Human aware navigation and reaction for a mobile service robot with a LIDAR unit

Vetle Kind Refsum

Thesis submitted for the degree of
Master in Robotics and Intelligent Systems
60 credits

Department of Informatics
Faculty of mathematics and natural sciences

UNIVERSITY OF OSLO

Spring 2019

Human aware navigation and reaction for a mobile service robot with a LIDAR unit

Vetle Kind Refsum

Human aware navigation and reaction for a mobile service robot with a LIDAR unit

http://www.duo.uio.no/

Printed: Reprosentralen, University of Oslo

Abstract

This thesis is about making a framework for a robot with a mobile base, where the only external sensor is a LIDAR unit. The framework is made to be expanded upon, but to also be switched out or used for other tasks with its modular design. In this work, the users' preferences are also taken into account. The main focus is the distance from a person and the location of the person in the house relative to the robot. To achieve this a general questionnaire was formulated for the purpose. The Godspeed questionnaire was also used to see what perception the participants had of the robot. For the robot to make use of this it has to know where a person is. A classifier using the LIDAR was implemented. To support the classifier a tracker and monitor where further functionality can be given, where implemented. Due to a lack of participants, the results from the questionnaires are inconclusive. However, improvements to be made in future work have been noted. A functioning framework was implemented and tested. However, problems with accurate classification propagated to dependent operations in the robot.

Acknowledgement

I would first like to thank my supervisor Weria Khaksar for good guidance and counsellings. The discussions and follow up to improvement regarding the thesis. Helping set up the questionnaire and conduction the interview with senior citizens.

I would like to thank Jim Tørresen, for taking the time to read through an earlier draft of the thesis. And giving guidelines on what could be done better.

I would like to thank fellow master student Veronika Lappyova for her help in various experiments. We also had good discussions on how to perform the experiments.

I would like to thank the members of the Multimodal Elderly Care Systems project for good discussions surrounding this project.

I would also like to thank the various people taking part in the data gathering processes. at Ifi and Kampen Omsorg+. The staff at Kampen Omsorg+ was also of tremendous help while interviewing senior citizens.

Finally, I would like to thank my family and friends for their support throughout this project.

Vetle Kind Refsum
University of Oslo
May 2019

Contents

List of Figures

List of Tables

List of Acronyms

GQS Godspeed Questionnaire Series. 34

HRI Human-Robot Interaction. 37

LIDAR Light Detection and Ranging. 1

MECS Multimodal Elderly Care Systems. 1

ROS The Robot Operating System. 15

SLAM simultaneous localization and mapping. 21

SVDD Support Vector Data Description. 72

SVM Support-Vector Machines. 89

The page number refers to the first use of the acronym.

Chapter 1

Introduction

The standards of living get better, people live longer and healthier. However, this means that there will collectively be more people in retirement for a longer period of time. Some of these people end up staying at a care center, but many choose to stay at home. For many, this is a way to keep being self-reliant and independent, as well as personal freedom. For our elderly, we always want the best for them, but health-related issues may come in the way of this. To help them, in pursuit of their wish, there are many devices made specifically for this purpose. Devices like the panic button or a security camera are examples of this. Using a camera raises ethical and privacy concerns, while the elderly person may not be able to utilize the panic button in the time of need. This thesis looks at the implementation of a self-contained system, that takes the users personal space into consideration. The system will incorporate a resource-constrained robot and the Light Detection and Ranging (LIDAR) unit that acts as the vision of the robot. As part of a larger project, the Multimodal Elderly Care Systems (MECS) project, this piece will focus on user interaction and awareness. The thesis will also show a method to implement such a system and it' s performance.

1.1 Motivation

For many, an elderly care center is seen as the removal of freedom and self-dependence. This among other reasons is why many prefer to stay at home for as long as possible. With the way things are currently in Norway, if you are eligible, you will get care at home. There are many who take this offer, and this proportion accounts for 85% of those eligible for health care[27]. This trend is also expected to continue in the future, whereby the year 2050 we are expected to see a doubling in the elderly population [28]. For these people who are in their golden age of life, it is expected that we recognize their choice to stay at home. This, however, may have its problems, if we are not to actively monitor people.

Some of the challenges with at home health care is that the caretakers are only available for so much time of the day. As there are many elderly who seek to stay at home and only so many people to take care of them, caretakers have multiple people to take care of per day. Once the caretakers have left there are people who may not have easy access to help, if anything should go wrong. Technology that is designed to help in the instances where things go wrong, may be viewed as imposing and threatening. Implementations made to help you may be of no use as you cannot reach them, or be able to utilize them.

One way the MECS project aims to mitigate these problems is with a home robot. It falls to this thesis to make a foundation for the robot, which can be utilized by others. By using a robot, there is an opportunity to get rid of the traditional ways of monitoring people. The robot is meant to be a hands-free device that is to call for help if need be. To do this, the purpose of the robot is to act as a sort of house companion that takes into consideration the personal space of the person.

1.2 Problem statement

In today's society, there are many senior citizens who wish to spend their time at home, and not in a care center. There are many devices made to make living at home easier for a senior citizen, but these devices may be intrusive or hard to operate when a situation may arise. Products such as a surveillance camera or a panic alarm are employed, to help take care of them. However, there are issues with these implementations, such as privacy concerns and availability. The camera may capture sensitive information, or you may not be able to press the panic alarm due to health-related causes. For a senior citizen to stay at home, it would instead be beneficiary to have a device that could fill in for some roles that a caretaker would do. This device should also discreet in its operation, to make its presence seem as natural as possible. For a device to fulfill this role, there is a need to look away from conventional methods, e.g a camera or a panic button. With the use of a robot to actively cater for the person in the place of camera or panic alarm, one can avoid the problems of the other methods, as the robot can be a self-governing and independent system.

There are currently many robotic systems that use a multi-sensor setup, however, these sensors may seem intrusive when in a private setting. To avoid the implications that come with a camera this research will be using a LIDAR to detect humans. The main problem here is how you identify a person with a LIDAR unit. What do you have to do to separate what is a person and what is background? And are there features distinct enough to do this separation?

You have now detected the person, but if it is not reliable what do you do. The apartment may also provide an opportunity to obfuscate the person. By objects being in the way or because of the apartment layout. We need a way to keep track of the person that relies on detection but can operate for a time without it. So how do you keep track of the person, while the person is moving? More importantly, how do you keep track of a person when both parties are moving around?

If the robot is to be around the user for an extended period of time, it could be perceived as an annoyance if the robot is not aware of the user. As the robot is going to be in the private setting of the user, it should be aware of the users' personal space. The robots interference in this space may make the presences of the robot a worse experience than the help the robot can provide. Users may see the actions of the robot as intimidating or frightening in a sense if there is no mutual respect. To come around this problem, there must be a sense of respect for privacy between the robot and the human. How can the robot be made to not be an element of disturbance? And is, what is considered a disturbance, a matter of individual opinion? What can be done to get an idea of what the expected user group wants?

The positioning of the robot can come in the way of the daily activities of the user. At home, there is always something to do, whether it is to cook a meal or do the laundry or to clean the house. Humans do move around when they are at home and the movement of the robot may come into conflict with the path of the human. Even though the robot is trying to respect the personal space of the user, due to the movement of the user or the environment, this may not happen. The robot may be seen as an interference in the daily activities at home and could be seen as a disturbance. How can the robot take into consideration the behavior in different areas of the house? Are there spaces that the robot should completely avoid? Or can the robot be present in an area but in an out of sight out of mind way? Lastly, how do these spaces affect the robots ability to navigate?

1.3 Objective

This thesis will focus on how to navigate a robot in a home environment. On part will focus on person detection and social awareness. While the other part will focus on social-aware navigation and collision detection.

The implementation side of the project will focus mostly on motion planning and human aware navigation. To find out what control mechanisms that can realistically be implemented, how these mechanisms behave. These solutions should be weighted in terms of the pros and cons of what they bring. Finally, there should be a conclusion to what works the best, from the data gathered. To do this we will be using a computer simulation of a Turtlebot and an environment resembling what the real Turtlebot will be used in. The system will then be implemented on a real Turtlebot and tested in the motion capture lab at Ifi. Finally, the robot will be taken to an environment that is equivalent or equal to that of an elderly persons home to see what has come from the research.

As the concept and implementation of a caring unit in a personal home may be intrusive, we want the robot to take into consideration the preferences of the involved parties. Making the best of the abilities of the robot without interfering with the day to day lives of the person under its care. This means respecting social norms of comfort distance, and not needlessly be around if the party does not prefer it to be so. Therefore a small survey will be carried out to make a model of the persons' area of comfort so that the robot will not be intrusive. A person's area of comfort may differentiate between the different areas in the house. This should also be taken into consideration in the final implementation of the system.

What is expected from this project is to have a robot able to navigate in a complex environment. This entails dealing with chairs, tables, and any other object or item that might be in the way of the robot. The robot should also be able to identify if a person is moving and decide the course of action given the situation. This can be whether to move away from the moving person if the robot is in an area that might be intrusive. If there are multiple people the robot should take into account the personal spaces for each individual.

1.4 Use Cases

The robot is driven by a wheel and can therefore not operate on multiple floor plans. A doorframe may pose a challenge if the robot is not able to cross it. In different situations, the robot and the use of personal space may need to be handled differently. Say for example that a person is sitting on a sofa and spending some quality time in front of the TV. In this situation, it may be better if the robot behaved close to that of a robotic vacuum cleaner and did its own thing. The opposing action to this would be for the robot to remain a certain distance from the person silently monitoring. For many, this would be unnerving and unwelcome. If the robot were to try and resemble something else this may be avoided. This is not to say that the robot should not take care of the person, but some sort of periodic monitoring may be better than active monitoring in certain cases.

This behavior of how the robot respect the personal space of the person in question should also extend to different areas of the house. An example of this is the kitchen. This is often an active area of the house where the robot under normal circumstances may be more in the way than to help. But the kitchen is also a place where people spend a considerable amount of time. Since this is the case the robot should remain in the area, but in a manner that is unobtrusive. There are many ways this can be done in, such as designated areas the robot can be in, or spaces that the robot can move in. However, this should be up to the person in question.

People move around from area to area within the house and in these cases, the robot should behave in the manner of a companion. This is because, as we get older, we will have a gradual reduction in bone mass. If one is to fall at an elderly age, there is a greater risk for bone fracture or other complications [5]. To mitigate this the robot should move with the person when the person is moving. However, the robot should also not make matters worse if the person in question were to fall. The robot should be in a position that is comfortable to the recipient, where it is not a hindrance or a menace. A menace in the sense where the robot is following directly behind the person, a behavior that may be seen as strictly surveillance. To get a better understanding of what people want the robot to do, a survey and/or experiments should be conducted.

The overall function of the robot is to take care of the elderly recipient in the stead of heal care personnel. But if the elderly person is receiving guests, the robot does not need present. As the guest are there if anything should happen, the robot can take the time to charge. However, in these cases the robot may serve as extra insurance if it is present. But how should the robot behave in this case, should it keep to its self or should it stay by the people?

People change mode by the time of day. In the morning a person might be irritable before they have had their daily cup of coffee. By midday, the person might be more forthcoming, and the presence of the robot might be welcome. At the end of the day, the person might be tired and the presence of the robot might not be welcome anymore. How does this influence the personal space of the person and how the robot should interact with it? As the robot is meant to spend a large amount of time with the person, an interaction that takes into account the time of day can be favorable. This can avoid needles irritation on the robot and could be better for cohabitation between man and robot.

There are bound to be events outside of normal for the robot and the person. For the robot, a state of normal would be to keep a respectful distance from the elderly person and try to unobtrusively monitor them. However, the person they are monitoring might actively try to get to the robot. Ether to check out the robot or to harm it. The robot is there for the person, so should the robot behave in a self-preserve manner, or not. For the person, an event outside the normal might be activity at odd hours. From time to time you might have a bad night and can not sleep. If the persons' decision is to get out of bed, what should the robot do and how should it respond to the person. The detector that the robot has does not differentiate between night and day. But to have a robot suddenly show up if you are walking around in the dark might not be the best of experiences. On the other hand, if an unfortunate event should happen and the robot is left to charge the whole night, the person may be on the floor the whole night. So if the robot is to be active at night, how should it interact with the person?

Many of these cases are subject to personal preference. To get a good picture of what people are looking for, there should be carried out a survey where there are uncertainties in what is considered good behavior. Experiments with the robot and senior citizens should also be carried out to get the elderly personnel a feeling for how the robot behaves.

1.5 Context

This thesis is written as a part of the Multimodal Elderly Care Systems (MECS) research project at the University of Oslo. The objective of the MECS project is to Create and evaluate multimodal mobile human supportive systems that are able to sense, learn and predict future events. In this aspect, the project seeks to address the challenges current technology have at handling the complex and different environments found in homes. And the threat to privacy and lack of interpersonal contact technology may be viewed as. This thesis part in the MECS project is in the interaction between robot and human. The outline for the project is how to make a resource conservative robot respect the privacy of an elderly person. At the heart of this project is how the robot approaches a person. Where the robot should place itself in relation to a person. And how the robot should react to an approaching person. Other related questions to this project are. What is considered intrusive space in a personal setting? Is the personal space static or dynamic over the course of a day? Does this space change depending on room setting? Even though some of these examples may not be available depending on the technologies used. They represent the ideas that this project aims to explore.

1.6 Thesis structure

The structure that this thesis follows.

Chapter 2 - Background: Describes the research that this thesis is built upon.

Chapter 3 - Methodology: The first part describes the design aspect behind the thesis and the theory to support it. In the next part, a description of the making of a questionnaire related to personal space and the elderly. And the gathering of data, using the questionnaire, at the retirement complex Kampen Omsorg+.

Chapter 4 - Implementation: Describes how the system is implemented and the design behind the algorithms

Chapter 5 - Results and Experiments: This chapter has two parts. The first part describes the data and results of the questionnaire. In the next part experiments related to the software implemented is conducted and the results are presented.

Chapter 6 - Further works: Lists improvements that can be made and what should be changed for future work.

Chapter 7 - Conclusion: Presents what has been done with a summary of the thesis.

Chapter 2

Background

The implementation will make use of the LIDAR unit on the Turtlebot and multiple onboard software systems. Mainly the odometer which is responsible for the velocity of the robot. And the GMapping unit, which will be responsible to keep track of the robots relation to the local space. The implementation of movement detection will base itself on the current viable methods of identifying a human. This is what will be presented here.

In resent years there have been an increase in the interest surrounding resource constrained service robots and human aware robots. The research from [22], [37], [36], [8], [38] focus on how to get the most amount of data from the least amount of hardware. Except for Talebpour et al.[38] which uses a depth camera, every system uses a 2D LIDAR to detect the legs of a person. There will be a short explanation on how these algorithms work and then there will be a focus on how robots interacts with humans.

Chung et al. propose a leg tracking scheme by taking advantage of the human walking model to achieve robust tracking [8]. With the use of a single LIDAR, they were able to derive the common attributes of legs from a large number of sample data. These attributes were derived by the use of the support vector data descriptor (SVDD), from the distribution of the width and girth, and depth and girth data. This data is then used in an algorithm which looks for clusters of data points that are inside the classification boundary. To track a person they carried out experiments to gather the walking motion. From this, they were able to derive the gap between the legs, the speed of the legs, the orientation and the displacement of the body during walking. By using this data they made two algorithms. The first method makes use of the data from algorithm I, to distinguish between different pairs of legs. The second method predicts the possible next leg position at the next time step, by the use of the displacement and orientation data. In this research, they mainly focus on detection and human tracking, while the social convention is not a major part. They also make a point in saying that the data is gathered from human legs, as such if a person wears a skirt, they may not be detected.

Taipalus and Ahtiainen[37] uses a list of predefined features for classifying the LIDAR data as clusters. These clusters are tracked separately and are evaluated to see if they fulfill a set of predefined conditions. If two clusters are close enough and they fulfill the conditions, it is perceived as human legs and tracked. However, they use a LIDAR with a much higher resolution than what is available in this project. The difference in resolution may affect the accuracy of the algorithm in a negative way. This is due to sparse data samples to work with which, as they say, will make the analysis for the shape of the cluster not as reliable.

In the case of Kim et al.[20] they derive, through a series of experiments, the characteristics of one leg and the relationship between two legs. Via the features belonging human legs, they made an algorithm to remove all other objects that come from a scan. And through the relationship features of two legs, they are able to find a person. The downside to these algorithms, as they point out, is that they do assume that people have visible legs. If say a person is wearing a skirt, these algorithms may not work as intended.

In the paper by Bellotto and Hu[4] a fusion between a camera and a LIDAR is used. Here the camera is used to detect faces, which is then matched to the legs' position. The legs are detected by the LIDAR using pattern recognition. In this implementation, edge detection is used to extract possible patterns. These patterns are then sorted into leg positions of which they describe three. The cases are: two legs are apart, forward straddle and two legs together. They provide the leg detector implementation as a GitHub repository[3]. However, they do point out that this system relies on both sensors for robustness.

After a person is found by the system, it needs to keep track of the person. The object of the tracker is to associate the target person or persons in the next frame of reference. This is done to avoid having to look for the target in each frame of reference, but rather predict where the target will be. To track a person's leg one have to take into consideration, the walking motion of a person and the obfuscation by other objects or other legs.

When trying to track a person with a LIDAR, researchers have adapted a human walking model to work with their data [22], [20], [8]. They combining the human walking model with empirically gathered data on the walking gate from the LIDAR. Through this, they are able to predict the next step that a person takes within a reasonable threshold.

After having established that we have found a moving object and its direction, we have to determine the personal space of the person. By having determined the orientation and heading there will be a better fit for the personal space. This is because not all the space is treated as equal. It is easier to approach a person from the front than it is to approach them from the rear. The personal space will set up areas where the robot is allowed to go and where not to go in relation to the person.

