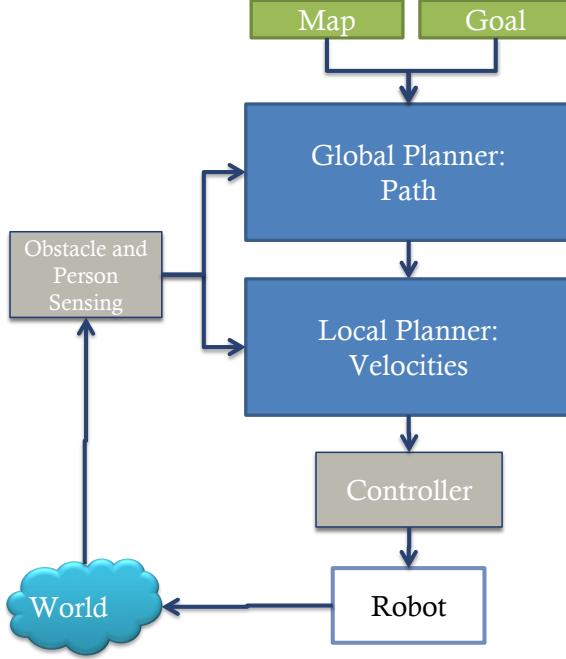
### **3.1 State-of-the-Art Approach in Autonomous Navigation**

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  $(x_0, y_0, \theta_0)$  to a goal pose  $(x_g, y_g, \theta_g)$ . In real-time operation,  $(x_0, y_0, \theta_0)$  is the robot's current pose as the robot tries to reach to the goal pose from where it currently is.  $\theta_g$  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 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 [54], approaching people [48] and evacuating buildings [41]. Some works used the personal space concept in cost-based general path planners [54, 25]. Sisbot [54] 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 [25] 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 [57]. Rios-Martinez [45] 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 [62]. 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 [42]. 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 [34] 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 [47]. One way to simulate human navigation behavior is to use social cost maps that capture social conventions [49, 35]. Contrary to the imitation approach, [7] tries to avoid predicted paths, with the goal to minimize the risk of interference. Kuderer [30] presents a tele-operated robot that computes the policy of a desired interactive navigation by learning from observations of pedestrians. Pellegrini [43] 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 [56, 39].

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 [33] 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 [17], humans observers watched two ways of person following. People found direction-following more natural than exact path following. Kruse [28] 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 [59] introduces the ‘freezing problem’, where traditional path planners fail to produce a feasible solution in crowded human environments. Muller [39] briefly mentions a ‘shooing away’ behavior, where the robot accelerates towards a human, hoping that he/she will get out of the way. Kruse [29] 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:**

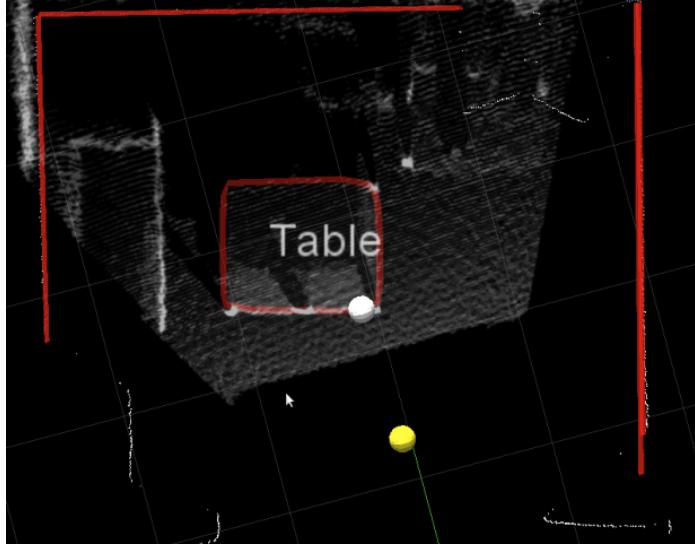
If a waypoint is labeled and saved, the robot attaches that label to the explicit coordinates, namely position and orientation. Therefore, if the robot is instructed to navigate to a labeled waypoint, then the goal is readily the pose of the waypoint.

#### **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.



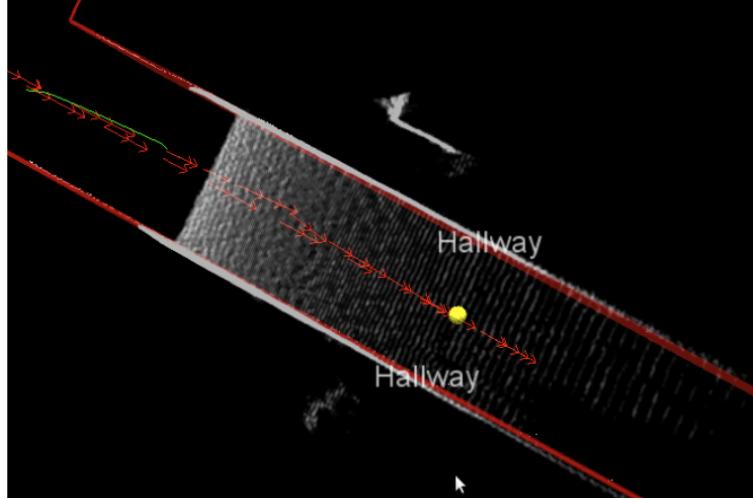
**Figure 2:** Top down point cloud view of a room. A planar landmark with label *Table* has previously been annotated by a user. The convex hull for the planar landmark is shown in red lines. When asked to navigate to *Table*, the robot calculates a goal pose, which is shown as the yellow point.

