Radbound University

Bachelor's Thesis in Artificial Intelligence

How Knowledge about the Environment Influences the Navigation of a Robot through a Crowd

Author: Marit Hagens (s4808061)

Supervisor: Serge Thill[1]

June 28, 2019

Nijmegen, The Netherlands

[1] Donders Institute for Brain, Cognition, and Behaviour, Radboud University, Nijmegen, Netherlands

Table of Contents

Abstract

Robots are integrating more into society and some of them have to navigate through crowds. For those robots it is important to know how much knowledge they need about the environment to efficiently navigate. This Bachelor's thesis examines how the amount of knowledge about the environment influences the navigation of a robot through a crowd. The influence has been researched using a simulation created in Unity. The knowledge has been provided by the Navigation Mesh (NavMesh) which the robot used to navigate. The size of the NavMesh has been varied to change the amount of knowledge. The influence of the knowledge has been researched in different environments with different properties, such as static and dynamic obstacles and narrow and broad passages. The size of the crowd has also been manipulated to see if that influences the effect of knowledge. The results show that the knowledge has no systematic influence on the number of collisions. It does affect the time a navigation takes in broad environments, narrow scenes show no effect. However, the additional benefit of more knowledge is quite minimal. This concludes that a robot's navigation through a crowd can do relatively well with relatively little knowledge.

1 Introduction

Robots are integrating more and more into society. They are already used in many different places, experimentally or in real life. E.g., social robots are being used in therapy for children with autism (Cabibihan, Javed, Ang, & Aljunied, 2013), assistive robots in elderly (health)care (Koceski & Koceska, 2016; Manti, Pratesi, Falotico, Cianchetti, & Laschi, 2016; Shiomi, Iio, Kamei, Sharma, & Hagita, 2015) or guide robots at an airport (Joosse & Evers, 2017). For some of these robots, it is important to navigate through a crowd, such as the guide robot at an airport (Joosse & Evers, 2017) or the wheelchair robot with social behaviours for the elderly (Shiomi et al., 2015). This can be crowds of humans, but also other robots, animals or a combination. Because the participants in the crowd can have a high variety, they all will be referred to as agents. Especially moving through a human crowd should go smoothly for the robot. When accidents often occur between a robot and humans, humans will not accept robots in the society. This means that the robot should move through the crowd without bumping into any agent of the crowd. Even if an agent makes a sudden direction change, the robot has to react and still navigate smoothly.

Moving through a crowd can be very difficult for a robot. Lots of research has already been done about robot crowd navigation. A common problem is the freezing robot problem. If the environment and the crowd become to complex, the path planner stays at its place, because all moves forward are unsafe (Trautman & Krause, 2010). Different solutions for the problem have already been researched. Trautman, Ma, Murray, and Krause (2015) try to solve it by anticipating human cooperation, using interacting Gaussian processes. They concluded that it is critical to use a cooperation model for safe and efficient navigation of robots in human crowds. A different approach was taken by Gelbal, Altuğ, and Keçeci (2016), they use interaction to solve the problem. An intelligent autonomous transport vehicle moves along pedestrians in different environments. When the vehicle approaches a pedestrian and there is no way around him, the vehicle asks the person for help. This way the vehicle will not freeze and has a smooth navigation through the crowd. As shown, the freezing robot problem has been researched quite broadly. That is why it is not necessary to solve the problem in this thesis.

Other researchers looked at optimizing the navigation of the robot through the crowd. Savkin and Wang (2014) present an algorithm for collision free navigation in unknown complex environments, with moving obstacles. The algorithm does not require information about the shape and velocity of the obstacles, which makes it very robust and useful in lots of cases. As result they found that the robot seeked a free path through the crowd instead of avoiding the crowd. Park, Ondřej, Gilbert, Freeman, and O'Sullivan (2016) created an algorithmic framework which is able to classify human intentions early. With the use of the algorithm, the robot can predict the movement of the humans and navigate through the crowd. They found that the algorithm made the crowd navigation more efficient and safer. Bera et al. (2019) presented another algorithm, where the robot uses the emotions of the pedestrians around him. To classify the emotion, they look at the face and trajectory of a person. The emotion is predicted multiple times using different models. The results are combined into a multi-channel model to classify the emotion of the person into one of the four emotion categories. The emotions of all pedestrians are combined with path predictions for socially normative, collision-free robot navigation. Taking emotion into account, the social comfort of the crowd and the navigation through it is improved.