Ferrer et al. presents in their research [10] a social-aware navigation framework for robots accompanying people. For their experiments, they used two identical robots developed in the URUS project [41]. Data from the environment are gathered from two laser range sensors, on in the front of the robot and in the rear. There is also a stereo camera located in the eyes of the robot, meant for computer vision. To make the robot work in a crowded environment they took use of AMCL, a probabilistic localization system for a robot moving in 2D [40]. They implemented the detection algorithm from Arras et al.[1], to detect people and a tracking method from Luber et al.[25]. To navigate the robot makes use of the extended social-force model (ESFM). This model is an extension on the work by Helbing and Molnár[15], where behavioral changes are used in terms of social fields or forces. Social forces are, as described by Helbing and Molnár, are a measure of the internal motivations of the individual to perform certain actions. And it is suggested that these forces can describe the movements of pedestrians. However, the standard social force model does not take into account robots. The extension proposed by Ferrer et al. expands the model to include the interaction between people, objects and robots. This work describes a navigational framework for robots accompanying a robot. As is the framework is designed with the purpose of the robot always accompanying the person. A robot that is always accompanying a person in a personal setting might be seen as rude. If the model is to be used it has to be retrained to better suit the circumstances.

The work of Talebpour et. al introduced the ranger unit [38]. The main focus of the project was the human-aware navigation problem in a structured environment. By the use of an RGB-D camera they were able to perceive and track a person. This was made possible by the use of two algorithms, one based on leg detection by an RGB-D camera and the other by 2D range data. The second method developed by Arras et. al [1], uses AdaBoost to train a classifier from simple features corresponding to legs from the range data. The ranger fused both algorithms by the use of a Kalman filter, where the initialization comes from the first algorithm and is updated with data from both. When the robot is aware of a person, it assigns a cost to the personal space of the person, to which the robot should avoid. From proxemics, the study of human use of space, they make the robot consider people and the environment differently. This is to make the robot plan a route that avoids interfering in the personal space of the person. The person's personal space is here modeled as a two-dimensional Gaussian cost function. This function is centered around each person and the variance is made to be proportional to the relative velocity of the person. The cost data is mapped onto a The Robot Operating System (ROS) costmap layer, this will then affect the path planning of the robot. The focus of the research was on the social awareness of the robot, where the detection was made by the use of a depth camera.

Why are there so few sources? This comes down to the fact that it is a niche field. Although people do make human detecting robots, these are usually done with a standard camera or a depth camera. LIDAR is mostly used as an effective way of measuring the distance to surrounding objects. So in what other instances is LIDAR solely used? LIDAR is at times used for people detection for autonomous cars[19] [33]. However, these LIDAR systems are not bound to a 2D plane. These 3D LIDAR systems have the advantage of the unique signature formed by the curvature of a human. Although some systems are the same, e.g. how the shape is segmented out of the data and how the data is classified. The feature extraction of the shape is not the same. Because of this, although there are few papers, the main focus will lie with 2D LIDARS.

Chapter 3

Hardware,Software and Tools

This chapter presents the hardware and software used for the implementation and experiments of the system. This includes the robot, the ROS environment, and Gazebo the simulation tool.

	Name	Version
Operating System	Ubuntu	16.04
Development Environment	KDevelop	4.7.3
Software framework	ROS	Kinetic
Simulation	Gazebo	7.0.0

3.1 Hardware

3.1.1 Turtlebot3 burger

The Turtlebot3 burger is a small programmable, ROS-based mobile robot. It provides a hardware platform that is modifiable. The Turtlebot3 burger is one of a series of two robots, with the other one being the Turtlebot3 waffle. The difference between the two is that the waffle is lower, but has an overall greater width. The robot has a mobile base driven by two wheels and stabilized by a metal ball. It is fully programmable using Open source, where the included software is under an Apache 2.0 license.

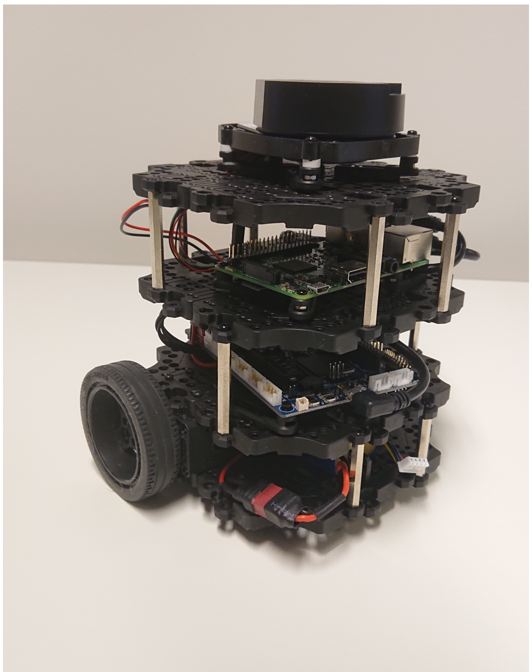


Figure 3.1: The Turtlebot3 burger

Table 3.1: Specification of the turtlebot3 Burger

Maximum Translational Velocity	0.22mm/s
Maximum Rotational Velocity	2.84rad/s162.72deg/s
Maximum Payload	15kg
Size (L x W x H)	138mm x 178mm x 192mm
Weight(+ SBC + Battery + Sensors)	1kg
DYNAMIXEL	XL430-W250-T
SBC	Raspberry Pi 3
Embedded Controller	OpenCR (32-bit ARM®Cortex®-M7)
Sensor	HLS-LFCD2
	3-Axis gyroscope
	3-Axis accelerometer
	3-Axis magnetometer

3.1.2 Hardware limitations

There are certain limitations to be aware of when developing for the Turtelebot3 burger. Due to its only external sensor being the LIDAR unit,

Table 3.2: Specification of the 360 LDS-01 (LIDAR)

Distance Range	120 3,500mm
Distance Accuracy (120mm 499mm)	±15mm
Distance Accuracy(500mm 3,500mm)	±5.0%
Distance Precision(120mm 499mm)	±10mm
Distance Precision(500mm 3,500mm)	±3.5%
Scan Rate	300±10 rpm
Angular Range	360°
Angular Resolution	1°

it has no perception of any object higher or lower than 19.2 cm above the ground. This makes the robot unsuitable in areas that have stairs or a high door frame. Tables and chairs may also be a problem because of the stretcher. When it comes to the LIDAR unit, reflective and transparent surfaces do pose a problem. With reflective surfaces, the laser may bounce off the surface and give back a wrong distance reading. When it comes to transparent surfaces like glass, the laser will go through the object and give a reading of the object behind. Another issue with the LIDAR is the resolution, as narrow objects may not always be picked up. This influences the navigation system on where the robot can travel. As the thin object can appear to disappear and reemerge based on position and distance. The speed of the robot also poses as an issue in this project. As one of the goals is having the robot move away from an oncoming person, may not be possible because of late detection and slow maneuverability.

3.2 Software

This section contains information about the various software tool that was used in this project. What will be described is first the various software used for the development of the robot. Then there will be a description of the software tools that were used for simulation and debugging.

3.2.1 ROS

The Robotic Operating System (ROS) [1] is an open source project, that is the result of a combined international community. The core of ROS is a message passing interface that provides interprocess communication. Otherwise, it provides an extensive set of tool and robot specific libraries. The function of ROS is to be a flexible software framework for robotic development. Here contributors provide packages on subjects that they specialize in. Giving an easier time for people who want to implement a full system. But does not have the know-how on narrow subjects.

Move_base

Move_base[2] is a package for ROS that provides an implementation of an action, that given a goal, would try to move a mobile base there. To accomplish this navigation task it links together two systems, a global planner[3] and a local planner. The purpose of is as its name suggest, to make a route from the robots current position to the goal, via a static map. The global planner is in the default setup and thus uses Dijkstra's algorithm to navigate. On the other hand, the local planner's task is to make the robot take the best route in a local space following the global plan. This does not mean it will always follow the global plan, as it is also responsible for avoiding obstacles that may not be present. Move_base was used for any motion planning task that were issued.

GMapping

GMapping[4] is a method to solve the simultaneous localization and mapping (SLAM) problem. It is based on the Rao-Blackwellized particle filter, where each particle carries a map of the environment. What GMapping does differently is to use adaptive techniques to reduce the number of particles. The end result is a method that drastically decreases the uncertainty of the robots pose in the prediction step of the filter. This gives a faster and better way to compute grid maps from laser range data.

rviz

Rviz[5] is a general purpose tool for visualization of three-dimensional data

[1] http://www.ros.org/
[2] http://wiki.ros.org/move_base
[3] http://wiki.ros.org/global_planner
[4] https://openslam-org.github.io/gmapping.html
[5] http://wiki.ros.org/rviz

for many external robotic sensors. It is a package for ROS and as such many of the common messages in ROS can be visualized by this tool. Among some of the other features are a visualization of the robot model and the position of the robot on a map. This makes it easy to identify possible problems or configuration errors. In this thesis, rviz was used as a debug tool to visualize what the robot detected. It was also used as a simple GUI to describe areas and locations relative to the world frame.

3.2.2 Gazebo simulator

Although originally a tool under the ROS project, it has since become its own thing. Gazebo simulator[6] is a 3D dynamic simulator that provides a robust physics engine. It offers the capability to accurately and efficiently simulate multiple robots in complex indoor and outdoor environments. While also having extensive dynamic interaction between objects. Gazebo is by default compiled with support for the ODE[7] physics-engine, but it also be used with the Bullet[8], Simbody[9] and DART[10] physics engines. Some of the key features of Gazebo are the libraries of robot models and environments. It also provides a wide variety of sensors and interfaces for both users and programs.

[6]http://gazebosim.org/
[7]http://www.ode.org/
[8]https://pybullet.org/wordpress/
[9]https://simtk.org/projects/simbody/
[10]http://dartsim.github.io/

Chapter 4

Methodology

This chapter will first present the design regarding what is to be implemented and why. After this, the theory surrounding the most prominent algorithms used will be presented. In the rest of this chapter, the formulation of the questionnaire will be explained. As will conducting the experiment at Kampen Omsorg+.

4.1 Theory

This section will present the overall design of what has been implemented. It will go into the thought process behind the design and what the design achieves. Next, there will be a comprehensive description of the most promonent algorithms used.

4.1.1 Design

The task is to implement a framework that can detect a person, follow this person and react to the person. As such implementation for this framework has been split into three separate nodes. A node is an individual process that performs computations. The first node is responsible for the detection of a person. The second node, which is dependent on the first, is responsible for tracking and predicting the path of a person. The last node, which is dependent on the previous two, is the monitor of the framework. Each of these nodes, in this implementation, is responsible for their own main objective. The goal of the design was to make it possible to switch out a node for something that did the same, without the problem of interconnected libraries.

The first node has the task of converting range data from the LIDAR, into a list of positions where people are in relation to the robot. To do this, it makes use of a classifier that determines if a cluster of range data is a human leg or not. The design of this implementation is based on the paper by Arras et al.[1]. This was the design decided on because of its use in other implementations [38] [10] [23]. So rather than pushing our luck with implementations whose functionality depend on one paper. The thought was to use something there was confidence in. There is however a difference in the classifier used by Arras et al. and the classifier this project uses. The difference here comes mainly from convenience, as OpenCV [6] has multiple variations of the AdaBoost classifier. Even though they don't have the implementation from the paper, they do have 4 other variants that make for good comparisons. This is also the case for how Linder and Arras[23] implemented their system. Another convenience with choosing this classifier was the SPENCER project, which the paper by Linder and Arras is a result of. The SPENCER project[1] was a research project in the area of robotics funded by the EU. As a consequence of this, a lot of their source code is open to the public. This was made use of to speed up development and to figure out how this implementation should be done. A single module of the SPENCER project was used. This module takes care of calculating the different features that classifier is going to use. Other modules were not used, because of the total redesign that had to be done, should it have worked in this project. It was easier to develop the necessary parts myself, based on the design by Arras et al.

In the tracking module, a standard Kalman filter is utilized. The standard Kalman filter was chosen because of limitations in the scanner used This can be seen in the results section where a test regarding this was conducted6.10. From the result of this, the effective range the robot can identify a person is below a 150cm. This combined with the fact that people have interpersonal distances they don't want to be intruded. And that it is beneficial if the robot is not stepped over because it is to close. One ends up with very limited space for the person to be tracked. It is therefore assumed that the person in this space will most likely hold a linear path. On the

[1]http://www.spencer.eu/

question of whether a walking model should be included or not, it was decided not to. This comes down to the classification results produced6.1. They were inadequate for this method, as such, a standard tracking system was decided upon. Here the tracker is given a position to follow based on the movement of the person.

The monitor is, as the name suggests, the main control center for what the robot does. Its main job is to take position data from the tracker node and determine what sort of action to take. The action taken is based on the position of the person within an area defined previously by the user. As an example, the user wants to make some food in the kitchen. However as the kitchen is a busy area, the robot should not roam around freely. But the user still wants the robot to be in the area. This action will ensure that the robot will be in the area at a predefined location, for as long as the user is in the area. The further purpose of the monitor node is to be a central node of action in the framework. For example, if people intend to add software onto the robot, it should be relegated to a node with a specific purpose. The action that this node wants the robot to do, should then be relegated to the monitor. This promotes a clean design, in which it is easy to navigate and make changes.

Another part of the design was the implementation of tools to aid in the development and to help debug the code. The first of these tools and also the most used is the interactive_data_gatherer. Its purpose is to make it easier to gather data for the classifier. This is done by pointing out each possible person that the robot sees. It is then up to the user of the program to decide what is a person and otherwise. The data gathered is then stored and can later be used for making a model for the classifier. Besides being a tool for data gathering, it is also used for debugging. It makes it quite clear how the robot is interpreting the objects around itself.

The second tool is the debug_gui, and as its name suggests, it is only used for debugging. Its purpose is to show how distant an object is to the robot. This can be done in two frames of references, one from the robot's perspective, and the other from that of the map. The tool was first used in debugging the conversion of frame of reference since the tool does not rely upon the conversion. However, it later found use as a cross-reference tool for implementations that do not warrant a GUI implementation.

The last tool is area_gui, which is responsible for defining areas on the global map. Its main purpose is to help give the robot more user-defined action. This has a twofold application, one is to specify areas of avoidance. An area where we want the robot to avoid at all costs. Then next is an "action" area, which gives a navigation goal to the robot if a person is seen being in the area. The way that this tool operates, is that it gives the user the option to create anything from a dot, a line or an N-dimensional polygon. From this, an area of avoidance or action area can be specified.

4.1.2 AdaBoost classifier

Boosting is the method of training a weak learning algorithm over various reweighed versions of the training data. The classifiers produced by the weak learner are then combined into a single composite classifier. These first boosting algorithms were presented by Freund [11] and Schapire [29]. The adaptive boosting algorithm or AdaBoost described by Freund and Schapire. [12] is the incremental evolution, which improved practicality and ease of implementation. There will first be an overview of AdaBoost (Discrete AdaBoost) as described by Freund and Schapire. After that, there will be a description of what the Gentle AdaBoost algorithm does different.

The AdaBoost algorithm, for a two-class classification setting, takes as input a training set. The training data $(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)$, where $\mathbf{x}_i$ is a feature vector and $\mathbf{y}_i = -1$ or $+1$, for the respective class label. A weak learner, which is, in this case, an unspecified algorithm, is then repeatedly called by the algorithm in a series of rounds. For each round m the weak learner is provided by with a distribution D_m of the training set. The weak learner then computes a classifier f_m whose goal is to minimize the training error, by correctly classify a fraction of the training set. For M rounds this process is repeated, in the end the booster combines the classifiers $f_1, ..., f_M$ into a single classifier $F(\boldsymbol{x}) = \sum_1^M c_m f_m(\boldsymbol{x})$. Here c_m are constants, while $sign(F(\boldsymbol{x}))$ is the prediction of the system. This is sometimes called Discrete AdaBoosting, because the weak learners produces a classification rule $f_m : X \rightarrow \{-1, +1\}$. Here X is the domain of the features $\mathbf{x}$.

An intermediary step between the standard AdaBoost and Gentle AdaBoost is Real AdaBoost. Gentle AdaBoost is similar to Real AdaBoost which is why it is mentioned here. Real AdaBoost[13] uses a tree where each training observation is assigned its weight w_i, rather than weighted resampling.

It also made the weak learner into the tree "stumps", single split trees with only two terminal nodes. Finally it changed what the weak learner returned to a class probability estimate $p_m(x) = \hat{P}_w(y = 1|x) \in [0, 1]$.

From here it is shown that AdaBoost (Discrete and Real) can fit an additive logistic regression model. The criterion for getting an estimation of $F(x)$ is here given as

$$J(F) = E(e^{-yF(x)})$$

where E represents the expectation. Depending on the context, it might be seen as the population expectation. The reason for this criterion is its a differentiable upper bound to misclassification error $1_{[yF<0]}$. For trying to minimize the criterion in this manner, the results are a format change as given by Friedman et al.[13, p. 346]:

1. Given an imperfect $F(x)$, an update $F(x) + f(x)$ is proposed based on the population version of the criterion.

2. The update, which involves population conditional expectations, is imper- fectly approximated for finite data sets by some restricted class of estima- tors, such as averages in terminal nodes of trees.

How the Gentle AdaBoost came about was in the way the criterion function was minimized. The Gentle AdaBoost classifier as described by Friedman et al.[13] uses Newton stepping for minimizing $Ee^{-yF(x)}$. Although quite similar to Real AdaBoost it does have a change in how it updates the functions. The update for how it uses its estimates of the weighted class probabilities is $f_m(x) = P_w(y = 1|x) - P_w(y = -1|x)$. By using Newton stepping it also ends up putting less emphasis on outliers. This is because Newton stepping provide a more reliable and stable ensemble.