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

People Aware Navigation

### 3.4.1 Global Planner

#### 3.4.1.1 Static Planner

Static 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:

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

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:

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

**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.

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

where  $d$  is the distance to the closest human,  $H$  is all humans,  $\mu_h = (|x - h.x|, |y - h.y|)$  and  $\Sigma = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix}$  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 4(a).

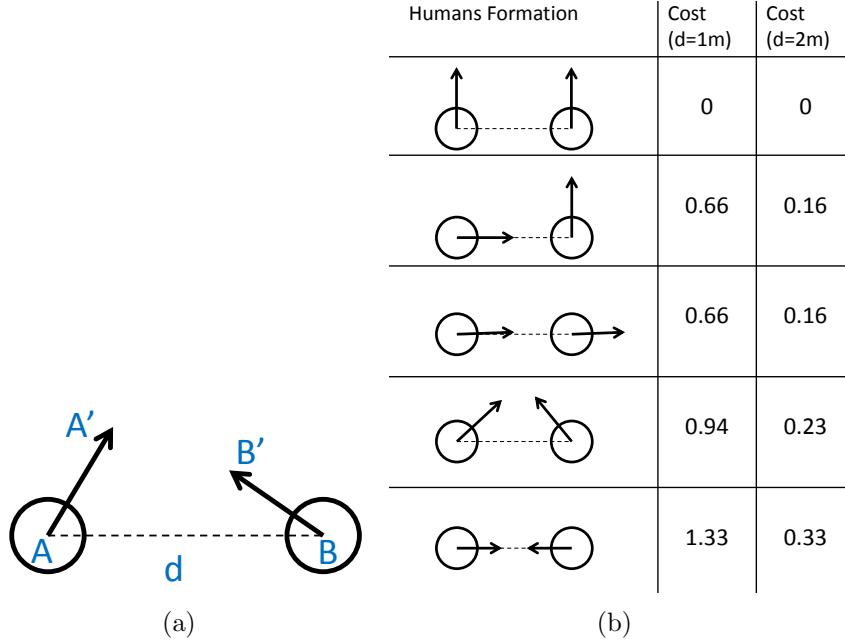
The disturbance cost is calculated as:

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

where all the vectors are normalized and  $d_{max}$  is the maximum distance between the humans that returns a disturbance cost. Figure 4(b) illustrates several examples of disturbance costs with  $d_{max} = 3$  meters.

**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 minimum cost static path.

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



**Figure 4:** 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.

#### 3.4.1.2 Dynamic Planner

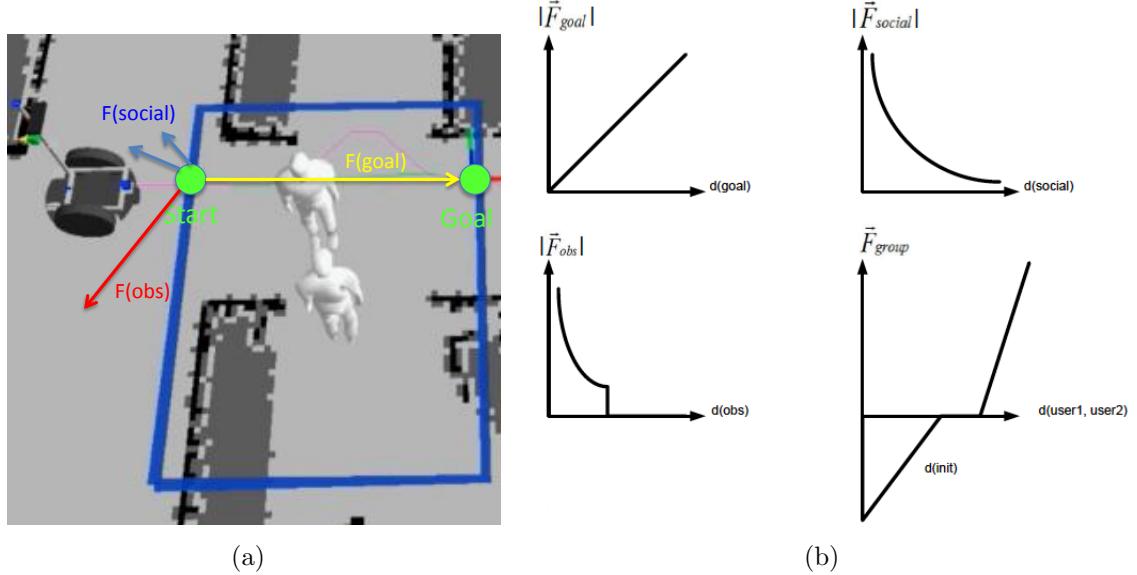
Dynamic planner is responsible for reasoning about how people will react to robot motions. Moreover, paths from static plans are not smooth, therefore robot motion might not be easy to understand for human observers. Dynamic planner receives the static plan and simulates only parts of it where humans are close. We use Social Forces Model (SFM) [?] 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 for robot navigation. The solution to the goal is found iteratively, as forces are recomputed for every robot position during planning.