As shown above, the freezing robot problem and optimizing the navigation of a robot have already been researched quite often. For that reason, this thesis will not be about either of those. A big problem in robotics are the sensors. Most sensors are highly affected by noise, especially small and lightweight sensors (Hornung, Wurm, & Bennewitz, 2010). Li, Jiang, Ge, & Lee (2018) explicitly take the noise and limitations of the robot's sensor measurements into account during evaluating their results. The sensors and its limitations influences how much knowledge of the environment the robot has. However, it is unclear how this knowledge influences the navigation of the robot. This thesis will be about this specific influence. To see if a robot who evaluates almost every agent and object in the environment reaches its goal faster and safer than a robot who only sees agents close by. The amount of knowledge a robot needs influences the type of sensors it requires. A simple distance sensor would probably suffice, if a robot navigates best while it only takes the closest agents into account. However, if a robot needs the location of almost every agent to properly navigate, a complex sensor will be necessary. That is why the amount of knowledge a robot has about the environment is important. This will be examined by answering the following research question: "Does the amount of knowledge about the environment improve the navigation of a robot through a crowd?" This leads to the following hypotheses:

- H_0: The amount of knowledge has no influence on the navigation of a robot through a crowd.
- H_1: More knowledge improves the navigation of a robot through a crowd.

The knowledge will be represented by everything the robot uses to navigate to the goal. This is the vision of the robot, shaped in a triangle, called the visual triangle. Everything inside of the triangle is the knowledge the robot has. The size of the triangle is varied to change the amount of knowledge. How the visual triangle works and what the robot perceives will be explained in section 2.5, “Robot Knowledge”. The knowledge is not obtained from real robot sensors. However, the goal of the thesis is to find if knowledge influences navigation at all. How the knowledge is obtained is not the aim of the research and for that reason also not examined. If an effect is found, further research can be done with sensors which can be used in real life.

The research question will be answered using a simulation in which the vision of the robot is manipulated. The simulation and experiment will be explained in section 2, “Methods”. Section 3, “Results”, shows the results of the simulation and experiment. The results will be discussed in section 4, “Discussion”, and a conclusion about them and the answer to the research question will be in section 5, “Conclusion”. Finally, the last part of section 5 will be about further research in response to the experiment and results.

2 Methods

The influence of the amount of knowledge a robot has while navigating through a crowd will be examined using a self created simulation in the game development program Unity (Unity Technologies, 2019). Unity will be explained in section 2.1. After that, the three most important objects of the simulation will be explained in section 2.2. These are the robot, the human agent and an obstacle. The simulation will run in three different environments. The first environment will simulate an office corridor, where a small number of agents fit next to each other. The second one will simulate a wide sidewalk, where many agents fit next to each other. The last environment is the wide sidewalk with static obstacles. All three environments will be further explained in section 2.3. To navigate, the robot and human agents use A* search with a Navigation Mesh (NavMesh). Section 2.4 will discuss how the navigation works and why this algorithm is used. Section 2.5 will explain how the NavMesh is used to manipulate the robot’s knowledge about the environment. The experiments in the simulation will be explained in section, 2.6. The last section, section 2.7, explains how the data obtained for the experiment will be analysed.

2.1 Unity

The experiment is performed in a self made simulation using Unity 2018.3.12f1 Personal. Unity (Unity Technologies, 2019) is a tool for game development, which is easily operable and supports multiple platforms (Chiu & Shiau, 2016). This makes Unity also very useful in making simulations. Unity has a various amount of physical characteristics, such as materials, mass, gravity and collision detection, already implemented, which makes it easy to create real-world scenes.