Figure 4.1: Gentle AdaBoost algorithem as described by [13]

---

1. Start with weights $\omega_i = 1/N, i = 1, 2, ..., N, F(x) = 0$
2. Repeat for $m = 1, 2, ..., M$:
   (a) Fit the regression function $f_m(x)$ by weighted least-squares of y_i to x_i with weights ω_i.
   (b) Update $F(x) \leftarrow F(x) + f_m(x)$.
   (c) Update $\omega_i \leftarrow \omega_i \exp(-y_i f_m(x_i))$ and renormalize.
3. Output the classifier $sign[F(x)] = sign[\sum_M^{m-1} f_m](x)$

4.1.3 Kalman Filter

The Kalman filter by Kalman[18] is a set of mathematical equations to estimate the state of a process. This is done in a way that minimizes the mean of the squared error, by the way of efficient computational recursive means.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}$$
$$z_k = Hx_k + v_k$$

where the state $x \in \mathbb{R}^n$ and $x \in \mathbb{R}^m$

What the Kalman filter does is to estimate a process given by a model. This is done by using a form of feedback control. It is done in a two-step process, the filter first estimates the process state at some time. This is done by the time update equations, which are responsible for projecting forward in time the current state and error covariance estimates. By doing so we obtain the a priori estimates for the next time step. For the second step, the filter is given feedback in the form of a noisy measurement. This measurement is then incorporated into the a priori estimates, which gives an improved a posteriori estimate.

The update to the Kalman filter can also be visualized as seen in figure 4.2. It is easier to explain this given an analogy. As such, let's say we have a robot that can move in one direction. The robot knows how much it moves, but this measurement is noisy. To compensate for this the robot also have external sensors. In (a) the robots current pose given by the blue line, where

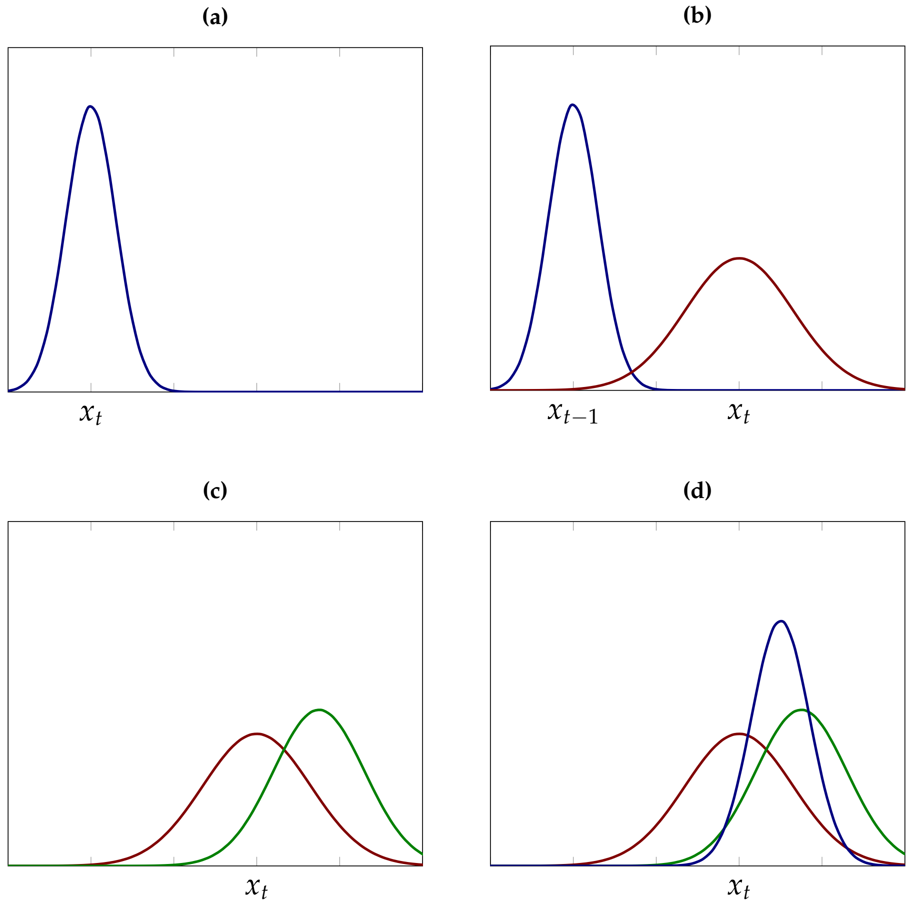


Figure 4.2: Visualization of the Kalman filter process in one dimension

the y-axis marks the probability. The robot moves to a new position and an estimate is given for its current pose, illustrated by the red line in (b). The robot then uses its sensor to get more data about what its current position might be. This is illustrated by the green distribution in (c). The update from the Kalman filter, given these two distributions, is the blue line in (d). This marks the posterior estimate which essentially tries to correct any nose the data given might have.

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1}$$
$$P_k^- = AP_{k-1}A^T + Q$$

The discrete Kalman filter time update equations, given above, describe how the filter goes from one time step to another. Here the state and covariance estimates are updates from time state $k-1$ to k.

$$K_k = P_k^- H^T (H P_k^- H^T + R)^{-1}$$
$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H \hat{x}_k^-)$$
$$P_k = (I - K_k H) P_k^-$$

The discrete Kalman filter measurement update equations, which tries to correct the filter. First, you have the Kalman gain, which determines how much to correct for. Next is the step that gives an a posteriori estimated, given a corrected measurement. Finally, the last equation gives the a posteriori error covariance estimate. Analogue's to the robot example above, the time update equations give the initial estimate for a new position. Then the measurement update equation takes in additional data, corrects the filter, and gives a final estimate.

4.1.4 Winding Number Inclusion

This is an accelerated version of the winding number algorithm developed by Hormann and Agathos[16]. The algorithm determines if a point is inside a nonsimple closed polygon. To determine this it calculates how many times a polygon winds around a point. A point is only determined to be outside of the polygon if the winding number (wn) equals to 0. Meaning that the polygon did not wind around the point. To find the winding number $wn(P, C)$ for a point P inside a closed continuous curve C on the 2D plane, a translation of the curve to a simpler format. This is done by mapping the continuous curve to a unit circle. To do this, C is defined by the points $C(u) = C(x(u), y(u))$, for $0 \leq u \leq 1$, and P is a point not on C. Then to map the curve onto a unit circle you have a vector from P to $C(u)$, $c(P, u) = C(u) - P$, and the unit vector $w(P, u) = \frac{C(P,u)}{|c(P,u)|}$. This gives the continuous function $W(P) : C \to S^1$, where $S^1 = \{x, y | x^2 + y^2 = 1\}$. The mapping can also be presented in polar coordinates as such $W(P)(u) = (\cos\theta(u), \sin\theta(u))$, where $\theta(u)$ is a positive counterclockwise angle in radians. The winding number is then defined as the number of times that $W(P)$ wraps C around S^1.

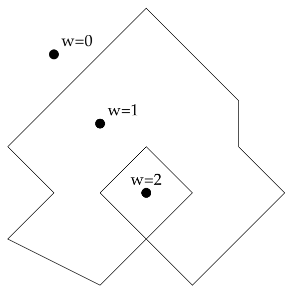


Figure 4.3: Winding number for a polygon

To count this number a point Q can be set on any point in the unit circle S^1. If we wound the curve $W(P)$ and count the number of times passes Q, with a +1 for counterclockwise passes and a -1 otherwise. The accumulated sum you are left with is the number of times the curve wraps around Q To extend this idea into something that can easily be computed, a ray starting at point P and going through Q is drawn. This ray R will cross the curve C in the points corresponding to where $W(C)$ passes Q. Here a distinction needs to be made. If we let R stand in place and move the curve C in a counter-clockwise fashion relative to P. We need to know how C crosses R. If C crosses R in a right to the left manner or in a left to right manner. This can be determined by the sign of the dot product between the normal vector to C and the direction vector q. If C is a polygon this is easier as one just have to determine this once for each edge. This can be made simpler by testing whether an edge's endpoints are above and below the ray R. If the ray crosses the edge, whose start point it to the right of the ray, and the endpoint is to the left, the crossing is positive (+1). Otherwise its a negative (-1). At the end, one adds all these crossings together. If the answer is larger then 0, point P is inside C.

4.1.5 Proxemics

Proxemics is a term coined by Edward T. Hall[14]. The term as described by Hall is defined as «the interrelated observations and theories of man's use of space as a specialized elaboration of culture»[14, p. 1]. Hall describes four levels of interpersonal distances that occur in different situations. First, we have the intimate distance which is between 15 and 45cm. Next, you have the personal distance which is between 45 and 120cm. Beyond this, you have the social distance which is between 1.2 and 3.5m. And finally, you have the public distance which is between 3.5 and 7.5m It should be noted however that these distances are not representative to all humans but to the natives of the northeastern seaboard of the United States. According to Hall, these generalizations are not representative of human behavior in general, as different cultures have different proxemic patterns.

4.2 Questions and hypothesis

To make a framework that takes into account the preference of the user we have to know what the user expect from the framework and the pitfalls to avoid. The question this thesis want insight on relates in a broad sense to distance between the user and the robot. We want to know what the user wants the robot to do when the robot is within a distance the user finds obstructive. Is the robot to back away, stay its ground, or shift location relative to the person but still in the same distance. To get a better understanding of distance we should also have an understanding of the levels of comfortability the user has in different scenarios with the robot. This relates to the visibility of the robot and how the user wants the robot to position itself. The robot may become a part of the day to day lives of the user, but as with anything else some actions from the robot may be viewed as an annoyance. The user might not appreciate the continuous motion of the robot in the house as the robot makes a bit of noise when in operation. There might also be certain times of the day that the operation of the robot is not wanted, as in the morning or evening hours, when people may want time for themselves. In certain situations, the robot may be required to be with the user to oversee actions the user does. This might be if the user is mobility impaired and have difficulty in traversing the house, making the presence of the robot an assurance. However there might be areas that the robot is not welcome to enter or will be in the way of the user, should the robot enter. Another issue is where it is acceptable for the robot to stay while the robot is inactive or not needed.

4.3 what kind of research

To have a general idea of what people want out of the robot qualitatively research is going to be conducted. What we are looking for is large discrepancies in the data. This means that personal preference is involved and that these features of the robot should be easily modifiable. The Godspeed Questionnaire Series (GQS) will also be utilized to get a general idea of the though people have surrounding the robot.

4.3.1 Wizard of Oz

Wizard of Oz is a prototyping method, where the user interacts with the software as it would have been the finished product. But in actual fact the software is controlled by a human operator to simulate the what the response would be[31, p. 395]. By making use of the programs that followed with the setup of the robot, we were able to control the robot remotely. This will help us get an opinion about the robot's action and not the person controlling it. While simultaneously provide a safe environment for experimenting.

4.3.2 Preliminary procedures

Before running tests with participants, I and two postdocs researchers at MECS, did some testing in what was then an unoccupied room. There the Turtlebot3 did a slam mapping of the room. This data is to be used for prototyping the robot, and to get a better understanding of what type of rooms we can expect the robot to operate in.

Data of the room were gathered in two scenarios. In the first scenario, the robot mapped the whole apartment, with no obstacles. In the second scenario, the robot mapped the living room of the apartment with 3 chairs as obstacles. The chairs legs were rather thin and proved to be a good challenge for the robot to detect. It was not until the robot was very close to the chairs that it was able to perceive its legs. Rather than having the robot confused about if it's seeing legs or not, it might be better to exclude such areas in the future.

4.4 scenarios

4.4.1 scenario: Morning coffee

The robot approaches the person in question with a cup of hot coffee. The person in this scenario is sitting down at a chair behind a table, while the robot is approaching the person from the most accessible point with a cup of coffee. The questions

4.4.2 scenario: Following

The robot is following the person in question. The person walks in front of the robot, and the job of the robot is to follow the person with a set distance between the person and the robot.

4.4.3 scenario: Avoidance

The person is walking towards the robot, and the robot avoids a collision. The person is walking across the room on a collision course to the robot. The robot responds by moving directly out of the way of the path that the person is on.

4.4.4 scenario: Standing still

The person is standing still, the robot is moving around.

4.5 Qualitative questions

Scenario one
What is your reaction to the action that the robot just performed?
How do you expect the robot to give you a cup of coffee?
(follow up question) Do you want the robot to be close enough that you can reach out and get the cup of coffee?
Do you want the robot to stick around at the current position or should the

robot move away?
(end question) How do you want the robot to approach you when giving you a cup of coffee?

scenario two
How would you describe the experience of the robot following you?
Was the robot to close for comfort or too far away?
Was the position of the robot a problem?
(end question) How would you have the robot to follow you?

Scenario three
What did you expect the robot to do?
How would you describe the reaction time of the robot?
How would you want to behave in this type of scenario?

Scenario four
If you and the robot was in the same room and the robot started to move around what would you do.?
How do you expect the robot to behave if it approaches you in this scenario?
What will make it easier for you to trust the robot when doing this action?

4.5.1 Godspeed questionnaire

To get a deeper insight into what participants thought of the robot, the Godspeed questionnaire[2] was used. This is a frequently used questionnaire series in the Human-Robot Interaction (HRI) community. As of this writing, it currently has 400 citations total and 40 citations in the International Journal of Social Robotics. This is according to the statistics page on Springer. Why this questionnaire was used, was because it lined up with what we wanted out of the experiment. It also had a good reputation among those who used it.

The GQS aims to give a tool to robotic developers, to help monitor their progress. To measure the attitudes a person has towards a robot it uses five-point semantic differential scales. For the evaluation of social human robotics five consistent questionnaires are used. These are Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety. One downfall of the GQS is the similarity between some of the questions[43].

4.6 Recruitment of participants

The people in charge tried to recruit some participants for us however since they only interact with the people who are sick, they found no willing person. To get people interested in the project and to participate we were advised to make a presentation about how this project can benefit them.

4.7 Pilot Experiment

Before having the full-on experiment at the retirement complex a pilot was conducted. The pilot was done by the writer, the supervisor and another master student. In this pilot, every scenario was replicated and done, multiple times if there was a reason to believe the outcome would be different. Before and after each scenario was conducted, the approach of the robot and the position of the individual was discussed. By doing this we found out what we expect from the individual participating and potentially what a participant expect from us. For instance in the scenario where the person is sitting down, it was not clear when the participant should interact with the robot. To remedy this we will clearly state that the robot can be interacted with when it is at a standstill. When the robot was following a person, it would stop too close two where the participant stopped. This would inconvenience the participant when they were moving too the next goal. As such when the experiment was done a second time the robot was stopped at a further distance. There were four different avoidance methods conducted. When the participant and robot approach each other, the robot will avoid to the left in the first experiment. In the second the robot will avoid to the right, this is to take into account any person from abroad. For the third experiment, the robot will stop dead in its approach at approximately a meter in front of the participant. This is to take into account unforeseen behavior by the robot. In the final experiment, the robot will stop and then move backward while the person is still walking. As an easy collision avoidance mechanism, if the robot came from a place it is able to move back to that place. It was decided to use the first experiment, as it was the most natural.

Until now it had not been finalized how we wanted the participant to act in some of the scenarios. Here we found how we want the person to act. When the robot is following the person, the person should first approach a table and stand there for a second. The person should then approach another area, at which point the scenario will end.

Some of the other potential problems that were addressed where which robot to use in which scenario. Even though this thesis is written with the basis of the turtlebot3, we do have at our disposable turtlebot2. It was decided that this would be used in the first scenario because of its size. Since the turtlebot3 is quite small the participant would have to reach down to grab the item. In the case of Turtlebot2's, it would be easier to grab the item because it's much taller. After all the pilot was done, all participants would answer the questioner. This was done in their own time, where eventual edits would be passed forward and sorted out by the supervisor. We also decided not to use coffee as it could be hazardous.

There were a few changes made to the task we wanted to do with the senior citizens. The standing still experiment was cut out because it had little to do with what we are trying to find out. There is no reason why the robot would behave as such, other than to look for the user. However, in this scenario the user is standing in a space visible for the robot. For the following scenario, we wanted a two-stage approach. This would give the participant more of a feel for how the robot behaved. The reason behind this is that having the robot just follow behind one can feel like little more than a tech demo. If the participant is placed in a situation where the robot is following in a more natural setting, the opinion might be better expressed. When on-site conducting the experiment, it was decided that this took to long. In the end, we reverted back to the robot following the person in a straight line.

Other than the changes to the scenarios, there were also changes to the qualitative questions. Since they were too general in their approach, they were reworked to be a bit more on point and a scale was introduced. This scale is meant to be interpreted in a general sense, as there will be no quantifiable measurements with it. Finally, we decided to not include the first questionnaire in GQS as it had nothing to do with our robot. The intention of the robot was not to see if human behavior can be attributed to the robot. But rather if the robot can be accepted in the living space. This is why the rest of the questionnaires are much more important for what we want to find out.

4.8 Conducting the experiment

Figure 4.4: Conducting the experiment

The ones doing conducting the experiments were the writer of this thesis, a fellow master student and the supervisors. A postdoc. was also present for a short time. But because his equipment did not work in the crowd, he decided to leave after a short while.

Our stay at Kampen Omsorg+ was from 9:40 to 15:00. From the participants view our stay coincided with the beginning of lunch to the middle of the dinner period. This was the time slot where most people would be available for us to ask if they would participate. One shortcoming that was made apparent at the start, was that people had no idea what our robot looked like. This was because the management had not been able to copy the image of the turtlebot3 to the poster. As such a picture of the anki cozmo robot[2] was used. So the robot that they expected, which were along the lines of a bigger robot with arms, did not line up with the reality that

[2]https://www.anki.com/en-us/cozmo

was the Turtlebot. However, a positive outlook of this was that they did not feel threatened by the small robot. Much of this information was gathered from bystanders that did not want to participate in the experiment.