First, groups of people are found on floor plane 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. Dynamic planner receives the static plan and finds out where it enters and exits each group region if it intersects the region. Goal of the dynamic planner

is to find a sub-plan between those two points. Forces apply to all agents, including the robot and humans. We defined 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 dynamic planning are illustrated on the robot in Figure 5(a). The force magnitudes with respect to distances between entities are plotted in 5(b).



**Figure 5:** 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.

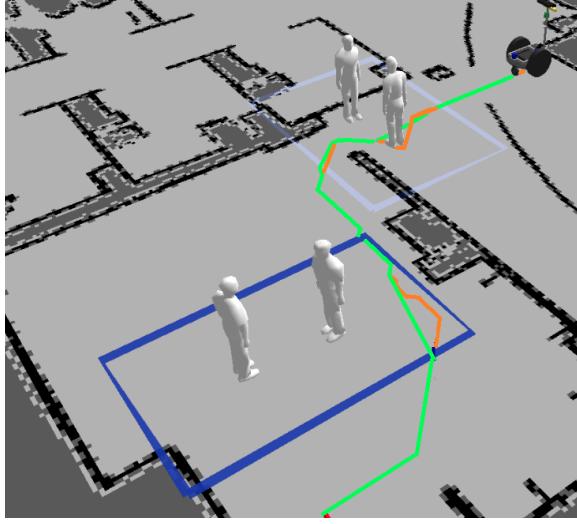
Starting from the first group region that intersects the static plan, the following procedure is applied within every group region: At every iteration, first the resultant

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 local minima [?], 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 has walks past and headed the other way, the social effect is zero.

### 3.4.2 Local Planner

The trajectory planner is responsible of computing the linear and angular velocity necessary to follow the dynamic path. We use Dynamic Window Approach (DWA) [?] for obstacle avoidance of non-holonomic robots. (TODO: DWA our version, derive motion model, vs. ) DWA samples linear and angular robot velocities and simulates the robot trajectory for each of them. First, a waypoint on the dynamic path is chosen as a sub-goal. At every control iteration, the waypoint 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. Robot is simulated with each velocity combination and the trajectory that gets closest to the sub-goal is sent to the motor controllers.



**Figure 6:** "Room Problem" revisited. The resulting static plan and dynamic plans are shown in green and orange, respectively.

### 3.4.3 Results

#### 3.4.3.1 Simulation

**Room Problem:** In this scenario, the robot is outside the room and a point inside the room is given as the goal (Figure 6). 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 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 7). 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 7(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 7(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 7(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

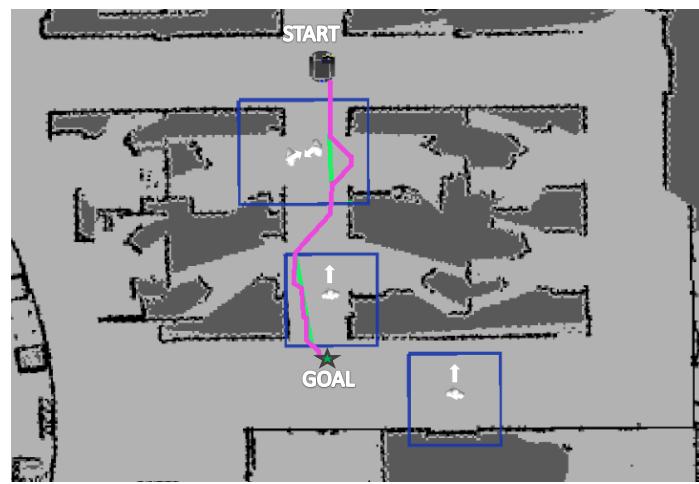
We demonstrate our anticipatory navigation planner on the real system in two environments: hallway and kitchen. Each scenario is run twice under different human positions and behaviors in order to show how the planner responds.