GameObjects are the basic software modules in Unity. They provide the visual appearance, all the physics, the interaction and animation of an object (Codd-Downey, Forooshani, Speers, Wang, & Jenkin, 2014). These GameObjects can be prefabricated (Prefab), so that they can be instantiated while running the application. Prefabs are generalized templates that can be easily customized (Wood, Margenet, Kenneally, Schaub, & Piggott, 2018). Changing the Prefab will change all the instances of the Prefab in the scene. This is very useful if you have the same object multiple times in a scene. The next section will describe the Prefabs created and used for the simulation.

2.2 Prefabs

For the simulation, different objects are used, such as a floor, goals, agents and obstacles. The objects which are used more often, are turned into Prefabs. The most important Prefabs for the simulation are the robot, the human agent and the obstacle. These are needed multiple times in the same scene and also in different scenes. For that reason, they are made into Prefabs. These objects and their characteristics will be explained below.

2.2.1 Human Agent

The experiment is about how knowledge affects navigation through a crowd. A crowd consists of many agents, this makes a Prefab very useful. With the Prefab, only one human agent has to be created and that one can be multiplied numerous times. The human agent, figure 1, is a red cylinder with height 2 and diameter 1. The speed of the human agent is 3.5 world units per second. The angular speed is 120 degrees per second. The goal of the human is to navigate to the bottom of the scene, where the robot starts. The human agent will start walking if the robot is within a distance of 20 from him. This way, not all human agents will pass the robot in the beginning of the scene. The human uses A* search on a NavMesh, which will be further explained in section 2.4, to navigate to its goal. With a chance of 10%, the human will step towards the side. This is to keep the behaviour of the human less predictable for the robot.

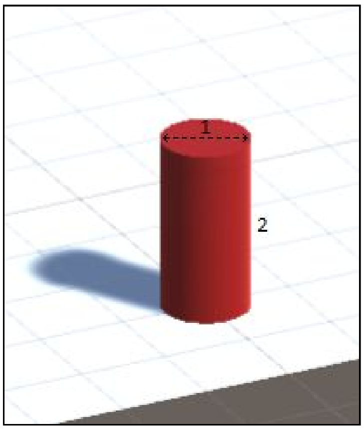

Figure 1. *The human agent Prefab with dimensions.*

2.2.2 Robot

The second important object for the experiment is the robot. The robot is required in every simulation of every scene. The characteristics of all robots should be the same, which is what a Prefab guarantees. The robot, figure 2, is a purple cuboid with height 2 and width 1. The shape and colour are distinct from the human agents, such that it is easy to keep them apart in the simulation. The speed of the robot agent is 2.5 world units per second. The angular speed is 60 degrees per second. The speed of the robot is lower than the human, because that makes the robot less intimidating towards humans. When something or someone comes towards you a lot faster as you, that might be frightening and intimidating for you. The goal of the robot is to navigate to the other side of the scene as fast and save as possible. The robot also uses a NavMesh with A* search, section 2.4, to reach its goal.

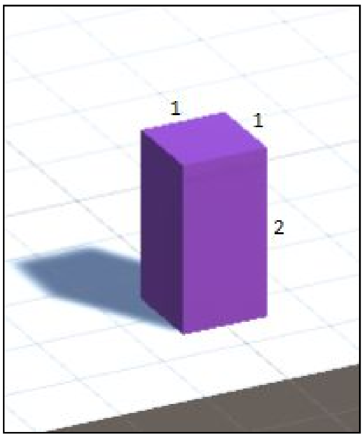

Figure 2. *The robot Prefab with dimensions.*

2.2.3 Obstacle

The last important object for the experiment is the obstacle. Obstacles are used in two of the three environments. They are just used as obstacles in the "Obstacles" scene, to make the navigation for the robot more difficult. In the "Office", they are used as walls to create the offices where the human agents start in. The obstacle or wall, figure 3, is a light blue cuboid with height 3, length 2 and width 1. The mass of the obstacle is a lot higher as the mass of the robot and human agents. This is to ensure that it will not move or fall over if it gets bumped into. The NavMesh is baked around the obstacles, such that the robot and humans will walk around them to go to their goals. More information about the NavMesh can be found in section 2.4.