We were seated close by the dining area/meeting area. Right by a corner between the dining area/meeting area and a general seating area, which were close to the entrance of the building. Our seating position was in an open arrangement. It was made to make us appear open to any an all, and encourage participants to approach us.

In that time we managed to interview 5 people. Out of those 5 interviews 4 were done by me and the last one were done by my supervisor. This was because the person did not speak any language that I knew. The lack of participants was unfortunate but there are some reasons that one can point to. For the first, the people being interviewed were much more willing to be a part of the project, when there where somebody that they knew close by. When one of the managers were there to help, people seemed much more willing.

While doing the experiment, we were in an unprepared state. The factor that contributed most to the unpreparedness was the rush we were in before doing the experiment. One of the things that contributed to this was the questionnaire whose final draft was not done until a few days before. At this time everything was prepared in English and it was up to me to translate everything to Norwegian. This took time because of the technical termination used in the language. Due to various reasons outside of my power no official recording device, of the voice or video kind, was secured. There are two ways that one can view this. Since we had to go, as this was the only day that we could do this, a phone acted as the recording device. This made our setup look less professional, but on the other hand, people might have been less intimidated. When looked at the other way, we ended up not having any voice recordings or good video setup that recorded their official opinion and reaction. That makes most of the opinions stated here those that I wrote down and remember, and not something I can point to.

There were four types of people that were met when asking around for participants. The people who were not interested in the project at all. The easily irritated person who was not asked, this was to avoid any complications. The busy person who did not have time. This includes persons that we asked and people who were having visitors whom we did not ask for due to respect. The interested person, but who did not want to participate. This was the majority of the people. And lastly, the people who did want to participate. In the time we spent there, many of the people who we first meet who had left would later come when it was dinner. It was by that time we had more or less spoken to all the people who were genuinely interested.

An earlier session by a different group[32], had a presentation but they found out they could not keep the attention of the participants. For our session, we wanted to retain the participant's attention. To do this we told whom we were and what we did in a one on one conversation. In light of the fact that the place we were stationed was quite active, with people coming and going all the time, this was a way to cater to most people.

However other than the small robots and the posters, it may not have been clear for people what we were doing there. A poster off some kind or maybe a video feed about the project might have helped with this. As such being three to four people sitting there while a person asked people to participate might have been to our detriment. Most people aren't really interested in taking part in, something that they don't have too when they are approached in this way. That was the kind of implicit feedback we did get from doing this.

Most of the people who were explained the idea of the robot did not see what it was good for. As most people did not have a camera in their apartment, by rather relying on a panic button, they did not see the potential of the robot. For them, the panic button was something more connected to its task than the robot. Another aspect of this, it was rather difficult to explain the function of the robot, because they could not understand how it worked. In this case, I would consider it my fault for not giving them a better explanation that was more down to earth. The comparison people understood the best was between the Turtlebot and a robotic vacuum cleaner. From this, they understood why we wanted the robot to be small, but it did still take time to explain the concept of the robot.

Each interview was a one on one session with me and the participant. To save time the form each participant had to sign was explained to them. The parts that were of great importance, like the video capturing, what happened to the data, and that they had the right to refuse at any one time, was heavily emphasized. The form was also given to them if they wanted to read it themselves before signing. Before the practical test where done, the participant was asked 3 questions from the Godspeed questionnaire about perceived Safety around the robot. Afterward, the participant was then asked to sit in a chair and the robot would move to them and then stop. The participant could then take a piece of banana that was on the robot. After this, the robot would then move back to its original location. Since there were people with different mobility issues a standard length between the robot when it stopped and the participant was not defined. What is meant by mobility issues is that some of the participants were bound to wheelchairs and some could walk, but with difficulty. Besides if we took into consideration the definite length between the robot for each case, they might more easily assume that we were the ones controlling the robot.

When the first practical test was done the participants were asked the questions related to the test. In the next practical test, the participant was asked to stand up, the small Turtlebot were then placed slightly behind them. To accommodate the different participants the setup of where they would walk was done on the fly. When we said go, the participant would start moving and the robot would follow. After having moved for a bit the participant was told to stop. In every case in which this test was don, except one, the participant always outpaced the robot. The one case where this did not happen where when the participant had a hard time walking. Another problem was that each participant did not see where the robot where in relation to them. This was not made better by our test being done in a busy environment.

In the last practical test, the participant was asked to stand up while the robot (Turtlebot3) was placed a bit away. The participant was then asked to go towards the robot with no further explanation of what would happen. This was done to get their genuine feeling about this situation. Some of the problems encountered during this experiment were that the robot could not avoid the participant in time. This is in relation too the robot being a bit to slow. Another related issue to this is that manually turning the robot with the controls that we used, that being Turtlebot3 teleop, took time.

After having answered the questions related to this experiment the participant was asked to answer the general questions and the GQS. This was what quite a lot of time went too. Another issue in relation to the GQS was that it may not have come across which robot we were talking about. This made the questionnaire more general towards the kind of robots the two Turtlebots are.

In general due too the on the fly method to accommodate for the different people the experiments were not done in the exact same way. For some of the cases, the Turtlebot was in a place hard to see for the driver, too make enough room for the experiment. Due to the speed of the robot, the last experiment was a particular issue. The driver had to account for the speed of the participant and the robot. Since there also was no markings on the ground, when the robot turned, and how far the distance was to the robot when it turned, wildly varied.

During the experiments, the participants were never explained that we were the ones controlling the robot. Since some of the participants did view the experiments before they themselves participated some did figure it out. Whenever somebody asked if it was controlled it was explained that the one controlling the robot was in charge of the dead man switch. It was never explicitly or implicitly said that we did control the robot, the subject was just diverted.

4.8.1 Ethics

When working with people of advanced age, there are ethical issues too take into consideration. Age-related vulnerabilities such as cognitive decline[26]. And special to our case the interaction between seniors and robots[30]. The management where there at the start to help us get people that get people who could understand and be able to sign the contract. However in the time that they were not present it was up too us. For all the participants the main parts of the contract were verbally explained. At the time of explanations, it was also checked if they understood what it meant, by having them respond. The consent form can be viewed in Appendix B.

The data gathered, whether it be the filled out questionnaire or pictures taken during our time there, were given to the supervisor at the end. Appendix A is the questionnaire used in the experiment. Pictures used for this thesis were formatted in a way that any personal information cannot be extrapolated.

In regards to physical hazards during the experiment. It was judged to be a low possibility of a participant having an accident with the robot. We were in an open location with ease of access to help. There was always one person of, our group, close by the robot to prevent any accidents.

Chapter 5

Implementation

This chapter outlines the work done to develop the detector, that includes the segmentation and classification, the tracker, and the monitor. The goal of this design was to make it modular, where each module is self-contained. There will then be a quick overview of the costmap used. Finally, there will be a description of the data gathering for the classifier.

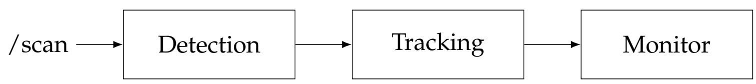


Figure 5.1: Overview of implemented nodes

5.1 Detection

This part is based upon the paper by Arras et al. [1]. The implementation is also influenced by the solution of the SPENCER project[23]. The detection module consists of two parts, segmentation, and classification. Here the segmentation is responsible for preparing the data for the classifier. The classifier will then evaluate the prepared data. After the data has been evaluated it is then filtered for anything that did not evaluate as a leg. The filtered data is then posted as a message in ROS for other nodes to see

5.1.1 Segmentation

Before anything can be done, the data need to be converted from distance coordinates to polar coordinates. This is done through the cartesian coordinates to polar coordinates formula. For k from 0 to 359.

$$x_k = r_k * cos(\theta_k)$$
$$y_k = r_k * sin(\theta_k)$$

After the data has been converted, it is then partitioned based on a few criteria. Each partition must have more than 3 polar coordinates of data. The distance of a point must be within the operating distance of the LIDAR. Meaning that any point closer than 120 mm or further away than 3,500 mm is excluded from a partition. Lastly, the distance between the first and the last point must be larger than 5 cm. This setup is based on the original paper but is specific for this LIDAR type.

The data is partitioned is based on the euclidean distance between point p_k and p_{k+1}. If the absolute distance between the points is more than 20 cm. This is to take into account legs that stand close together. For each partition, the mean and median are computed and stored with them. Furthermore, for each partition the data point just before and after the partition is stored. Together with the number of points per partition, these are stored for later use in the features

$$mean(x, y) = \frac{1}{n}\sum_{i=0}^{n} x_i,\ \frac{1}{n}\sum_{i=0}^{n} y_i$$

After having segmented the data, the partitions are filtered by length. Every partition that has a length higher than 5 cm and lower than 60 are stored. It is these remaining segments that represent possible candidates.

One part of the segmentation that is not taken into account is the possible partition directly in front of the robot. This is a potential blind spot, where what should have been one partition is two. Or what should have been a partition is not, because the two partitions are not large enough.

5.1.2 Features extraction

For each possible candidate, 13 features are calculated to be used in the classification. These features are based on the implementation by Arras et al[1]. The implementation used in this project is from the SPENCER project[23]. There will now be a quick overview of each feature.

1. Number of points per segment, n of S_i.

2. The standard deviation.

$$\sigma = \sqrt{\frac{1}{1-n}\sum_j \|\boldsymbol{x}_j - \boldsymbol{\mu}\|^2}$$

   Here the μ is the mean value of the segment.

3. The mean average deviation from the median. This is designed to measure compactness more robustly than the standard deviation for each segment. The median $\tilde{x}$ is defined as such:

$$\tilde{x} = \begin{cases} x_{(K+1)/2} & \text{if K is odd} \\ \frac{1}{2}\left(x_{K/2} + x_{K/2+1}\right) & \text{if K is even} \end{cases}$$

   Here K is an ordered set of scalar random samples x_i. Where the median is then used to calculate the average deviation:

$$\varsigma = \frac{1}{n}\sum_j \|\boldsymbol{x}_j - \tilde{\boldsymbol{x}}\|$$

   Here $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y})$.

4. TheJump distance from preceding partition. Here the Euclidean distance is used:

$$d = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2}$$

   Where q is the first point of S_i and p is the last point of S_{i-1}.

5. The Jump distance from succeeding partition. Just like the last feature, the Euclidean distance is used. But here the distance between the last point of S_i and first point of S_{i+1} is used.

6. The width. This is the euclidean distance between the first and last point of the segment.

7. The linearity. This measures how well a line is fitted to a segment S_i using the residual sum of squares method. The line is fitted in the least squares sense on a segment. Where the residual sum of squares is calculated as such:

$$s_l = \sum_j \left(x_j \cos(\alpha) + y_j \sin(\alpha) - r\right)^2$$

Here $x_j = p_j \cos(\theta)$ and $y_j = p_j \sin(\theta)$. The points are given in polar coordinates and the line is fitted in the Hessian (α, r) representation.

8. The circularity. In this implementation, the residual sum of squares is used. Here the best circle in the least squares sense is found by parameterizing the problem from a vector of unknowns. Where $x = (x_c,\ y_c,\ x_c^2 + y_c^2 - r_c)^T$ and x_c, y_c, r_c denote the circle center and radius. From this the overdetermined equation system $A \cdot x = b$ is established.

$$A = \begin{bmatrix} -2x_1 & -2y_1 & 1 \\ -2x_2 & -2y_2 & 1 \\ \vdots & \vdots & \vdots \\ -2x_n & -2y_n & 1 \end{bmatrix} \quad b = \begin{bmatrix} -x_1^2 - y_1^2 \\ -x_2^2 - y_2^2 \\ \vdots \\ -x_n^2 - y_n^2 \end{bmatrix}$$

This is solved by using the pseudo-inverse.

$$x = (A^T A)^{-1} A^T \cdot b$$

The residual sum is then used to find the circularity.

$$s_c = \sum_{j=1}^{n} \left(r_c - \sqrt{(x_c - x_i)^2 + (y_c - y_i)^2} \right)^2$$

9. The radius. Here the value of r_c from the last feature is used. This feature was proposed to serve as an alternative measurement for the size of a segment.

10. The boundary length. This measures the total distance of the poly-line made by the segment.

$$l = \sum_j \| \boldsymbol{x}_j - \boldsymbol{x}_{j-1} \|$$

11. The boundary regularity. Here the deviation of the distances between the points in the poly-line, from the previous feature, is calculated.

12. The mean curvature. To calculate the average curvature over a segment, an approximation over a discrete curvature is used. Given three points in succession $\boldsymbol{x_A}, \boldsymbol{x_B}, \boldsymbol{x_C}$, the three distances between the points d_A, d_B, d_C are used. From this the approximation is calculated:

$$\hat{k} = \frac{4A}{d_A d_B d_C}$$

The A denotes the area of the triangle made by the points. Then the average curvature is calculated:

$$\bar{k} = \sum_j \hat{k}_j$$

13. The mean angular difference. For this feature, the intention is to measure the convexity/concavity of a segment. This is done by calculating the average of angle β_j between two vectors, as shown below:

$$\beta_j = \angle\left(\overline{x_{j-1}x_j}, \overline{x_j x_{j+1}}\right)$$

The original paper lists 14th features, but we have opted not to use the last one in this implementation. This is because Arras et al. described it as a marginal improvement. The most important features in the paper where features 9, 4, 5, 2 and 3 in this order.

5.1.3 Classification

The AdaBoost classifier was implemented as its own small library with four main functions. These four functions are the training function, that takes a dataset of features and a label set, to then make a model for the AdaBoost classifier. Then there are a save and load function to respectively save and load a model. Finally, there is the evaluation function, that takes a matrix of features and gives out a predicted label for each feature set, given a model. Each function makes extensive use of the OpenCV library[6] on boosting. The code functions as a shell to get it to work with this specific implementation. To train the classifier another program was made with the necessary functions. This includes reading and writing raw data, splitting a data set and randomly shuffling the data.

5.2 Tracking

The second node has two main responsibilities. It responsible for filtering the result of the classifier, as the result of the classifier is not perfect. And it is responsible for keeping track of a person for a while, even if the person is obfuscated. One final thing that the tracker node does is to give a predicted future path for the tracked person.

5.2.1 Filtration

To filter out possible noise in the environment the Nearest neighbor classifier was used. The purpose of the filter is to take care of all the false positives. For this to work, we need to know what can be a person and what probably is not a person. A way to do this is to say that whatever moves is a person. This is the way it is implemented here, if a detected point is moved from its original location it is seen as something to be tracked. To start out, each point that is classified as a person from the classifier will get a change in frame of reference. The points frame of reference goes from the robot's point of view to a global point of view. This is done to have control over the location of the points. As the robot is a moving object, if the points are in its frame of reference, they would also move. These global points are what the nearest neighbor classifier will be working on. When the classifier first starts running it stores all the global points. In the next instance, when a new set of global points are given, the new points are compared against the old. Any points that are not in their old location, but have moved within a certain threshold are classified as a person. Since a person may stand still at any one time, the algorithm has to take this into account. To do this a previous point that has been classified as a person, where the point is now standing still, will still be classified as a person. When calculating the distance between a previous and a current point the distance metric used is the L2 norm.

$$\|x\| = \sqrt{x_1^2 + ... + x_n^2}$$

The reason this distance metric is used is that it is here assumed that the walking pattern of a human is fairly linear across a short timespan.

There are some drawbacks of implementing it this way. For one, this algorithm has no knowledge of the environment that it is put in. As such it can only draw its conclusion from how it interprets the movement of the points. Whether it is the same point compared to its old position or an entirely new point, it does not know. Another drawback is that it will not identify anything that does not move like a person. Say for instance the robot is roaming around and looking for a person, but the person is standing still. This algorithm will not see that person. One final issue that this algorithm can not deal with is that of sporadic points. These may inadvertently make the classifier misclassify an object of being a person. This often happens with points that randomly appear and disappear, besides a static object.

5.2.2 Kalman Filter

The Kalman filter is going to be implemented with the use of OpenCV. Here the calculation of the filter is automated and the operation only relies on the model of the system. The model for the filter chosen here is the equations of motion for 2 axes:

$$
\begin{aligned}
x_{k+1} &= x_k + v_{x_k} t + 1/2 * at^2 \\
y_{k+1} &= y_k + v_{y_k} t + 1/2 * at^2 \\
v_{x_{k+1}} &= v_{x_k} + at \\
v_{y_{k+1}} &= v_{y_k} + at
\end{aligned}
$$

The factor to why this was chosen is speed. It ends up being a filter that converges fairly quickly. This is quite important as the speed of a person walking is fairly quick. Which relates to the operational detection range of the LIDAR which is fairly small6.10. This also relates to the speed of the LIDAR which is quite slow for this application, only 5 revolutions per second.

The aim here is to fit the above formulas to the linear stochastic difference equation x_k and the measurement z_k.

$$
\begin{aligned}
x_k &= Ax_{k-1} + Bu + w_{k-1} \\
z_k &= Hx_k + v_k
\end{aligned}
$$

Here A represents the translation matrix. B represents the operational control input. And H represents the measurement matrix. To converting the formulas to matrix notation, based on the formulas above. Not including the error terms as those will be dealt with later

$$\begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T^2/2 \\ T \\ T \end{bmatrix} a$$

For the H matrix, we are only recording the position, as such:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

For the measurement error matrix (R) we have that x and y are independent. This means that the variance will only be along the diagonal of the matrix. However, as we don't know the error it will be found out by trial and error. For the process noise matrix (Q) we again go by the fact x and y are independent, but x and v_x are dependent. Since this models the error to the input of the system, we are going to use the acceleration term. There is however a problem, the model is dependent on the variance of the system, but we don't have the mean (μ). As such we are going to model the system first without the mean, but put the mean on later as a scalar to the matrix. From this we get:

$$Q = \begin{bmatrix} T^4/4 & 0 & T^3/2 & 0 \\ 0 & T^4/4 & 0 & T^3/2 \\ T^3 & 0 & T^2 & 0 \\ 0 & T^3/2 & 0 & T^2 \end{bmatrix}$$

In the end, we are left with 5 matrices for our model.