**Hallway passing:** In this scenario (Figure 8), 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.

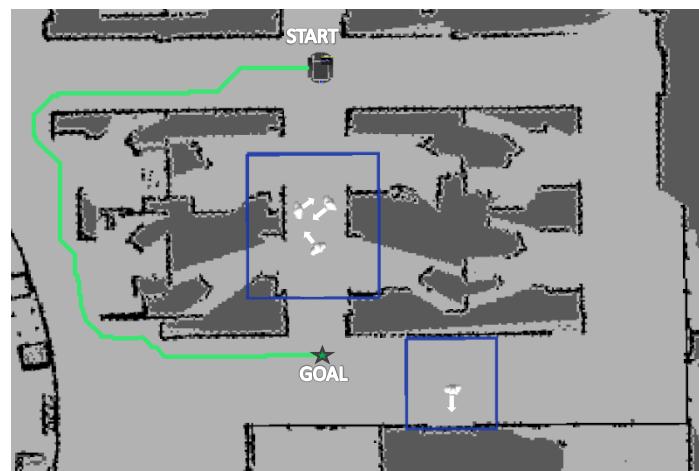
**Narrow corridor:** In this scenario (Figure 9), robot's goal is to drive towards the exit door. There are 3 people nearby the robot. The robot can either take the shorter route that is the direct path, or take a longer path that is to the left of the table. Each important step is described in the caption 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***

Speed Maps for Safer Navigation



(a)

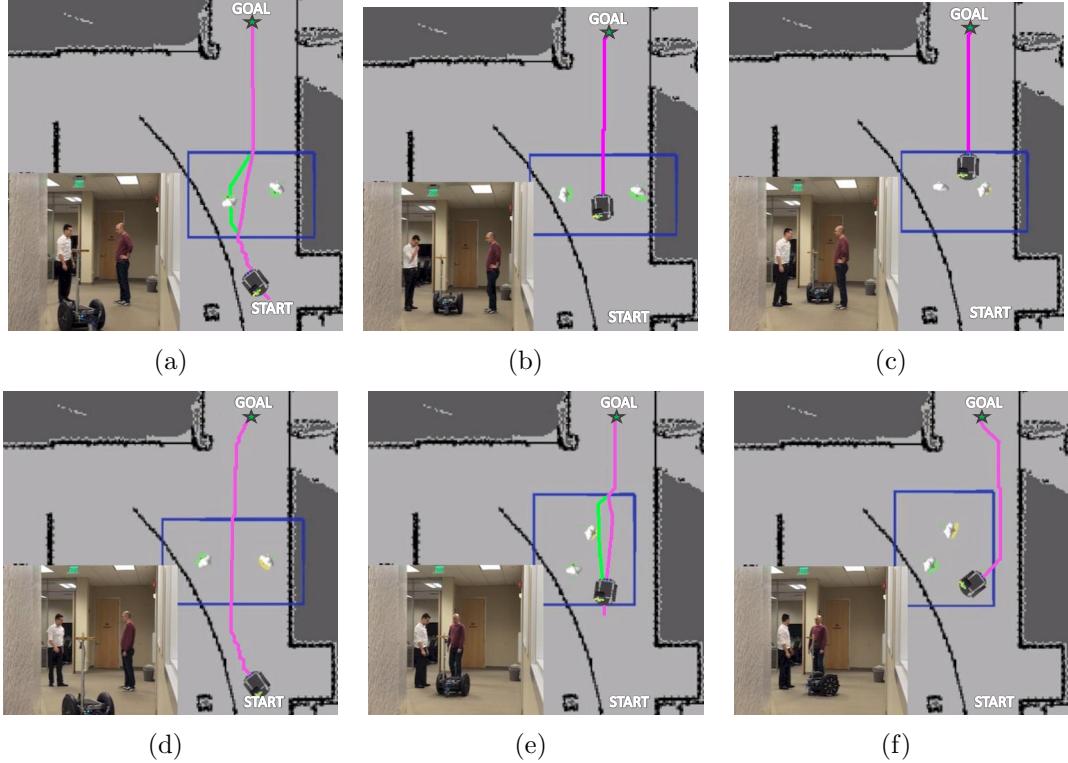


(b)