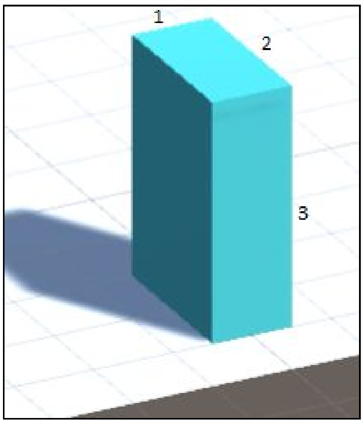

Figure 3. *The obstacle Prefab with dimensions.*

2.3 Environments

For the experiment, three different scenes are created in Unity. Each scene represents an environment. The environments have different characteristics to see the effect of knowledge in different situations. With these environments, situations from daily life are replicated. The environments and their characteristics will be explained in detail below. The length of all the scenes is the same, which is 51. The robot will have to navigate to its goal as optimal as possible. The starting position and the goal of the robot are at the same location in all the scene, (0, -23) and (0, 25) respectively. This way, the robot has the same distance as the crow flies to cover in all the scenes. The human agents have to navigate to the robot's starting position. This means that all the agents have to pass the robot. The agents are randomly generated in all the scenes. This way the robot never knows where the agents will come from and cannot take them into account before they are within the visual triangle.

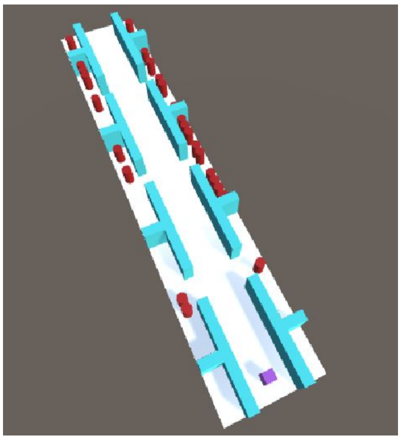

Figure 4. *The Office scene with 25 randomly generated human agents.*

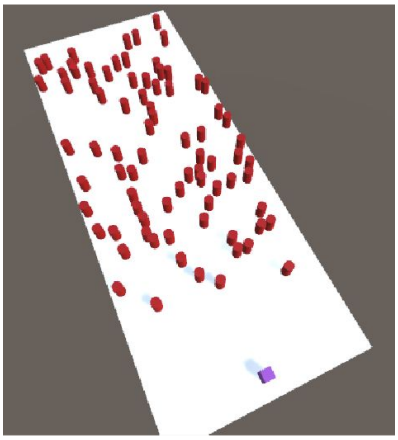

Figure 5. *The Street scene with 80 randomly generated human agents.*

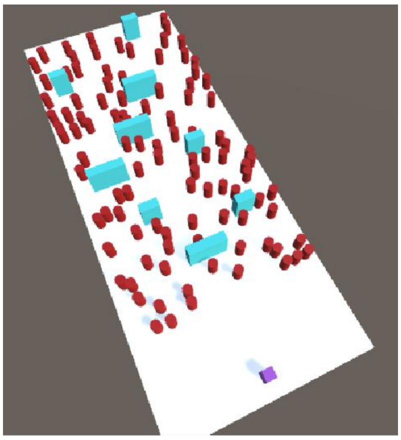

Figure 6. *The Obstacles scene with 105 randomly generated human agents.*

2.3.1 Office Scene

The first scene simulates an office corridor, which will be referred to as "Office" in this thesis. The Office, figure 4, has a narrow corridor, where only three agents fit next to each other to reach the other side. On the sides, offices are created with the obstacle Prefab. The agents are generated inside the offices. This way, the robot cannot see them until they come into the hall. This created unexpected situations for the robot, which makes the navigation more interesting. Because of the randomly generated agents, the robot does not know out of which office an agent, or multiple agents, will come.

2.3.2 Street Scene

The second scene simulates a broad sidewalk, referred to as “Street”. On the Street, figure 5, many agents fit next to each other. The robot has to take many agents into account and navigate through them. Depending on the vision, the robot sees more agents and knows where the biggest space is to pass through.