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad R = \begin{bmatrix} \sigma_x & 0 & 0 & 0 \\ 0 & \sigma_y & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0. \end{bmatrix}$$

$$B = \begin{bmatrix} T^2/2 \\ T^2/2 \\ T \\ T \end{bmatrix} \qquad Q = \begin{bmatrix} T^4/4 & 0 & T^3/2 & 0 \\ 0 & T^4/4 & 0 & T^3/2 \\ T^3 & 0 & T^2 & 0 \\ 0 & T^3/2 & 0 & T^2 \end{bmatrix} * c$$

The c term represents a scalar for how much variability we expect for our model. In the R matrix, the sigma terms represent the measurement noise in the x and y-direction respectively. Since we are using a LIDAR, if we can find the person, the error is going to be very low. This is because it is not much that will interfere with a LIDAR in this setup.

5.3 Monitor

The monitor is the node that decides what the robot is going to do, given the position of a person. Given a predefined action, the monitor will make use of move_base to move as desired. The purpose here is to make a centralized node regarding the actions of the robot.

5.3.1 Area detection

The robot has to respond to actions that the user takes, but can only see the position of the user. To make use of this, user-specified areas that are on the global map will convey the action of the user. How this works, is that if a user enters one of these areas and the robot detects this, the robot will respond with a movement action. This is where the winding number algorithm is used. It is implemented in the monitor function where a generalized implementation[35] is converted for this application.

5.4 Costmap

The costmap specifies an area where the robot should not tread. This is a package[1] that follows with the software of the Turtlebot. Its function is being a program that takes in plugins related to the issue of costmaps. However, it only specifies costmaps around static structures, like a wall. To have a costmap for a person another package was needed. In this case, a plugin for a social navigation layer[2] was used. A social force model was, in this case, decided against because of the complexity and lack of necessity.

5.4.1 Area avoidance

To make the robot avoid user specific areas, the ROS package costmap_prohibition_layer[21] was used. As this package is only responsible for drawing the areas on the global costmap, an interface had to be developed to point these areas out. The area_gui program is responsible for this. It lets you specify an area on the global map as a dot, a line, or an N-dimensional polygon. This area will then be saved to an external file, such that costmap_prohibition_layer can make use of it.

5.5 Data gathering

To gather data we first made use of the interactive_data_gathering tool. This happened in the robotics master room where we had 5 people a fellow master student and myself included stand in front of the robot doing predetermined actions. These actions were, to stand in front of the robot with a distance of approximately 1 meter. While in this pose you should move 45 degrees to the left on the spot 8 times. This ensured that we had a few standard poses that were the same for all participating individuals. The next action was to walk directly towards the robot in a straight line. And then walk directly from the robot in a straight line. The last action that we asked the individual to do was to walk across the line of sight of the robot in two distances. One distance of a meter in front of the robot, and

[1]http://wiki.ros.org/costmap_2d
[2]http://wiki.ros.org/social_navigation_layers

one distance of half a meter in front of the robot. 88 original samples taken by hand.

These poses were initially chosen to gate a baseline of poses that the classifier could familiarize with. The first 8 poses were to get a picture of a causal pose from many angles. By having the person walk directly towards the robot we had the scenario where the person came to pick the robot up or otherwise have an interaction with the robot. When the person was walking directly away, we had the scenario where the robot followed the person. The last two actions are meant as a common interaction, where the robot is present, but not part of any objective of the user.

An aspect of the data gathering was diversity. In the sense that exact measurements of how far the person was from the robot were not enforced. The reason for this is that practically no two people will have a definite distance that they stand in front of the robot with. The data's one and only purpose is to be of use to the classifier. Other aspects that were not indicated to the participants where how their legs should be positioned. They were told to stand in a casual relaxed pose, but no indication was made to what that might be. People have different natural standing poses, and we wanted to highlight that. This procedure was revised the next time we gathered data for a few reasons. First, it was inefficient to gather the data by hand even though we have a tool developed for this. Second, we only gathered 1 point of data for each action the user took. third, the action the user took was not necessarily natural. fourth, the action that the user took was not enough to make a good base set of poses. This comes down to the fact that the user might not be as close to the robot as we had out participants be. Given that we hope the robot will be a natural part of the house a person would move more casually around the robot.

To handle this we introduced a few new actions for the participants to do. First of all, after having done the 8 poses in front of the robot, the participant should take two steps backward and stand causally in. Then he/she should turn 90 degrees to the left. This increased our baseline for poses and took into account that data gathered through a LIDAR looks different based on the distance of the object. The next actions were as previously described, but we introduced a new last action. Here the participant should move freely around the robot and behave in a natural manner. To handle the case of data gathering, we opted for the use of rosbag[3]. Rosbag is a tool meant for recording and playing back ROS topics. This way we could gather all the distinct poses that the participants took. To later play back the recording and gather the data.

To gather data we set the robot up in a room with good space for the robot and few disturbances that could hamper the data. After having everything set up we went out together to ask for participants. At first, we asked single individuals, but we quickly found it easier to ask groups of people. We found that when some of the group members were interested, the rest of the group who may not have been as interested, naturally followed. After having asked all nearby groups of individuals, we moved location to the bachelor room for robotics. The robot was set up in the same manner and all the people present were asked. This room was chosen because it is the room for people who take the robotics line. And the belief that the people present may be more inclined to take interest in the robot and the project, than the general person. As all but one person present in the room participated, this belief could be said to hold true. Finally done in the robotics room we headed back to the masters quarters. Here we set up the robot one final time and got participants from the 4 employees at Ifi. The second dataset that was gathered was of non-label data. Everything that has nothing to do with legs but things that the Turtlebot might come into contact with. The data was in its entirety collected at Ifi in various hallways, meeting rooms and computer labs.

As stated, most of the participants were students from Ifi. We gathered data from students, primarily because of availability. It was easy for us to come into contact with them and most students could afford to spend 2 minutes of their time with us. Most students that we came into contact with, were generally willing to participate. They also showed an interest in the project. This might be because most students were bachelors and showed an interest in what they might be doing when they became masters.

There are some biases in the data that is collected. First of all, the labeled data represents the poses and walking movement of students for the most part. This should not have an impact as the identification algorithm works on segments of clustered data and not walking motion. As long as the leg features of students and elderly are not extensively different it should not be a problem. People were not explicitly told how to stand and this resulted

[3] http://wiki.ros.org/rosbag

in some people standing with their legs packed closely together. This is not what one would consider a natural relaxed pose as the footing and balance is not as firm as if you had your feet apart. Another point is that most the students had shoes on. Some of the shoes did stretch long enough up on the legs to impact the LIDAR. The non-label data is heavily biased towards an office space. This will most likely have an impact when sorting feet from the environment in a home space.

In the first data gathering session, there were 5 participants. In the next data gathering session where rosbag were used, there were 32 participants. Of these 32 participants 10 girls and 22 guys. In the later labeling part, we found that some of the rosbag files were not recorded properly. This was not checked under the data gathering sessions as it was assumed that the system would work. As we did not record any name or gender to any of the files, it is not known how many of each group are represented in the labeled data. An assumption can be made that roughly 30% of the labeled data does represent females.

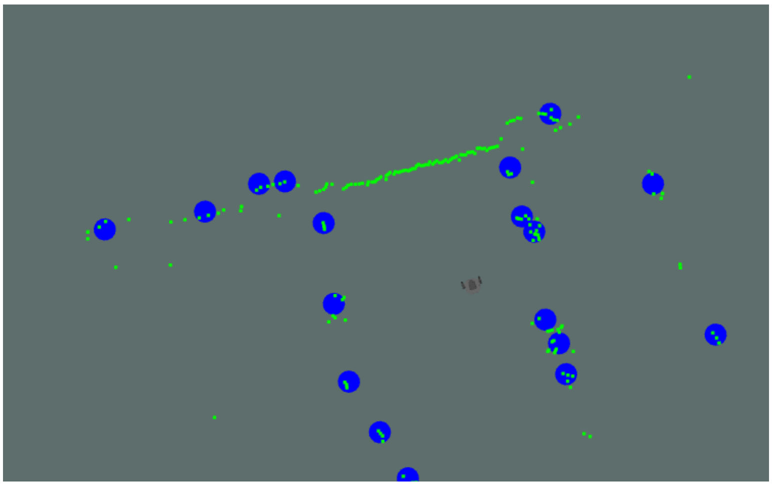


Figure 5.2: The interactive_data_gatherer

In figure 5.2 you can see the data gathering of non-label data, through the use of rviz. The blue dots represent what the system views as data clusters of LIDAR point. In this session of data gathering, a minimum of 3 points per data cluster was used. However, this number of points per cluster turned out to be detrimental to classification accuracy. This is because, a cluster of three points of non-label data can easily be confused with a leg, by the classifier. On the left-hand side of the figure one can see two blue dots with three green points in the middle. This is also how a leg would be viewed when at a distance from the robot. To get around this, the minimum number of points per cluster was increased to 4. This made the classifier more robust, while not having too large of a negative side effect. The negative side effects here being a decreased distance of classification area around the robot. Had the minimum number of points been set to 5, the classification distance would have been too small to use. This can be viewed in figure 6.10.

Chapter 6

Results and Experiments

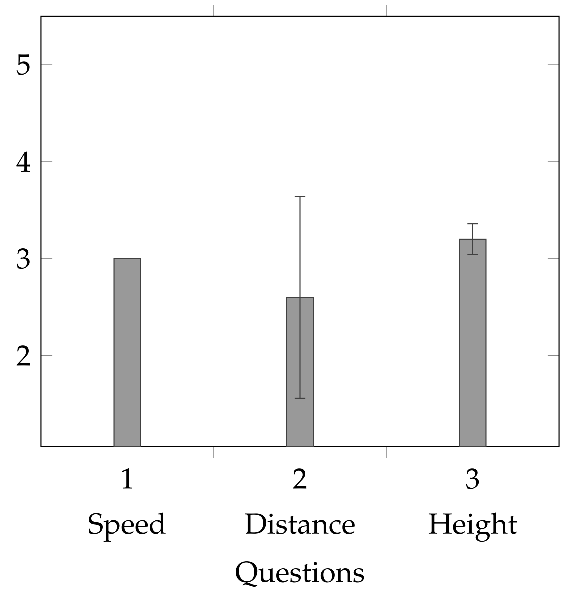
This chapter will go the various experiments and results that were conducted. There will first be a section going through the questionnaire in regards to the experience at Kampen Omsorg+. Then there will be a section on the result of training the classifier and how it compares the results of others. Finally, there will be three sections that test the capabilities of the robot. The first section tests the detection range for a person. After that, there will be a section testing how the navigation system handles noninterference areas. Then lastly there will be a section on experiments with the action areas.

6.1 Questionnaire results

This section presents the results from the questionnaire used at Kampen Omsorg+. It is split up in two sections, the first part gives the results for the general questions. The next sections give the result for the Godspeed Questionnaire.

6.1.1 Results of the general questions

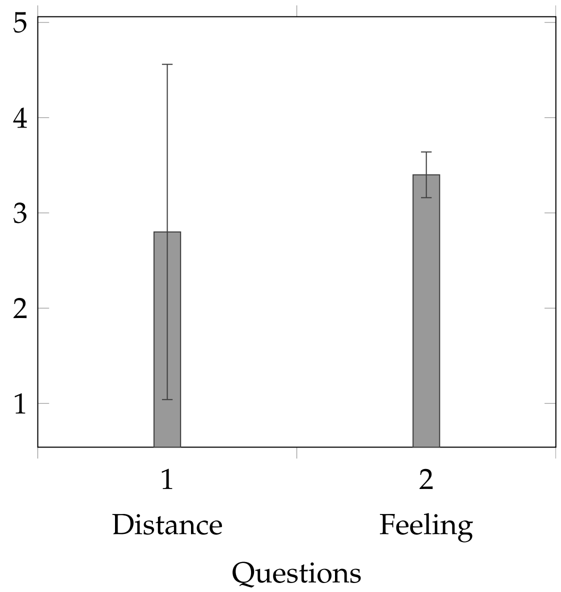
The results for the general parts of the questionnaire should be taken with a grain of salt. By reason of the sample size and quite possibly of the type of people who participated. The type of people who participated may have been seen from the onlookers' point of view as their representatives. In this way, other opinions may have been suppressed. As such it should be viewed as the thoughts of a small group of people. The discussion of these graphs will also cover what can be done better next time, based upon the answers and observation.



	Participants				
Q	1	2	3	4	5
1	3	3	3	3	3
2	1	3	3	2	4
3	3	4	3	3	3
4	1	2	3	1	1

Figure 6.1: Candy results

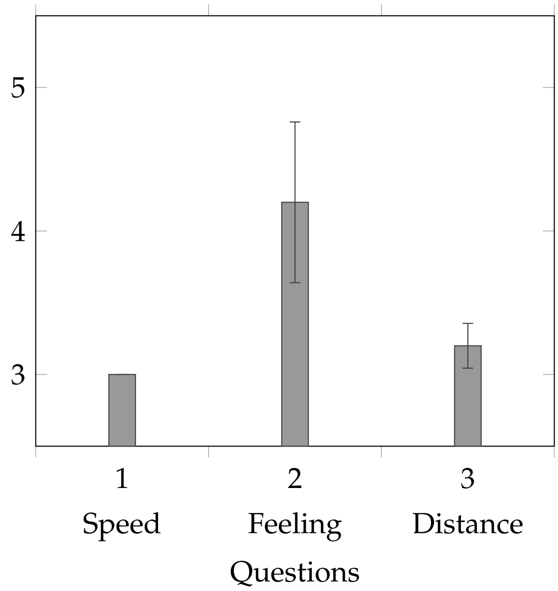
In the figure above, the bar reflects the mean value, while the error bar indicates the variance. To the right of the bar graph is a table of the answer for all participants. The experiment was conducted with the Turtlebot 2, which is much larger than its successor. Because of this, the result will be interpreted as independent of the other results. The first graph asks about how the participant interpreted the speed of the robot to be. As can be seen, by the bar graph and table, the speed was determined to be ok. The uniformity of the answers does not give an indication if it would be fine with other velocities. If this experiment should be performed again, a higher velocity could be used to test the comfort zone. For the next experiment, there are different opinions of what the distance from the robot to the human should be. In the experiment, we used a distance of approximately 10 to 15 cm from the front leg of the chair the person was sitting on. There are several factors that can affect the outcome of this. Among others are the person's height and arm reach. This points out, that for this group of people, that distance is a personal matter. At least in terms of reach, it does not say anything about how comfortable people were with the robot at that distance. In regard to the hight of the robot, no participant had any major problem with it. However, if the robot is going to serve as a service robot a variable high might be beneficial. Even though it may not make a difference with chairs, if the person was standing it might. For the final question, we asked people what they wanted the robot to do after it was done serving the snack. From the table, we can see that most people requested that it stood still. However individual opinion does seem to affect the answer of this group.



	Participants				
Q	1	2	3	4	5
1	5	2	3	1	3
2	4	4	3	3	3
3	1	2	2	2	1

Figure 6.2: Following result

For the rest of the experiment, the Turtlebot3 was the one in use. In this experiment, the robot was following the person for a short distance. The robot was in most of the instances too slow to keep up with the person in question. In the instances that the robot was able to keep up, the participant had mobility issues. A problem observer during this experiment was that most participants did not know where the robot was in relation to them. This makes it impossible to judge this result. All in all a higher velocity should be used if this was to be tested again. The speed of the robot could be reduced if it posed a problem. In the question of how people felt, by having the robot follow them, the answers were along the same line. However, this experiment was conducted in an open space with people around. To get a better answer for such a question is should be done with fewer people. Preferably at an apartment, where the question would reflect the situation better. For the last question regarding the position of the robot relative to the person. A separate experiment where this is the case might be necessary to get a better opinion.



	Participants				
Q	1	2	3	4	5
1	3	3	3	3	3
2	4	5	3	5	4
3	3	4	3	3	3
4	2	1	2	2	1

Figure 6.3: Avoiding result

In this experiment the robot was in full view, will it and the participant walked towards each other. The robot would try to avoid the participant before they meet each other. For the first experiment, it could be beneficial to have to robot be a bit faster. By the same reason, as in the result for candy. For the next question, the feeling of the situation was evaluated. Here the person in question had the robot in view at all times. However, it was also in a place with a lot of people around. The reflection from the last graph on the same subject is the same here. In the next question, the general distance of the robot from a standing person was covered. The operator of the robot tried to turn the robot when it was a little over a meter away from the participant. Due to the slow velocity of the robot, this was not always achieved. For this group of people, it appears to be an appropriate distance for this robot. In the final question for this section, we wanted to know what the participant wanted the robot to do in this situation. Her the answers where ether to have to robot do what it already was doing or to stop completely. Nobody wanted the robot to stop and back out.