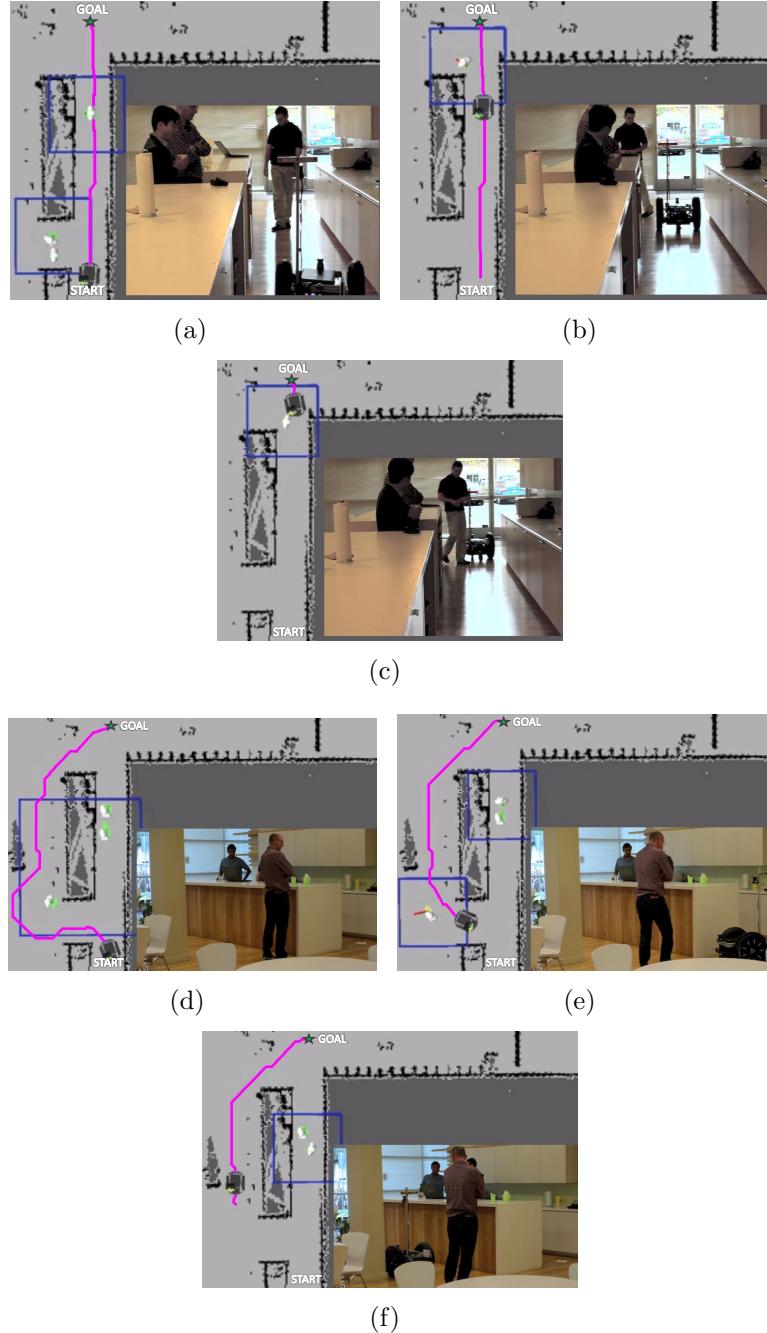


(c)

**Figure 7:** 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.



**Figure 8:** 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. The robot has entered the group region, therefore the velocity is halved. 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.



**Figure 9:** 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.

## 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  $\theta$  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  $270^\circ$  Field of View (FOV),  $0.25^\circ$  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 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 [37]. Face detection is a common method for detecting people, with the work of Viola and Jones [63] being the most popular one. See Zhang [66] for a survey on contemporary approaches on vision based face detection. Another popular topic has been pedestrian detection in crowded scenes Leibe [32] and Tuzel [61].

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 [38] and Schulz [50]

focused on tracking multiple legs using particle filters. Legs are typically distinguished in laser scans using geometric features such as arcs [64] and boosting can be used to train a classifier on a multitude of features [2]. Topp [58] 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 [10], which uses a second laser scanner at torso level. Glas [16] 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 [27] 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 [15].

Laser-based person methods pertain tracking of humans in 2D, projected to floor plane. Tracking of the body parts has long been a topic of interest in vision [5, 52]. With the recent introduction of 3D sensors such as the Velodyne, Swissranger and Kinect, more robust tracking became possible. Spinello [55] trains geometrical features at different height levels in the 3D point cloud for pedestrian detection. Ganapathi [13] estimates body part locations with a probabilistic model. One of the well-known skeleton tracking algorithms is the Microsoft Kinect SDK by Shotton [51], 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 [9].

Face recognition is a widely used application as surveyed by Phillips [44]. 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 [60]. [67] 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 [26] and body motion characteristics [40]. Biometric person identification techniques, such speaker recognition [24], 3D ear shape [65] and multi-modal cues [14] 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 [36], 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:

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

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

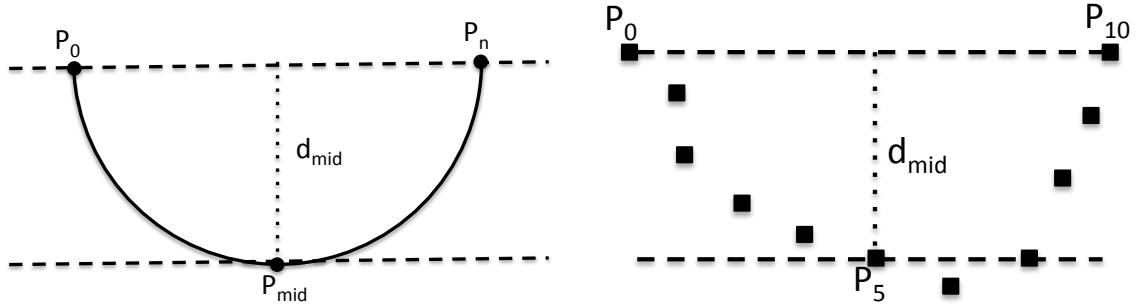
Then we apply segmentation, Segmentation produces clusters of consecutive scan points, which due to their proximity, have a high likelihood of belonging the same object. Two adjacent distance measurements are considered to be in the same segment if the Euclidean distance between them is below a threshold 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 [58]. 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 10, the circularity criterion is  $|\overline{P_0P_n}|/d_{mid} = 0.5$ . In case of a laser scan, as can be seen in Figure 11, 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 [64], 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 12. As a geometric