2.3.3 Obstacles Scene

The last scene simulates a big open area with static objects, referred to as “Obstacles”. The static objects can represent a variety of different real live objects. Varying from a bench in the park to a toy on the floor in a house. The scene, figure 6, has the basics of the Street scene, with some stationary obstacles in the way of the robot. The vision of the robot influences how many obstacles the robot sees and are taken into account while navigating to the end. When two obstacles are close after each other, the vision influences if one or both of them are seen. This can have consequences for the side the robot passes the first obstacle.

2.4 A* with Navigation Mesh

As search algorithm for the robot and the agents, A* will be used. A* search is widely used to find the shortest path between two points. The algorithm brings features of uniform-cost search and heuristic search together. A* examines the most rewarding, unexplored locations again and again (Barnouti, Al-Dabbagh, & Naser, 2016). To create these locations, Navigation Mesh (NavMesh) is used. Zikky (2016) concludes in its paper that NavMesh with A* search is the best solution for finding the shortest path in today’s industry. With NavMesh, the floor is divided into polygons, this is called baking the NavMesh. Using the NavMesh, the robot can move to its goal, in the straightest line possible. The start point needs to be in the same polygon as the goal to make the path. The robot start with its current polygon and repeatedly adds polygons which it needs to pass, until the goal is in the same polygon as itself. Then, a straight line is drawn between the start and goal as a path (Zikky, 2016). The A* algorithm is used to choose the best polygons to pass through towards the goal. Figure 7 shows the NavMesh of the human agents in the Obstacles scene. The obstacles create a gap in the NavMesh, because the agent is not able to walk there. Figure 8 shows the path of an agent created with the A* algorithm using the NavMesh. All the dark polygons are the polygons which the agent has to pass to reach the goal, the red cross at the bottom. The red line is the path which he will follow. The little blue line coming from the agent is the direction he is currently going in. When the agent walks further, the path gets extended to the end.

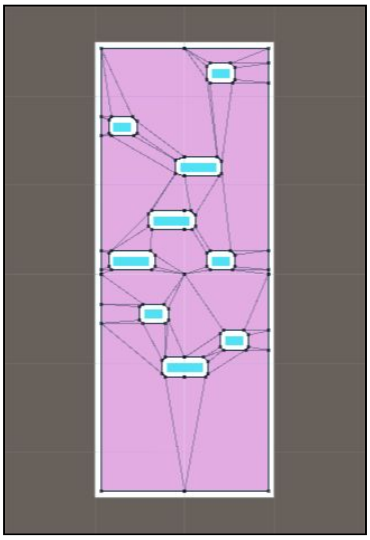

Figure 7. *The NavMesh of the human agents in the Obstacles scene.*

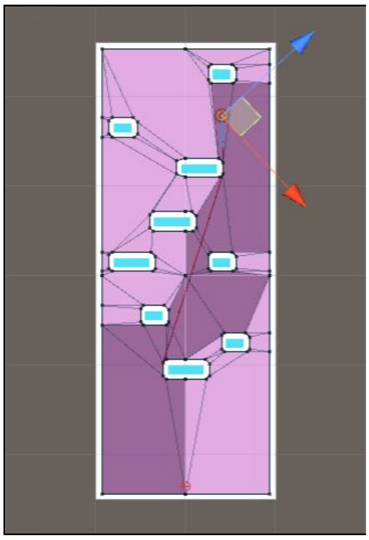

Figure 8. *The path of an human agent on the NavMesh of the Obstacles scene.*

2.5 Robot Knowledge

To answer the research question, the number of agents and obstacles the robot takes into account while navigating will be manipulated. This will be perceived by the robots vision and changes the amount of knowledge the robot has. The vision will be manipulated by baking the NavMesh only for a certain distance. The visual field of the robot has the shape of a triangle, the visual triangle. The vision of the robot is 90 degrees wide. The visual triangle is plotted in the simulation, to approximate for the user what the robot sees. However, this is only for the user, it is not what the robot uses to navigate. Underneath the visual triangle, the NavMesh is baked, which is used by the robot. The robot only uses the agents and obstacles on its NavMesh to navigate. These are the obstacles and agents the robot sees and has knowledge about. The visual triangle with a watch distance of 10 from the robot in the Office scene is plotted in figure 9. Figure 10 shows the NavMesh belonging to the visual triangle of figure 9. The NavMesh shows that the robot can see through the door of the office, but not the rest of the office.