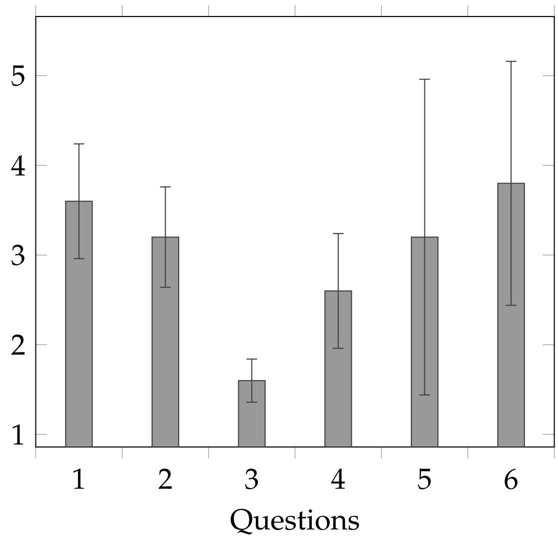
	Participants				
Q	1	2	3	4	5
Parking	3	1	1	3	1
Kitchen	1	1	1	3	1
Living room	3	1	1	3	1
Bed room	3	1	1	3	1
Bathroom	2	2	1	3	3

Figure 6.4: Location result

Here the results from the parking and location experiments are presented together. The parking question had four possible answers, while the rest of the questions have three. For the parking question, the answers were either near me or in the corner of the living room. However, this opinion might change if the participant is with the robot for a longer time. In the question regarding what the robot should do while the participant was in the kitchen, most people answered that the robot should be near. This goes against an assumption of this thesis that the robot should be out of the way when in the kitchen. As with the last question, this might change over time. The trend in answers continues with the questions regarding the living room and bedroom. However, in regards to the question of where the robot should be when the person is in the bathroom, it changes. This is likely out of privacy, which is natural. However, for one participant, the robot should always be near him. This was because he had experienced an incident in the bathroom, where it took some time for him to get help. He reasoned that if the robot could be there for him in such a situation of need to call for help when he can't, it did not matter what he did.

6.1.2 Godspeeds results

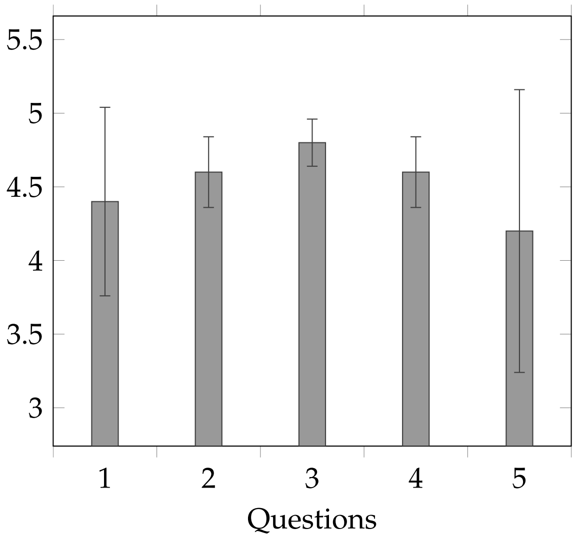
When using the GQS it is recommended to calculate Cronbach's alpha. However, because of the small sample size, it is not possible to test the reliability of the questionnaire[7]. The findings of the questionnaire will be presented, however, it should be noted that it is not representative. The questions where asked in regard to the Turtlebot3.



	Participants				
Q	1	2	3	4	5
1	3	3	4	3	5
2	2	4	3	3	4
3	2	2	2	1	1
4	2	3	2	4	2
5	4	3	3	1	5
6	4	5	2	3	5

Figure 6.5: Animacy

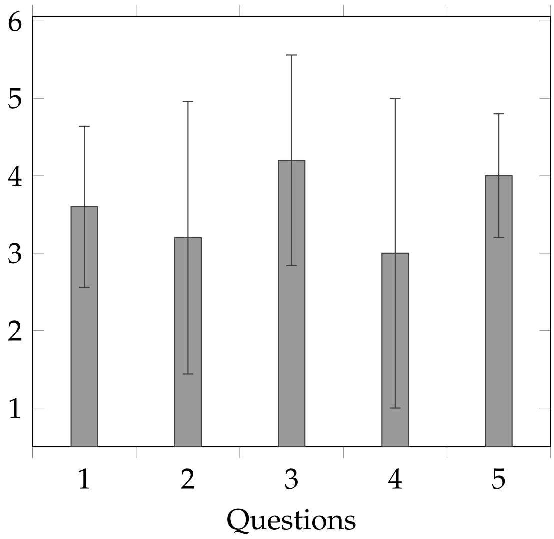
This graph shows how this group perceived the robots liveliness. The lower the score the more robot it is perceived as. For the two first questions regarding if the robot is dead or alive and stagnant or lively, the answers are indecisive. However, in regard to the robot being mechanical or organic, it is a decisive mechanical. A reason for why this might be is that the robot was presented in its dev kit state. Meaning all PCBs and wiring where visible. If this experiment was to be done again, a shroud to cover these parts could give another answer. This explanation could also hold true for the next question of artificial or lifelike. Here the response tends towards artificial. The last to questions on inert or interactive and apathetic or responsive does show how each individual experienced the experiments. Some found the robot to do actions that they welcomed, some did not.



Q	Participants 1	2	3	4	5
1	3	5	4	5	5
2	4	5	4	5	5
3	4	5	5	5	5
4	4	5	4	5	5
5	3	5	3	5	5

Figure 6.6: Likeability

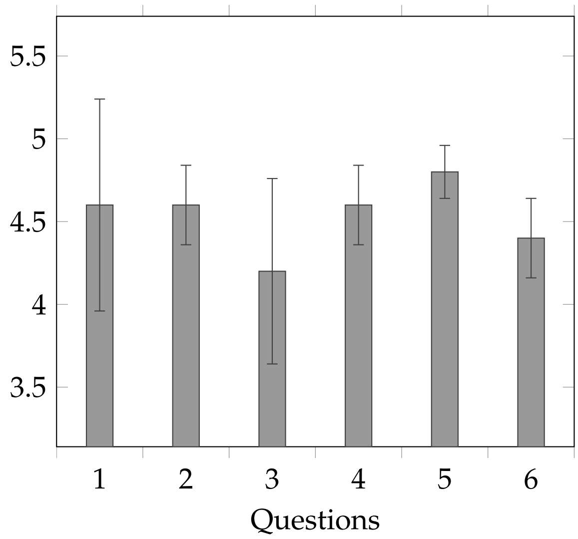
In the graph regarding likeability, the answers are quite uniform. However, the perception of an object is something that can change over time. Annoying behavior may show up in the form of sound or motion that the user does not want out of the robot. It could, therefore, be beneficial if the robot took part in the life of a participant over a longer stretch of time. This might give a more accurate account of likeability. However, this does serve to show the first impression that the group had of the robot.



Q	Participants 1	2	3	4	5
1	4	4	2	3	5
2	4	5	1	3	3
3	4	5	2	5	5
4	3	4	2	1	5
5	3	5	3	4	5

Figure 6.7: Perceived Intelligence

Perceived intelligence is also something that might change over time. In the same argument as with likeability, unwanted behaviors that the robot has might not be apparent at the start. These numbers show the reaction to the individual experience with the robot at a first meeting. The numbers show that the participants do not agree on what intelligence the robot has. Prior knowledge of robotics might contribute to this or expectations that were or was not meet.



Q	Participants 1	2	3	4	5
1	5	5	3	5	5
2	4	5	4	5	5
3	4	4	3	5	5
4	4	5	4	5	5
5	4	5	4	4	5
6	4	5	4	4	5

Figure 6.8: Perceived Safety

The questions regarding perceived safety were asked two times. It was asked at the start of the experiment and at the end of the experiment. The first three questions are in regards to at the start. When it comes to perceived safety, all the participants felt quite safe. The numbers and graphs also reflect this. This might in large part be because the experiment was conducted in a group. Otherwise, there is not much difference between the perceived safety at the start and the end of the experiment.

6.2 Training and Classification

This section will present the two classifiers that where tested The result that they produced and a comparison between our results and the results of others.

6.2.1 Data preparation

The data was prepared in two files, one for the labeled data and on for the non-labeled data. All the data was labeled either +1 or -1, regarding whether it was labeled or non-labeled data. These two datasets where then brought together and randomized. After the randomization, the data was partitioned width a ratio 60:20:20 for the training, testing and valid respectively. Each of the sets containing approximately 50% of each class. For the Support Vector Data Description (SVDD) classifier the data was also scaled to a value between $[0, +1]$. As it is a one-class classifier all the non-labeled data was also removed from the training set.

6.2.2 AdaBoost

The AdaBoost classifier was trained on the training set, where the parameters for the classifier was set through trial and error on the test set. Through testing, it was found that the type of AdaBoost classifier that gave the best result was the Gentle AdaBoost classifier. This boost type puts less weight on outlier data and is good with regression data.[13]. With further testing, the best result for the test set was found by setting the number of weak classifiers to 100 and the max depth of the tree to 30. The weight trim rate was left as is, at a default of 0.95. This function excludes all samples with a summary weight of $\leq 1 - Weight - trim - rate$ and is generally used to save computational time. After setting all the parameters the classifier was put on the validation set. This yielded an accuracy score of 85.7%. It is a slight increase from the best test scores.

Table 6.1: AdaBoost classification Confusion matrix

		Actual class	
		Person	No person
Predicted	Person	523 (85.32%)	88 (13.88%)
	No person	90 (14.68%)	546 (86.12%)

6.2.3 SVDD

To use SVDD, an implementation by Wang et al.[42] was used. This is a library for the LIBSVM library[24] program. The SVDD classifier was trained and the parameters for the width of the kernel was set through trial and error. For the regularization parameter C however, it was chosen to be 1 based on the recommendation from Tax and Duin.[39] This is because, there where no expected errors in the training set, as such all target data is accepted. For the kernel, the radial basis function (RBF) was used.

$$\begin{aligned} K(x^{(i)}, x^{(j)}) &= \phi(x^{(i)})^T \phi(x^{(j)}) \\ &= exp(-\gamma ||x^{(i)} - x^{(j)}||^2), \qquad \gamma > 0 \end{aligned}$$

This is the same as a Gaussian kernel, with the difference being that $\frac{1}{2\sigma}$ has been replaced by γ. The γ parameter was set to 16. In the end, this resulted in an accuracy of 77.63% when running the classifier on the validations set.

Table 6.2: SVDD classification Confusion matrix

		Actual class	
		Person	No person
Predicted	Person	485 (79.12%)	151 (23.82%)
	No person	128 (20.88%)	483 (76.18%)

6.2.4 Classification Results

Table 6.3: AdaBoost comparison

True Label	Detected Label	Result	Arras et al. corridor	Arras et al. office
Person	Person	85.32	99.58	97.45
	No Person	14.68	0.42	2.55
No Person	Person	13.88	1.03	2.73
	No Person	86.12	98.97	96.26
	Accuracy	85.72	99.01	97.27

Table 6.4: SVDD comparison

True Label	Detected Label	Result	Chung et al.	Jung et al.
Person	Person	79.12	97.1	-
	No Person	20.88	2.9	-
No Person	Person	23.82	9.4	-
	No Person	76.18	90.6	-
	Accuracy	77.63	-	98

The comparison table represents the results that are available. Even though these papers [38] [10] [23] use an implementation of the adaboost classifier the results are not given. However from the comparisons that we have, one can see that the result of the classifiers does not compare well to the result of others. One striking result is how the two classifiers used, compare to each other, where the AdaBoost outperforms the SVDD classifier, in every way. We will now go through some of the reasons that might have lead to this result. The one reason that probably has the biggest factor in this result is the LIDAR and its configuration. In the original paper by Arras et al. it is stated that they used a 180-degree SICK[1] laser range finder. It is not explicitly stated what resolution they used, but these LIDARs can be tuned to have a resolution of 0.25, 0.5 or 1 degree. The argument that they used a much higher resolution than 1 degree that was available to us, is also supported by the equipment of the other papers. The paper by [10] used Hokuyo UTM-30LX 2D laser range sensors with a resolution of 0.25, and The SPENCER project clearly stated that they used a LIDAR with 0.25 degree of resolution[23]. This is also supported by Jung et al. who used a LIDAR with 0.25 degrees of resolution [17]. On the other hand, Chung et al. used a LIDAR with a 1 degree of resolution [8]. This is where we come to another significant difference between the test. The height of the LIDAR in our case was at 19.5 cm, in the case of Chung et al. it was at a height of 28 cm. One of the reasons that this difference matter relates to the time of year our data was gathered. We collected our data during the winter time, and the participants wore winter shoes for the most part. Winter shoes come with a collar that stretches high up on the leg, and as such might have influenced our data. How high the LIDAR is mounted also differs in the other papers. Jung et al. used the features

[1]https://www.sick.com/se/en/

of the upper part of the body, with the LIDAR mounted at a height of 1.3 meters. Arras et al. had the LIDAR mounted 30 cm above the ground and Linder and Arras had the LIDAR mounted at 75 cm. From this one can conclude that it does not matter much where the LIDAR is situated. However one could argue that the features of the body change little if the surface area is large when different types of clothes are used. But if the surface area is small, like the lower half of the leg, a shoe does make a large difference. For the papers who used the AdaBoost classifier the features stated the same, however, the features for the ones using SVDD do vary. The paper by Jung et al. uses a combination of width, girth and width/girth, while the paper by Chung et al. used a combination of width and girth. In our case we used all 13 features for the SVDD classification, this might have given sub-par results. As many features in SVDD does increase computational complexity, it might have been better to use a subset. The environment may also have played an important factor in how the result turned out. As stated before, the data was gathered at university grounds in the department of informatics. Some of the items that may have contributed to the misclassification are chairs with metallic legs in a mirror finish. Mirrors have a tendency to bounce light around and as such given a wrong reading. Another item that may have contributed is the various sofas with a porous cover that distorts the range data. Other items where the various backpacks from the fellow master students with reflective parts. These items were noted when gathering data, however, it was decided to include them in the gathering process. This is because the items are part of the environment, and there are not few of them. Lastly, the number of people used to train the classifier. Chung et al. used 4 people. Jung et al. used 5 people. Arras et al. used 1 person in the first experiment and 2 people in the next. The amount of people we used, by comparison, is 25. Coming from the data, we were able to salvage. It should be stated that these papers aimed to find out if you could detect people with a LIDAR, not necessarily making a general implementation. It can also be argued that no two scans of a pair of feet will look the same, because of uncertainty in the data readings. However, since people do have different body types the uncertainty won't be enough to make up for this difference. As such the amount of people we used in our implementation might have affected the result.

What could have been done to improve these results? For the SVDD classifier, we could have done an extensive test in feature combination, as was done in [17]. As some people do prefer to use shoes indoor, mounting the LIDAR higher on the Turtlebot might have helped. The resolution of the LIDAR is also a problem, however, the cost of a LIDAR with higher resolution have a significant increase in cost. Since this project aims to have a low-cost solution this may or may not be viable, depending on the LIDAR and resolution. Besides Chung et al. did manage to get good results with a 1 degree of resolution LIDAR. As such another way to improve upon the results is to improve the segmentation algorithm before the classification. Chung et al. used a novel implantation in regard to this.

6.3 People Detection

In this experiment, we will test how far a person can be from the robot, while still be detected by it. The Turtlebot will be placed in the middle of a room. In this place, it will remain static only scanning the area. Then a distance of 3.5 meters will be measured from the robot to an open space in the room. From the robot to the 3.5-meter mark, every 50 centimeters will be marked with masking tape. On the software side, the robot will have access to the map and global position. What is specifically tested against is how well the system can detect the person. This means that the segmentation method plus the interactive_data_gatherer will be run. From this, we can draw a conclusion for the upper bound of detection. For the participant in this project, each 50 cm mark represents a position of detection to be tested against. Since the classification recognizes the segment of one leg and two legs these cases will be tested separately. For the first run through the participant will stand on one leg on each 50 cm mark position. How many LIDAR points that connect with the leg will be recorded. For the next run, the participant will stand on both legs, with the legs close enough together to be seen as one segment. Then the same actions as the first run will be done.

From the figure, we can clearly draw some conclusions. First of all the realistic detection range for the robot is up to 150 cm for a moving person. If the person is standing still, the range is somewhat better, with a max distance up too 2 meters. However, the graph represents the best possible results. As the table show, for one leg the number of points in a cluster

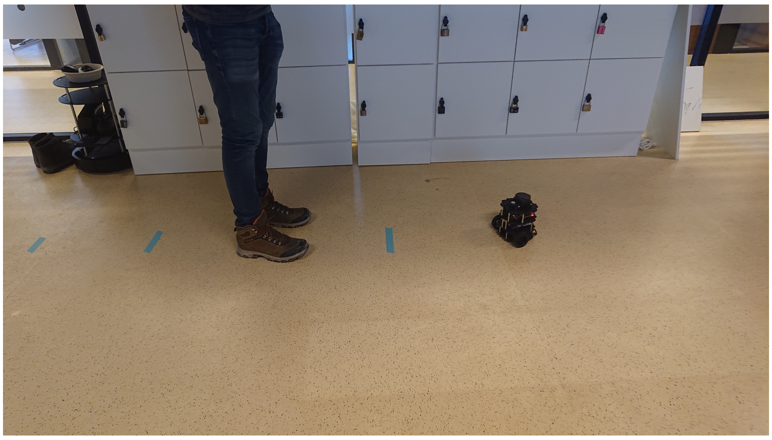


Figure 6.9: Distance experiment

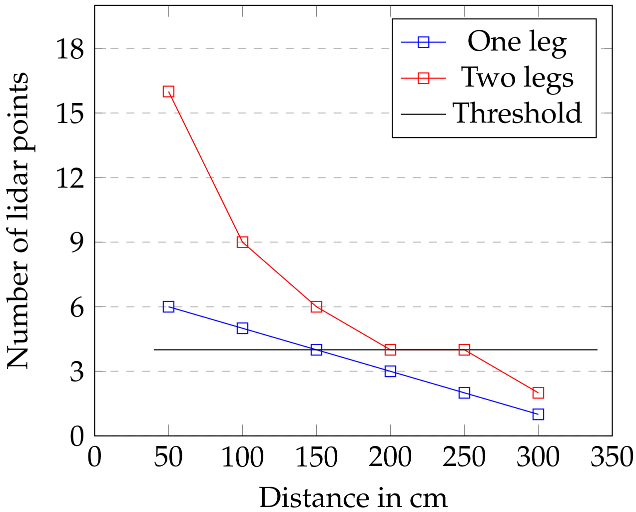


Figure 6.10: Clustering of point by legs at distance

Table 6.5: Clustering of point by legs at distance

	One leg	Two legs
50	6	16
100	5	9
150	3∼4	6
200	2∼3	4
250	2	4
300	1	3

varied between 3 and 4 at 150 cm. Meaning that for the classifier to successfully detect a leg, a person would have to be closer than 150 cm. When using the data gatherer, this view was reinforced. The software was only able to see one leg up to 100 cm and two legs up to 200 cm. Beyond that the LIDAR points were manually counted.