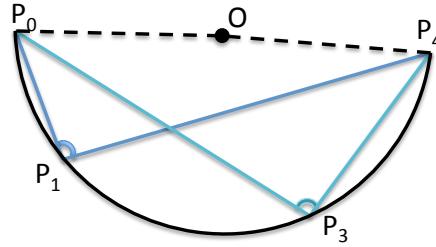
**Figure 10:** Circularity criterion in a perfect circle is:  $|P_0P_n|d_{mid} = 0.5$

**Figure 11:** 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,  $IAV_S = 90^\circ$ . 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 12:** 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( <i>m</i> )		Circularity		IAV( <i>radians</i> )	
	$\mu$	$\sigma$	$\mu$	$\sigma$	$\mu$	$\sigma$
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  $17 \times 10^3$  Single Leg patterns and  $0.6 \times 10^3$  person-wide blobs were manually labeled in the data. In addition,  $120 \times 10^3$  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.

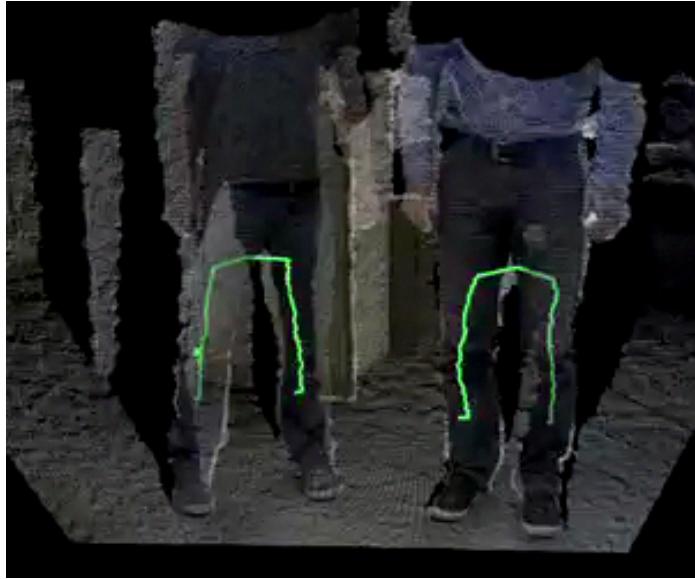
For every segment  $S_i$  in a test laser scan, we first extract the geometric features  $f_1^i, f_2^i, f_3^i$ . We then calculate the weighted Mahalanobis distance to the average leg parameters for each leg pattern:

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

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  $D_{mah}^i < Threshold_{leg}$ , 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.

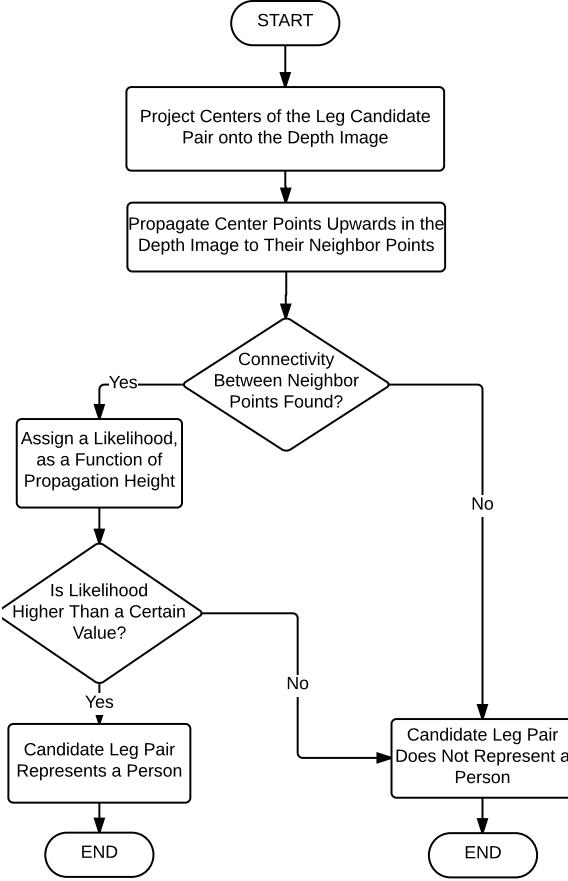
#### 4.2.1.1 Associating Leg Segments

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 13 as an example result. Figure 14 shows the flow chart of the association algorithm.