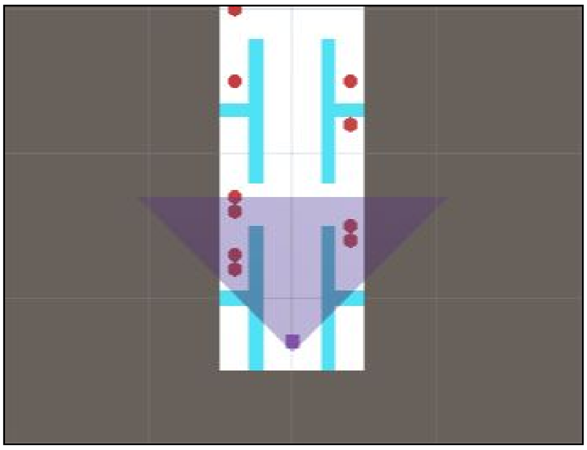

Figure 9. *The visual triangle of the robot with watch distance 10 in the Office scene.*

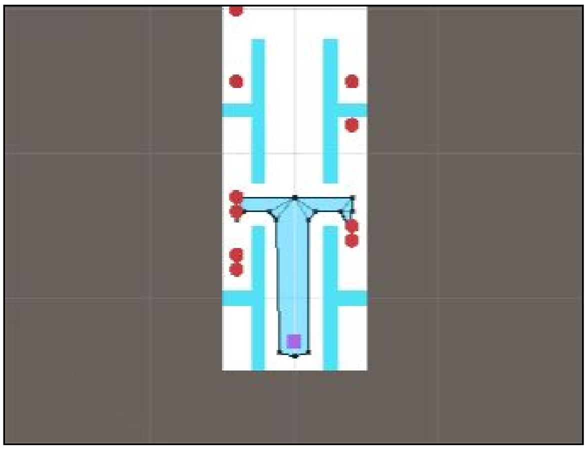

Figure 10. *The NavMesh of the robot belonging to the visual triangle with watch distance 10 of figure 9.*

2.6 Experiment

To answer the research question, an experiment will be performed using the simulation explained before. The knowledge of the robot is set by the distance the robot takes into account while navigating. For all three scenes, six levels of vision will be compared. Starting with a distance of 5 and incrementing to 30 with steps of five, giving 5, 10, 15, 20, 25 and 30. This will cover short distances, long distances and some steps in between. The watch distances are the same in all the scenes, because the total distance the robot has to walk in the scenes is also the same.

As illustrated in section 2.3, about the environments, three different environments will be experimented with. The effect of knowledge in different situation can be examined this way. The Office scene examines narrow spaces, where you can suddenly encounter agents on your path. In the Street scene, a big open space without surprises is considered. And lastly, the Obstacles scene examines open spaces with static obstacles. More knowledge might have different influences on these different aspects.

The third aspect that will be manipulated is the number of humans in the scene. This is to see how the effect of the knowledge is influenced by the size of the crowd. In bigger crowd, more knowledge might lead to situation which are to complex and unclear. All the information leads to more computations, which could make the pathfinding slower. On the other hand, in small crowds there is quite some space between the agents. More vision might not give a difference, because there are enough gaps between the agents to pass through for the robot. In each scene, five different numbers of human agents are compared. The numbers and steps are different, because the scenes are different in size and represent different situations. For each of them, an applicable number of humans is devised. They are chosen in such a way that the ratio of humans are approximately the same in all of them. The number of spots where a human agent can be generated is calculated for each scene and about 30% is taken as the middle level. This comes to 100 humans for the Street, 95 for Obstacles and 25 for the Office. For the Street and Obstacles, an interval of 10 humans is taken, this gives approximately the same ratio for both of them. The Office has a lower step size of 5 humans, because a significantly smaller number of humans fit in the scene. The ratio in the Office is a bit bigger as in the other scenes, otherwise the numbers of humans stayed too similar in the Office. In table 1 are the final numbers of humans displayed for each scene.