6.4 Navigation

6.4.1 How the robot handles a complex environment

The purpose of this experiment was to find out how the navigation system handled a cluttered environment. The environment was purposefully set up in such a way to be disadvantageous for the LIDAR system. With this setup, we can ascertain how the robot will handle itself.

The experiment was conducted in the eating area for the personnel at Ifi. This is a fairly open room with few things for the robot to stabilize its position. The robot was placed at the end of the open area and chairs were set up in front of the robot. These chairs where used because they have similar legs to the chairs at Kampen Omsorg+. It also represents another obstacle for the robot, manly that the legs are so thin that they may not be picked up by the LIDAR. The formation of the chairs was set up as such. All the chairs were facing the robot. This was because the non-facing side of the chairs had a binding element between two of the legs that were too low for the LIDAR to see. Straight in front of the Turtlebot, there was a chair, position in such a way that the robot could drive the legs. Right behind but at a far enough distance that the robot could turn was another chair. This chair positioned such that the robot would collide with the left leg. To the left of this chair was another chair with enough space between the two, to let the robot pass. This setup continued for another row. This formation would ensure that the robot could not drive straight through, but had to actively avoid the oncoming obstacles.

The Turtlebot was first made to map this entire area. To make sure the robot had the best opportunity to do well, it was made to drive multiple times under all the chairs. This, however, did not give good results. The mapper that was used in this case was Gmapping. Time and time again

Figure 6.11: Turtlebot in complex environment

the mapper would find the legs and would record them. However, when the robot moved a bit too far from the legs, the LIDAR would no longer see them. This made the navigation system overlook the for the most time. Or if the object was recorded, it would later be deleted from the map.

For the experiment, the robot was given an out of the box outlay. Nothing more than what was already present on the robot after the original configuration was used. The robot was also given some time to orient itself before the start. This was done using manual control. When the robot had asserted its position it was driven to the front most chair and given a position to drive through. This was done through the move_base system.

The robot tested a total of three times width this setup. Out of the three runs the robot managed to complete the course 2 times. On the second run, the navigation system could not find a possible course of action and subsequently stopped trying. The other two runs were plagued with a lot of difficulties. The robot would frequently find and obstacle through the local planner. It would then try to adjust for this by turning or driving a bit backward only for the local planner to lose track of the object it was trying to avoid. This made the navigation task quite slow as the robot would constantly try to correct itself. It also made the Turtlebot oscillate in place from time to time.

6.4.2 How areas of noninterference affect a complex environment

In this experiment, the same area set up as in the previous experiment was used. The navigation system, however, was set up with prior knowledge about the chairs. This came in the form of custom costmaps in the navigation stack. These custom maps were only given to the global costmap and not the local costmap. The local costmap could not correctly map the areas given correctly, because of this the robot would not navigate correctly.

The robot was placed in front of the foremost chair and given five tries. Although the robot was given more runs in this scenario, this happened because the robot failed each time. By having the robot run more times we could establish the cause of failure. Each time the robot would run the global planner would always choose an option to avoid all the chairs. However, the local planner would not see the chairs. This is because it did not have the information that the global planner had and the map was static. This meant that the local planner would influence the robot to drive closer and closer to the costmap areas. Eventually, the robot would collide with the leg and be stuck.

How to make the robot heed the global path planner. To make the robot avoid a predefined area on the map that is not there. To do this a simple box will act as an obstacle for the robot. The goal is to make the robot avoid this obstacle. Two tests will be done, one with a general setup of the robot and one where the goal_distance_bias for the dwa_local_planner is set lower. This weight control how much the controller should attempt to reach the local goal. Setting it lower will ensure that the robot will follow the global path more closely.

In the first test, the robot was given a costmap to avoid, otherwise, nothing had changed. This resulted in the global planner finding a path around the object while the local planner was trying to economize the rout. Because of this, the robot ended up cutting corners on the costmap.

In the next test, the goal_distance_bias parameter was set down from 20 to 15. This solved the issue of the robot cutting corners. The area was successfully avoided multiple times without incident. However, it should be stated that in arriving at this setting there were times the robot would drive into the enclosed area. This would result in the robot being trapped by its own navigation system, unable to escape.

For a further test, a chair was placed in the middle of the area. The Turtlebot was able to find a way, but the chair did disturb the path greatly. Along the way there wear frequent stops, and times when the robot oscillated trying to find a way. This was mostly due to the robot not knowing its position with great enough accuracy. After having the robot localize itself it did not pose a problem. The subsequent tests ended all good avoidance of the area.

6.4.3 How the navigation handles complex noninterference areas

With this experiment, we want to test out how the standard navigation system handles a complex environment of noninterference areas. The purpose is to see if the navigation system can be bated into taking a suboptimal route. This might happen because the navigation is too greedy and ends up spending time in convex areas. The robot will be allowed to have a map of the room. It will have time to orient itself in regards to localization. After orientation, it will be driven to a predefined starting point. This point will be the same for all variants of the experiment. The full framework system will be run, however, person detection and action areas will be turned off. First, there will be a sanity check. The robot will drive unhindered to its goal point. In the next test, the robot will be subjected to a horizontal line as an obstacle. This is to ensure that the robot doesn't just drive straight through the obstacle. In the final run, the robot will be med with a concave shape, where the hollow part is aimed towards the robot.

For the first test, the robot finished its task with no problem. In the second test, when the robot was presented with the line, it chose to go through the nearest exit to get to its goal. This was at the top of the line where the noninterference area of the wall and the line collided. This slowed the robot down somewhat but it made it in the end. For the final test, the robot was presented a concave shape. The shape was as wide as we could draw it on the map without not leaving space on either side for the robot to pass through. When presented with the object, the robot quickly found a way around it without it ever posing a problem. It ended up cutting a corner of the object because of the local planner, but it did not pose a problem. Since the concave object did not pose a problem here, it should not pose a problem even if it were a larger size. At least if we take into account the size of the average living room. Result can be viewed in figure 6.12.

This experiment was done in the motion capture lab at Ifi. In this lab, for the experiment, the robot was driving on a carpet. At first this did not seem to be an issue, however, the robot would latch on to the carpet from time to time. This happened because of the metal ball under the robot that stabilizes it. The ball would dig into the carpet, then when enough force was applied from the wheels it would detach. This made the robot jump, balancing on its two wheels. Although it never fell over, it did mess with the localization of the robot, giving the robot an offset in position.

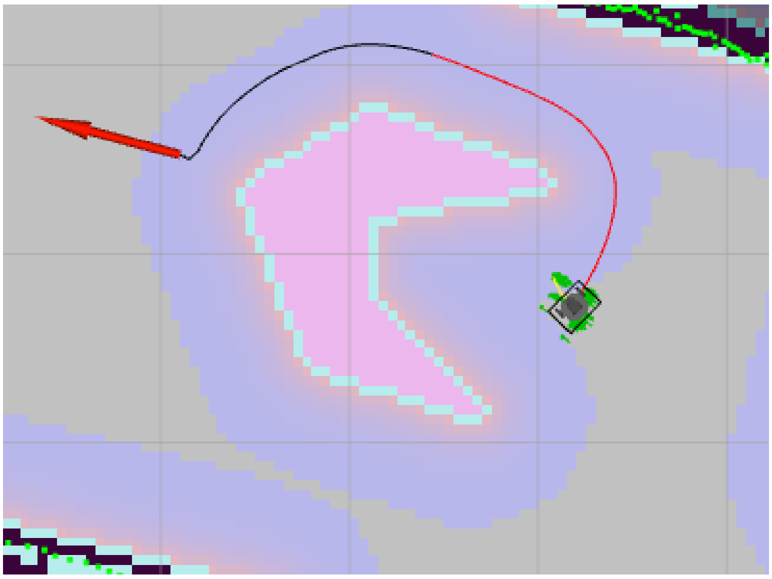


Figure 6.12: Robot avoiding convex area

6.5 Preset action

6.5.1 How does the robot handle going to a action area

This experiment was set up in the robotics room. An area in the middle of the room was sectioned off to function as an action area. The robot was placed on the right side of the room, to the right of the action area. There were no obstacles for the robot in this experiment. For this setup, the robot would be given a position to move to initially. While the robot is halfway through moving, a person would step into the action area. For this experiment to be a success the robot should start moving to a preassigned position.

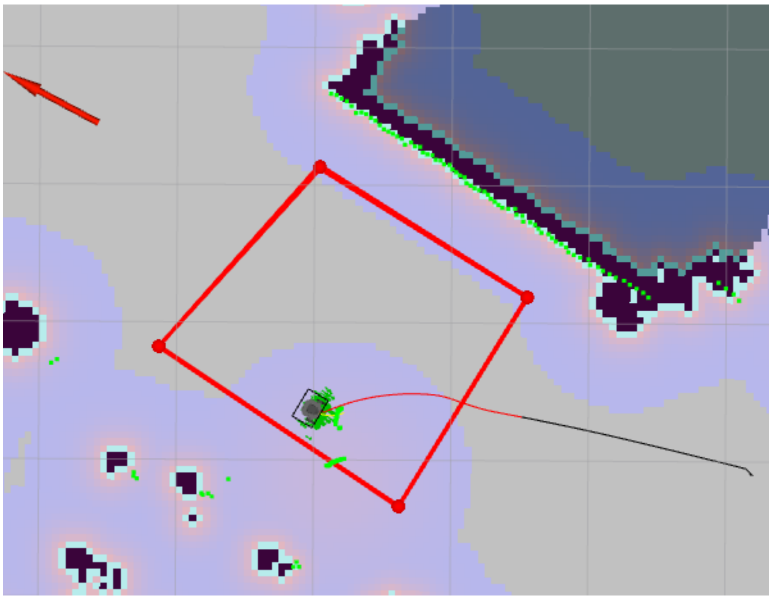


Figure 6.13: Robot responding to action area

For this setup, the Turtlebot will have the social_navigation_layer in the navigation stack. This is purely for visual confirmation that the Turtlebot has detected a person. The nodes detector, single_tracker and monitor_client will also run. In figure 6.13 the social area around the person can be viewed. The person is indicated by two green clusters close together by the robot inside the red area. And the social area is the blue space extending out from these cluster in a circular fashion. The red area indicates the action area.

The Turtlebot was given a position to move to. Halfway through a volunteer stepped into the action area. The robot had some problems detecting the person, but otherwise, it successfully completed its task. It turned around and went back to a predefined position. The red arrow in figure 6.13 indicates the original goal for the robot. While the black line extending from the robot to the edge of the picture indicate the new goal.

6.5.2 How does the robot handle going to multiple action areas in succession

The experiment was conducted in the robotics room. There where no obstacles for the robot to avoid. In this setup there where two action areas, one on the right side of the room and one on the left. The action area on the left side had its goal position set to be in the action area on the right. The goal for the right action area was in the middle of the room. At the start of the experiment, the Turtlebot would be placed in the middle of the room. It would there be given a goal position to the left action area. For the experiment to be successful the robot had to do both tasks in one run. The setup for the Turtlebot was the same as the previous experiment.

After the Turtlebot was given its goal and was halfway done, a person walked into the left action area. When the robot detected the person, it immediately changed its goal and began turning. When the robot was halfway through its second journey, a person walked into the right action area. The robot stayed on course and did not stop until it reached its goal. When the goal was reached, the person was detected and the robot moved to its final goal. The behavior of the Turtlebot is due to how the node monitor_client is set up. This can easily be changed for more active behavior, by changing how often it is supposed to detect people.

Chapter 7

Further Works

This chapter will present the various improvements that can be done to the system. The aim here is to point out deficiencies to make it a more suitable robot for the task of elderly care. First, there will be a section on the various hardware improvements that can be done. It will include what should be added to the robot and what should be upgraded. The last section will cover the software implemented in this thesis and what can be done to improve it.

7.1 The Hardware

In this thesis a framework for the turtlebot3 burger has been developed. However the only way for a user to know what the robot is going to do, is to know the back end of it. Say that the robot has reached its goal and it is ready to be interacted with. Nothing on the robot currently conveys this, as was made apparent in the experiment with the elderly. For the elderly to know when the turtlebot was ready, we had to tell them. A fix for this could be easily implemented in the form of a light. This light can function in the same was as a traffic light for ease of adaptability for the user. However this does not work for all users, in particular those with red-green colorblindness, which is the most common[9] To reinforce what action the robot is going to do, a speaker spelling out the intention could be implemented.

After having implemented a speaker, it might also be beneficial to implement a microphone on the robot. This comes back to the fact that when an emergency happens, having someone to talk to might be reassuring. Having someone to talk to might also prevent further harm, in case of disorientation. This is a feature already implemented on a lot of panic buttons aimed at the elderly. However, a senior citizen might not always be able to press this button, if the worst comes to the worst. As part of the MECS project, other modules currently under development are intended to have the functionality to detect if something happens. Thus circumventing the need to press a button. An automatic solution will ensure that all possible to detect problems will be reported. As such bringing forth a safer environment.

A safer environment does not only come with the equipment needed to detect and notify. The robot also needs to be at the place of accident. When doing the experiment with the elderly it was made painfully clear how slow the robot is. It could not follow the elderly person as it always fell behind. The current robot is fine for development, but not for deployment in this regard. More powerful servos should be implemented to increase the current speed of the robot to a more reasonable 0.88 m/s at the very least[34]. This should ensure that the robot is at the place if anything happens. There is however something that could hinder the progression of the robot throughout the house. In a lot of houses, at least in Norway, it is common to have a high door frame, between the individual parts of the house. With the current setup of the robot, it is hard to traverse these places, if at all possible. This is admittedly a hard problem to solve. It either involves the removal of all high door frames that the user has or a redesign in the moving base of the robot. However, it should be kept in mind when for future works.

7.2 The software

When it comes to the software side of things, there are certain methods and techniques that were not utilized in this thesis. For instance, when training the classifiers, all of the features where utilized. However, by doing an exhaustive search of various combination of features, the results may have been better. That being said, there may be other features not represented here, that could also improve the classification. Some of these are the girth and depth, and relation between features, such as girth/depth as used in the paper by Jung et al.[17].

When it comes to the methods implemented in this thesis, there was only the utilization of only two classifiers. However, there may be other methods that are better suited. For instance, if you have enough data neural networks that have a lot of research going for it as of late, may be a good alternative. There are also other methods like Support-Vector Machines (SVM) who might give better results through the use of grid search. Among others, the OpenCV library[6] doe have other classifiers that make for easy implementation. This comes down to the structure of the classifier functions in OpenCV, which are similar.

In the implementation, there was also no true identification of two legs to a person. For what was needed for the framework, the movement of one leg was more than enough. However other features for the robot may require a better knowledge of where each leg are. For example, if one wants a better tracker, knowing the motion of the legs would be a good starting point. This was done in the paper by Chung et al.[8] where they modeled the gait cycle.

Chapter 8

Conclusion

In this thesis, a framework for a mobile robot was built. This framework enables the robot to detect a person through the used of a Gentle AdaBoost classifier. Track the detected person via a Kalman Filter. And react to action in predetermined areas. The framework has also been designed to be modular. Meaning that individual parts, like the classifier or tracker, could be switched out for other modules that do the same. When it comes to the classification, a large amount of data has been collected to differentiate human legs from everyday objects. Tools have also been implemented to efficiently perform this task. This tool can further be used, if the LIDAR is changed, to make a model for a new system.

A study was conducted into the preferences of the elderly for certain aspects of the robot. These are the speed, distance and certain actions given certain situations. The GQS was also used to get a better understanding of the perspectives and feelings surrounding the robot. Due to the lack of participants, the answers can't be used to say anything conclusive. However, mistakes and improvements have been pointed out for future work.

There are however this that could have been done better. For one the system could have been implemented to recognize two legs to a person. Currently, it does not interfere with the implementation as it is assumed that there is only one person present. The ability to track multiple people could also have expanded the functions of the robot. Like what to do in a situation when a person has a visitor or if the robot is to accommodate multiple people. Another aspect to improve is the questionnaire. It could have been more manageable and better prepared.

Goals that were not met is a reaction based on time of day. This, however, can be implemented easily, but a new format for how the action area data is saved would have to be implemented. Another goal was to make a system aimed at the elderly, where the specification had no limitation in mind. This, however, is not possible as some elderly like other people use wheelchairs. The system won't recognize the shape of a wheelchair as a person. For this case, the goal was a bit too lofty.

All in all, a large proportion of what was described in the introduction have been implemented. The implementation works and does perform given its hardware limitations. Lastly, experience was gathered from the interview with the elderly that can be important for further works.

Bibliography

[1] K. O. Arras, O. M. Mozos and W. Burgard. 'Using Boosted Features for the Detection of People in 2D Range Data'. In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. Proceedings 2007 IEEE International Conference on Robotics and Automation. Apr. 2007, pp. 3402–3407. DOI: 10.1109/ROBOT.2007.363998.

[2] Christoph Bartneck et al. 'Measurement Instruments for the Anthropomorphism, Animacy, Likeability, Perceived Intelligence, and Perceived Safety of Robots'. In: *International Journal of Social Robotics* 1.1 (1st Jan. 2009), pp. 71–81. ISSN: 1875-4805. DOI: 10.1007/s12369-008-0001-3.

[3] Marco Becerra. *ROS Package for Leg Detection with a Laser Scanner: Marcobecerrap/Edge_leg_detector*. 26th Feb. 2019.