**Figure 13:** 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



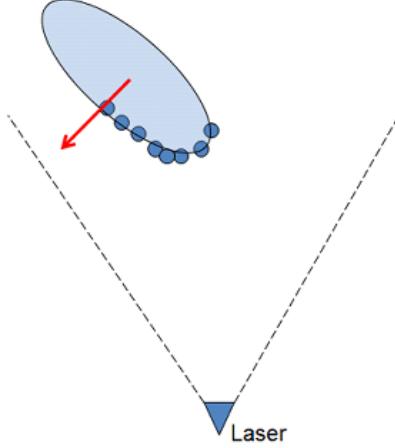
**Figure 14:** Flow chart for determining if two leg segment candidates belong to a person.

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 15). 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 15:** Our torso detector fits an 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 [12]. 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. We use the minor and major axis lengths, as well as the three geometric features introduced in Section 4.2.1 in order to determine if the laser segment is a person. We gathered a 450 laser scans 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

Torso Features	$\mu$	$\sigma$
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.

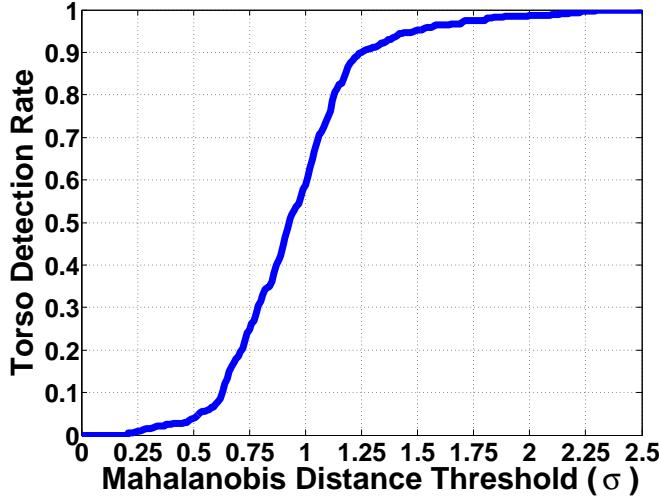
distance in Equation 1 to the averaged parameters. If  $Dmah^i_{torso} < Threshold_{torso}$ , the segment is considered a detection. The feature weight constants we used was  $\mathbf{W}_{torso} = (0.19, 0.09, 0.35, 0.24, 0.13)$ , in respective order given in Table 2. These values were empirically determined, although one can do more sophisticated analysis for optimal weights.

Figure 16 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 each target. Experimental setup from the sensor’s view is shown in Figure 17. 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.

Table 3 shows the angular error at every target distance and human orientation



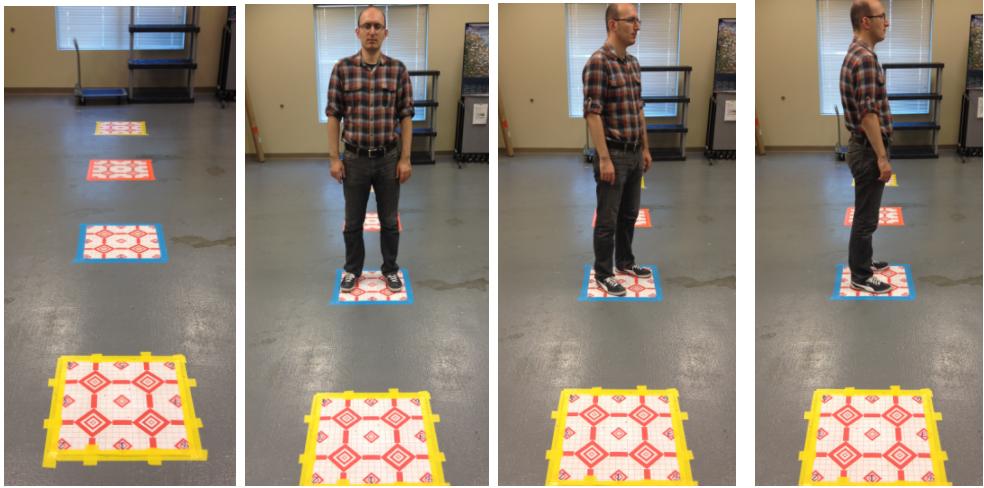
**Figure 16:** Torso detection rate vs weighed Mahalanobis Distance Threshold in our dataset

with respect to the laser scanner.

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

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



**Figure 17:** Experimental setup for the evaluation study of the Torso Detector.

sensor ( $4^\circ$ ) or the opposite way ( $6^\circ$ ). On the other hand, average orientation error was  $24^\circ - 27^\circ$  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 [21] to predict the position of a person. There are other types of filtering techniques available in the literature, such as Particle Filters [23]. 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.