	Number of Human Agents				
Office	15	20	25	30	35
Street	80	90	100	110	120
Obstacles	75	85	95	105	115

Table 1. *The levels of number of human agents in each of the scenes.*

For the experiment, all the combination in the scenes are run 50 times. This gives an accurate average to evaluate the results. For each simulation, the time it takes the robot to reach its goal is stored. This is to see if more knowledge makes the robot faster. Furthermore, the number of collisions with a human agent, a wall/obstacle and the total number of collisions are stored, to be sure that the navigation was safe. If the robot was super fast but ran into every human agent it sees, then the navigation method was not optimal. An optimal consideration between speed and safety has to be made. The data of the environments are stored in separate data sets.

2.6.1 Baseline

There will also be a baseline comparison. In the baseline, the robot will have perfect world knowledge. The NavMesh will not be delimited, the robot can take every obstacle and human agent in the environment into account. The baseline is to see how limited knowledge performs compared to knowing everything. The baseline will be ran in all three environments, with all five number of humans. The watch distance of the baseline will be referred to as infinity or Inf.

2.7 Data Analysis

Starting with the analysis, the data will be checked on outliers. First, the data will be visualized using boxplots with time on the y-axis, the vision on the x-axis and grouped on the number of humans. This will already show points far from the rest of the data, potential outliers. To get more clarity about them, a Bonferroni Outlier test on the time will be used. The test checks for each point if it is a mean-shift outlier, assigning the Bonferroni p-value to the points. While testing the points, it takes the grouping, the different numbers of humans, of the data into account. If the p-value is significant, the data point significantly shifts the mean and is highly likely to be an outlier. For each potential outlier, it will be investigated if the data point is actually an outlier, using the box plot of the data. It will also be explained what probably caused the outliers. Outliers will be removed from the data, this gives a clean dataset which will lead to a better analysis of the data. The data will be analysed for the effects on the time and on the total number of collisions. The baseline data will only be compared to the experimental data based on times.

2.7.1 Time Analysis

After the data is cleaned, the time data will be visualized using two different kinds of graphs. Each environment will get a separate set of graphs. This gives a clear overview which data belongs to which environment. The first kind of graph is a bar plot. This will show the mean of all the different combinations, including the 95% confidence interval. The graph will have the time on the y-axis and the different levels of humans on the x-axis. The distances will be represented with different colours. The graphs will show the significance of the effect of the vision. The baseline data will be added to the bar plots with white bars, to visually distinguish them. The baseline data is only added to the graphs for visual comparison and not taken into account in any of the statistics. The second kind of graph is a two-way interaction plot of the means of all the combinations. The time will be on the y-axis again and the watch distance on the x-axis. The number of humans will be represented with different plotting symbols and colours. This graph will clearly show the differences of mean between the numbers of humans and between the different distances.

The time data will be statistically tested, to see if the vision of the robot significantly influences the speed of the navigation. Only the experimental data will be tested in these statistics and not the baseline data. First, the assumptions of homogeneity of variances and normality of residuals should be checked on the data. Homogeneity of variance will be checked using the Levene's test. If the results are significant, the assumption is violated. A Shapiro-Wilk's test on the residuals of an ANOVA test will be used to check for the normality of residuals. A significant result means that the residuals are not normally distributed and the assumption is violated. If both of the assumptions are met, a parametric, two-way ANOVA test will be run to see if the watch distance has a significant effect on the time depending on the number of humans. If at least one of the assumptions is not met, then an ANOVA test is not allowed. For that reason, a non-parametric Friedman test is performed with the mean of each combination. The Friedman test will test the differences in time caused by the watch distance, taking the effect of the number of humans into account. A significant Friedman test shows that the vision of the robot has an effect on the speed of the navigation.

2.7.2 Collision Analysis

The data of the collisions will also be visualized. The collision data will be visualized and tested without the baseline data. For each scene, a bar plot with of collision data will be created. This will show the mean number of all collisions for each condition, including the 95% confidence interval. The graph will have the time on the y-axis and the vision on the x-axis. The levels of humans will be represented with different colours. The graphs will show the general trend of difference in the number of collisions over the different conditions.