[4] N. Bellotto and H. Hu. 'Multisensor-Based Human Detection and Tracking for Mobile Service Robots'. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39.1 (Feb. 2009), pp. 167–181. ISSN: 1083-4419. DOI: 10.1109/TSMCB.2008.2004050.

[5] A. Bergland and T. B. Wyller. 'Risk Factors for Serious Fall Related Injury in Elderly Women Living at Home'. In: *Injury Prevention: Journal of the International Society for Child and Adolescent Injury Prevention* 10.5 (Oct. 2004), pp. 308–313. ISSN: 1353-8047. DOI: 10.1136/ip.2003.004721. pmid: 15470013.

[6] G. Bradski. 'The OpenCV Library'. In: *Dr. Dobb's Journal of Software Tools* (2000).

[7] Mohamad Adam Bujang, Evi Diana Omar and Nur Akmal Baharum. 'A Review on Sample Size Determination for Cronbach's Alpha Test: A Simple Guide for Researchers'. In: *The Malaysian Journal of Medical Sciences : MJMS* 25.6 (Nov. 2018), pp. 85–99. ISSN: 1394-195X. DOI: 10.21315/mjms2018.25.6.9. pmid: 30914882.

[8] W. Chung et al. 'The Detection and Following of Human Legs Through Inductive Approaches for a Mobile Robot With a Single Laser Range Finder'. In: *IEEE Transactions on Industrial Electronics* 59.8 (Aug. 2012), pp. 3156–3166. ISSN: 0278-0046. DOI: 10.1109/TIE.2011.2170389.

[9] *Facts About Color Blindness | National Eye Institute*. URL: https://nei.nih.gov/health/color_blindness/facts_about (visited on 31/03/2019).

[10] Gonzalo Ferrer et al. 'Robot Social-Aware Navigation Framework to Accompany People Walking Side-by-Side'. In: *Autonomous Robots* 41.4 (1st Apr. 2017), pp. 775–793. ISSN: 0929-5593, 1573-7527. DOI: 10.1007/s10514-016-9584-y.

[11] Yoav Freund. 'Boosting a Weak Learning Algorithm by Majority'. In: *Inf. Comput.* 121.2 (Sept. 1995), pp. 256–285. ISSN: 0890-5401. DOI: 10.1006/inco.1995.1136.

[12] Yoav Freund and Robert E Schapire. 'A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting'. In: *Journal of Computer and System Sciences* 55.1 (1st Aug. 1997), pp. 119–139. ISSN: 0022-0000. DOI: 10.1006/jcss.1997.1504.

[13] Jerome Friedman, Trevor Hastie and Robert Tibshirani. 'Additive Logistic Regression: A Statistical View of Boosting (With Discussion and a Rejoinder by the Authors)'. In: *The Annals of Statistics* 28.2 (Apr. 2000), pp. 337–407. ISSN: 0090-5364, 2168-8966. DOI: 10.1214/aos/1016218223.

[14] Edward T. Hall. *The Hidden Dimension*. New York: Anchor Books, 1982. 217 pp. ISBN: 0-385-08476-5.

[15] Dirk Helbing and Péter Molnár. 'Social Force Model for Pedestrian Dynamics'. In: *Physical Review E* 51.5 (1st May 1995), pp. 4282–4286. DOI: 10.1103/PhysRevE.51.4282.

[16] Kai Hormann and Alexander Agathos. 'The Point in Polygon Problem for Arbitrary Polygons'. In: *Computational Geometry* 20.3 (1st Nov. 2001), pp. 131–144. ISSN: 0925-7721. DOI: 10.1016/S0925-7721(01)00012-8.

[17] E. J. Jung et al. 'Development of a Laser-Range-Finder-Based Human Tracking and Control Algorithm for a Marathoner Service Robot'. In: *IEEE/ASME Transactions on Mechatronics* 19.6 (Dec. 2014), pp. 1963–1976. ISSN: 1083-4435. DOI: 10.1109/TMECH.2013.2294180.

[18] Rudolph Emil Kalman. 'A New Approach to Linear Filtering and Prediction Problems'. In: *Transactions of the ASME–Journal of Basic Engineering* 82 (Series D 1960), pp. 35–45.

[19] K. Kidono et al. 'Pedestrian Recognition Using High-Definition LIDAR'. In: *2011 IEEE Intelligent Vehicles Symposium (IV)*. 2011 IEEE Intelligent Vehicles Symposium (IV). June 2011, pp. 405–410. DOI: 10.1109/IVS.2011.5940433.

[20] H. Kim, W. Chung and Y. Yoo. 'Detection and Tracking of Human Legs for a Mobile Service Robot'. In: *2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. 2010 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. July 2010, pp. 812–817. DOI: 10.1109/AIM.2010.5695844.

[21] Stephan Kurzawe. *Costmap_prohibition_layer*. Version 0.0.5. TU Dortmund - Lehrstuhl RST, 25th Jan. 2017.

[22] J. H. Lee et al. 'People Tracking Using a Robot in Motion with Laser Range Finder'. In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems. Oct. 2006, pp. 2936–2942. DOI: 10.1109/IROS.2006.282147.

[23] Timm Linder and Kai O. Arras. 'People Detection, Tracking and Visualization Using ROS on a Mobile Service Robot'. In: *Robot Operating System (ROS): The Complete Reference (Volume 1)*. Ed. by Anis Koubaa. Studies in Computational Intelligence. Cham: Springer International Publishing, 2016, pp. 187–213. ISBN: 978-3-319-26054-9. DOI: 10.1007/978-3-319-26054-9_8.

[24] D. V. Lu, D. Hershberger and W. D. Smart. 'Layered Costmaps for Context-Sensitive Navigation'. In: *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. Sept. 2014, pp. 709–715. DOI: 10.1109/IROS.2014.6942636.

[25] Matthias Luber, Gian Diego Tipaldi and Kai O. Arras. 'Place-Dependent People Tracking'. In: *Robotics Research*. Springer Tracts in Advanced Robotics. Springer, Berlin, Heidelberg, 2011, pp. 557–572. ISBN: 978-3-642-19456-6 978-3-642-19457-3. DOI: 10.1007/978-3-642-19457-3_33.

[26] Josh McGuire. 'Ethical Considerations When Working with Older Adults in Psychology'. In: *Ethics & Behavior* 19.2 (2009), pp. 112–128. ISSN: 1532-7019(Electronic),1050-8422(Print). DOI: 10.1080/10508420902772702.

[27] Eiliv Mørk et al. *Kommunale helse- og omsorgstjenester 2015 - Statistikk om tjenester og tjenestemottakere*. Rapport 2016/26. 2016–26, p. 54.

[28] Jorun Ramm. *Eldres bruk av helse- og omsorgstjenester*. Statistsisk analyse 137. 27th June 2013, p. 107.

[29] Robert E. Schapire. 'The Strength of Weak Learnability'. In: *Mach. Learn.* 5.2 (July 1990), pp. 197–227. ISSN: 0885-6125. DOI: 10.1023/A:1022648800760.

[30] Amanda Sharkey and Noel Sharkey. 'Granny and the Robots: Ethical Issues in Robot Care for the Elderly'. In: *Ethics and Information Technology* 14.1 (1st Mar. 2012), pp. 27–40. ISSN: 1572-8439. DOI: 10.1007/s10676-010-9234-6.

[31] Helen Sharp. 'Interaction Design : Beyond Human-Computer Interaction'. In: (2011).

[32] Vegard Dønnem Søyseth. '"Hey, I'm Walking Here!" - An Explorative Study of Spatial Encounters between Older Adults and Autonomous Robots'. In: (2017).

[33] L. Spinello, R. Triebel and R. Siegwart. 'Multimodal Detection and Tracking of Pedestrians in Urban Environments with Explicit Ground Plane Extraction'. In: *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. Sept. 2008, pp. 1823–1829. DOI: 10.1109/IROS.2008.4651109.

[34] Fiona F. Stanaway et al. 'How Fast Does the Grim Reaper Walk? Receiver Operating Characteristics Curve Analysis in Healthy Men Aged 70 and Over'. In: *BMJ (Clinical research ed.)* 343 (15th Dec. 2011), p. d7679. ISSN: 1756-1833. DOI: 10.1136/bmj.d7679. pmid: 22174324.

[35] Dan Sunday. *Inclusion of a Point in a Polygon*. 2000. URL: http://geomalgorithms.com/a03-_inclusion.html (visited on 19/03/2019).

[36] Yoonchang Sung and Woojin Chung. 'Human Tracking of a Mobile Robot with an Onboard LRF (Laser Range Finder) Using Human Walking Motion Analysis'. In: *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*. 2011 8th International

Conference on Ubiquitous Robots and Ambient Intelligence (URAI). Nov. 2011, pp. 366–370. DOI: 10.1109/URAI.2011.6145998.

[37] T. Taipalus and J. Ahtiainen. 'Human Detection and Tracking with Knee-High Mobile 2D LIDAR'. In: *2011 IEEE International Conference on Robotics and Biomimetics*. 2011 IEEE International Conference on Robotics and Biomimetics. Dec. 2011, pp. 1672–1677. DOI: 10.1109/ROBIO.2011.6181529.

[38] Z. Talebpour, I. Navarro and A. Martinoli. 'On-Board Human-Aware Navigation for Indoor Resource-Constrained Robots: A Case-Study with the Ranger'. In: *2015 IEEE/SICE International Symposium on System Integration (SII)*. 2015 IEEE/SICE International Symposium on System Integration (SII). Dec. 2015, pp. 63–68. DOI: 10.1109/SII.2015.7404955.

[39] David M.J. Tax and Robert P.W. Duin. 'Support Vector Data Description'. In: *Machine Learning* 54.1 (1st Jan. 2004), pp. 45–66. ISSN: 1573-0565. DOI: 10.1023/B:MACH.0000008084.60811.49.

[40] Sebastian Thrun, Wolfram Burgard and Dieter Fox. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents Series. The MIT Press, 19th Aug. 2005. 672 pp. ISBN: 978-0-262-20162-9.

[41] E. Trulls et al. 'Autonomous Navigation for Mobile Service Robots in Urban Pedestrian Environments'. In: *Journal of Field Robotics* 28.3 (1st May 2011), pp. 329–354. ISSN: 1556-4967. DOI: 10.1002/rob.20386.

[42] Leland Wang et al. *LIBSVM for SVDD and Finding the Smallest Sphere Containing All Data*. 24th Dec. 2016. URL: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/#libsvm_for_svdd_and_finding_the_smallest_sphere_containing_all_data.

[43] Astrid Weiss and Christoph Bartneck. 'Meta Analysis of the Usage of the Godspeed Questionnaire Series'. In: *2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 2015 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). Kobe, Japan: IEEE, Aug. 2015, pp. 381–388. ISBN: 978-1-4673-6704-2. DOI: 10.1109/ROMAN.2015.7333568.

Appendices

Appendix A

Questionnaire

The general and the Godspeed questionnaire used in the experiment at Kampen Omsorg+. Although it is here in English, a translated version was available during the experiment. However, it was only translated for the use of the questioner.

MECS Project Experiment Questionnaire

Subject: ………………………… Date: …………………….. Time: ………………………..

CANDY Experiment	**Remarks**
1. How did you feel about the speed of the robot? Too Slow 1 2 3 4 5 Too Fast	
2. How was the distance of the robot to you when it stopped moving? Too Close 1 2 3 4 5 Too Far	
3. What do you think about the robot's height? Too Low 1 2 3 4 5 Too High	
4. What do you want the robot to do after delivery? ○ Stay Still ○ Move back to original location ○ Move a bit away from me	
FOLLOW ME Experiment	
1. How was the distance of the robot to you while following you? Too Close 1 2 3 4 5 Too Far	
2. How do you feel about the robot following you? Awful 1 2 3 4 5 Great	
3. Where do you prefer the robot to be when following you? ○ Behind me ○ Beside me	
AVOID ME Experiment	
1. How do you feel about the speed of the robot? Too Slow 1 2 3 4 5 Too Fast	
2. How did you feel about this situation? Awful 1 2 3 4 5 Great	
3. How do you feel about the distance between you and the robot? Too Close 1 2 3 4 5 Too Far	
4. How do you want the robot to react when moving towards you? ○ Turn left or right ○ Stop immediately ○ Move backwards	

PARKING Experiment

1. Where do you want the robot to be parked when idle?
   - ○ Near me
   - ○ In a different room
   - ○ Always in a corner of the living room
   - ○ Always in the bedroom

LOCATION Experiment

1. Where do you prefer the robot to be when you are in the kitchen?
   - ○ Near me
   - ○ Not in the kitchen
   - ○ Always in the parking spot
2. Where do you prefer the robot to be when you are in the living room?
   - ○ Near me
   - ○ Not in the living room
   - ○ Always in the parking spot
3. Where do you prefer the robot to be when you are in the bed room?
   - ○ Near me
   - ○ Not in the bed room
   - ○ Always in the parking spot
4. Where do you prefer the robot to be when you are in the bathroom?
   - ○ Near me
   - ○ Not in the bathroom
   - ○ Always in the parking spot

Anthropomorphism
Please rate your impression of the robot on these scales:

Fake	1	2	3	4	5	Natural
Machinelike	1	2	3	4	5	Humanlike
Unconscious	1	2	3	4	5	Conscious
Artificial	1	2	3	4	5	Lifelike
Moving rigidly	1	2	3	4	5	Moving elegantly

Animacy
Please rate your impression of the robot on these scales:

Dead	1	2	3	4	5	Alive
Stagnant	1	2	3	4	5	Lively
Mechanical	1	2	3	4	5	Organic
Artificial	1	2	3	4	5	Lifelike
Inert	1	2	3	4	5	Interactive
Apathetic	1	2	3	4	5	Responsive

Likeability
Please rate your impression of the robot on these scales:

Dislike	1	2	3	4	5	Like
Unfriendly	1	2	3	4	5	Friendly
Unkind	1	2	3	4	5	Kind
Unpleasant	1	2	3	4	5	Pleasant
Awful	1	2	3	4	5	Nice

Perceived Intelligence
Please rate your impression of the robot on these scales:

Incompetent	1	2	3	4	5	Competent
Ignorant	1	2	3	4	5	Knowledgeable
Irresponsible	1	2	3	4	5	Responsible
Unintellegent	1	2	3	4	5	Intelligent
Foolish	1	2	3	4	5	Sensible

Perceived Safety
Please rate how you felt on these scales at the beginning:

Anxious	1	2	3	4	5	Relaxed
Agitated	1	2	3	4	5	Calm
Quiescent	1	2	3	4	5	Surprised

Please rate on these scales how you felt towards the end:

Anxious	1	2	3	4	5	Relaxed
Agitated	1	2	3	4	5	Calm
Quiescent	1	2	3	4	5	Surprised

Appendix B

Consent form

The consent form provided to all participants of the experiment at Kampen Omsorg+. Since the consent form was to be provided to each participant for them to read at their leisure it was provided in Norwegian.

MECS
Multimodal Elderly Care Systems

Informert Samtykke til Sensoraktiviteter

Bakgrunn og formål

Vi ønsker å skape modeller av menneskelig adferd og tilstand slik at vi kan bedre analysere og forutse når eldre mennesker som bor hjemme trenger hjelp. Dette vil hjelpe dem til å bli hjemme og uavhengig lenger.

Om MECS

MECS er et pågående forskningsprosjekt ved Universitetet i Oslo, avd. Institutt for informatikk. Prosjektets formål er å undersøke bruk av informasjons- og kommunikasjonsteknologi (IKT) for å hjelpe eldre til å være trygge hjemme. Vi bruker brukersentrert design for å utvikle en effektiv forståelse av hverdagens aktiviteter, samt utvikle læringsmetoder for å forutsi uønskede hendelse. På denne måten vil prosjektet demonstrere mulighetene forstå avvik fra forventet handling og oppførsel, og på den måten gi økt sikkerhet og personvern som kan hjelpe eldre til å bo hjemme lenger.

Målgruppen for studien er eldre mennesker som bor hjemme. Siden vi vil sørge for at ting fungerer bra, tester vi sensorer i et laboratoriemiljø.

Hva skal du gjøre i denne studien?

Vi registrerer din reaksjon og tilbakemelding ved å ha en robot hos deg. Aktivitetene vil bli tatt opp av kameraer og ultrabrede båndsensorer. Kameraene registrerer vanlig video, termiske bilder og dybdeinformasjon. De ultrabrede båndene og bærbare båndsensorer registrerer tilstedeværelse, åndedrett og hjerterytme. Video og lyd vil også bli tatt opp.

Hva skjer med informasjonen om deg?

Informasjonen som samles inn fra sensorene vil bli brukt til å lage modeller som kan brukes til å kjenne igjen hendelse, oppdage hendelser og forutsi hendelser. All personlig informasjon vil bli behandlet konfidensielt. Informasjonen vil bli lagret på en kryptert harddisk. Etterpå blir det lastet opp til Universitetet i Oslos tjeneste for sikker informasjon (TSD), der bare utvalgte medlemmer av prosjektet har tilgang. Navn, alder og annen personlig informasjon vil ikke bli samlet.

Prosjektet er planlagt avsluttet til 31.12.2020. Alle opptak blir slettet etter denne datoen.

Frivillig deltakelse

Deltakelse er frivillig, og du kan trekke ditt samtykke til enhver tid uten å gi noen grunn. Hvis du trekker tilbake dit samtykke, vil all informasjon om deg bli slettet. Har du spørsmål om dette prosjektet, vennligst kontakt Weria Khaksar +47 46216735 eller professor Jim Tørresen + 47-22852454.

Studien er rapportert til personvernombudsmannen for forskning, NSD-Norsk Senter for Forskningsdata AS.

Samtykke til deltakelse i studien

Jeg har mottatt og forstått informasjonen om studien, og jeg er villig til å delta (Ja) ______

Deltakerens fulle navn: ____

_______________ ____________________________________

Sted, dato Signatur av deltaker

____

Signaturen til representanten for MECS-prosjektet