According to Hicheur [19], 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:

$$s_k = As_{k-1} + Bu_{k-1} + w \quad (2)$$

$$z_k = Hs_k + v \quad (3)$$

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:

$$s_k = \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} \quad (4)$$

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:

$$x_k = x_{k-1} + \dot{x}_{k-1}\Delta t_k + w \quad (5)$$

$$y_k = y_{k-1} + \dot{y}_{k-1}\Delta t_k + w \quad (6)$$

$$\dot{x}_k = \dot{x}_{k-1} \quad (7)$$

$$\dot{y}_k = \dot{y}_{k-1} \quad (8)$$

resulting in the following Kalman Filter matrices:

$$A = \begin{bmatrix} 1 & 0 & \Delta t_k & 0 \\ 0 & 1 & 0 & \Delta t_k \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (9)$$

where  $\Delta t_k$  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  $\Delta t_k$  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:

$$Q = qI_4 \quad R = rI_2 \quad (10)$$

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 if it 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.

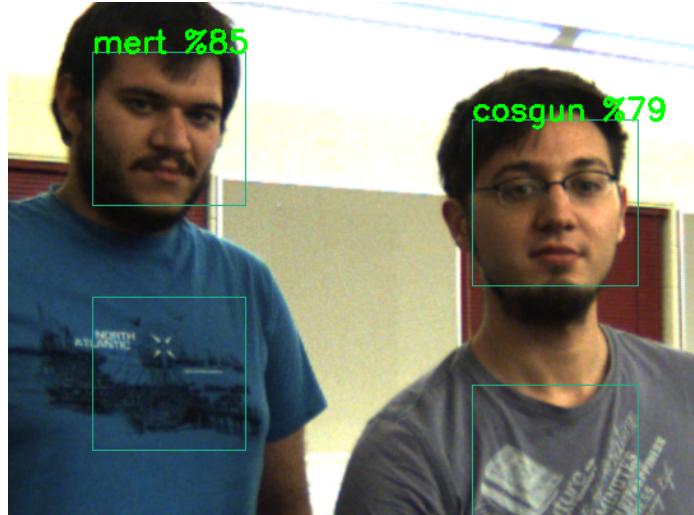
Depending on the task, a single person or multiple people must be tracked. We examine each case below:

- **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 [63]. We use the Eigenface method by Turk and Penland [60] 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 [53] 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 18:** 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 18.

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 [46]. 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

Person Following

### ***5.1 Related Work***

Related Work

### ***5.2 Basic Person Following***

Basic Person Following

### ***5.3 Situation Aware Person Following***

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***

Application To Telepresence Robots

## CHAPTER VI

### PERSON GUIDANCE

Person Guidance

#### ***6.1 Related Work***

Related Work

#### ***6.2 Guide Robot***

Guide Robot

#### ***6.3 Application To Blind Users***

Application To Blind Users

## **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 *Human-Robot Interaction (HRI), 2013 8th ACM/IEEE International Conference on*, 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] 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.
- [10] 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.
- [11] 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.

- [12] 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.
- [13] 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.
- [14] 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.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] HALL, E. T., *The hidden dimension*. Anchor Books, 1966.
- [19] 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.
- [20] 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.
- [21] KALMAN, R. E., “A new approach to linear filtering and prediction problems,” *Journal of Fluids Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
- [22] KENDON, A., *Conducting interaction: Patterns of behavior in focused encounters*, vol. 7. CUP Archive, 1990.
- [23] 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.

- [24] 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.
- [25] 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.
- [26] 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.
- [27] 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.
- [28] 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.
- [29] 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.
- [30] 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.
- [31] LAM, C.-P., CHOU, C.-T., CHIANG, K.-H., and FU, L.-C., “Human-centered robot navigation towards a harmoniously human–robot coexisting environment,” *IEEE Transactions on Robotics*, vol. 27, no. 1, pp. 99–112, 2011.
- [32] 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.
- [33] 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.
- [34] 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.

- [35] 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.
- [36] MITZEL, D. and LEIBE, B., “Close-range human detection for head-mounted cameras,” in *British Machine Vision Conference (BMVC)*, 2012.
- [37] 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.
- [38] 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.
- [39] 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.
- [40] 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.
- [41] 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.
- [42] 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.
- [43] 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.
- [44] 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.
- [45] 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.

- [46] 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.
- [47] 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.
- [48] 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.
- [49] 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.
- [50] 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.
- [51] 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.
- [52] 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.
- [53] SIROVICH, L. and KIRBY, M., “Low-dimensional procedure for the characterization of human faces,” *JOSA A*, vol. 4, no. 3, pp. 519–524, 1987.
- [54] 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.
- [55] 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.
- [56] 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.
- [57] 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.

- [58] 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.
- [59] 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.
- [60] 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.
- [61] 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.
- [62] 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.
- [63] VIOLA, P. and JONES, M. J., “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
- [64] 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.
- [65] 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.
- [66] ZHANG, C. and ZHANG, Z., “A survey of recent advances in face detection,” tech. rep., Tech. rep., Microsoft Research, 2010.
- [67] 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

54 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.