The statistical tests on the collisions will be the same as for the time. If the data has homogeneity of variance and the residuals are normally distributed, a two-way ANOVA test will be used. However, if the data does not have those properties, the non-parametric Friedman test will be performed.

2.7.3 Baseline Analysis

Just like the experimental data, the baseline data will first be checked for outliers. The Bonferroni Outlier test will be performed using the times of the simulations. The baseline data then will be plotted in a box plot, to check if they actually look like outliers. The outliers will be removed from the baseline data. The baseline data is then ready to be analysed.

The baseline data will not be analysed on its own. It will be compared to the experimental data based on the times. For visual comparison, the baseline is plotted with white bars in the bar plots of the times of the experimental data. This is already explained in section 2.7.1, “Time Analysis”.

For statistical analysis, each watch distance of the experimental data will be compared to the baseline data for every number of humans. If the data is normally distributed, then parametric t-tests will be used to test each comparison. Otherwise, a non-parametric Wilcoxon rank-sum test will be performed. From the time analysis will be concluded if the experimental data is normally distributed. The p-values of all the tests will be stored in a table to examine the results. Because of the high number of comparison on the same data, there is a big likelihood to find a significant difference by chance. This will be corrected using the Bonferroni correction. The Bonferroni correction rejects the null hypothesis if the p-value is smaller than 0.05 divided by the number of comparisons.

3 Results

First, the data will be cleaned in section 3.1, all the outliers will be removed. Then the cleaned data will be visualized in bar plots and two-way interaction plots. The plots will show the difference in time and number of collisions between the different conditions. After that, the effect of the watch distance on the time and collisions will be statistically analysed using a two-way ANOVA or a Friedman test. The time data will be visualized and statistically tested in section 3.2 and the collision data in section 3.3. Finally, the baseline data is compared to the experimental data based on time in section 3.4.

3.1 Removing Outliers

First, the time data is visualized in box plots, to see how the data is distributed. Figure 11 shows the box plot of the Office data. This clearly shows one point which is a lot higher as the rest of the data. Furthermore, there are some points outside of the inter quartile range, but they are not extreme enough to worry about them. Running the Bonferroni Outlier test gives back one highly significant data point, p-value = $3.56e^{-147}$. The simulation took 67.4 seconds with a watch distance of 25 and 25 humans. This corresponds to the very high data point in figure 11, marked with a black circle. In the simulation, the robot probably got stuck for a while in one of the offices. This caused the robot to take about three times as much time to reach the goal. For the research, it was not taken into account that the robot could get stuck. To get free, the robot should turn around and go back, however this is not implemented. This caused the robot to only get out of the office by accident, which took a lot of time and does not represent a correct navigation. For this reason, the simulation can be removed. After removing one data point from the Office data, the data is clean and ready to be analysed in the next section.

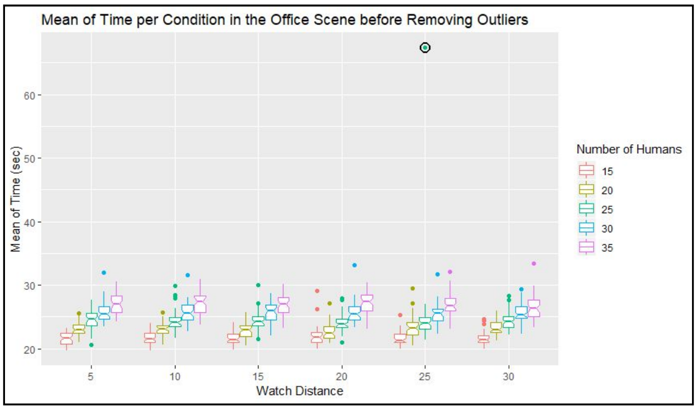

Figure 11. *Box plot of the mean of time for each condition in the Office. The outliers are marked with a black circle.*

The Street data is shown in figure 12. The graph does not show any point very far from the data. There are a few points outside of the inter quartile range. However, the Bonferroni Outlier test does not give a significant result for any of the data points. The most extreme value, with a p-value of 0.364, is a simulation with a watch distance of 5 and 110 humans which took 25.5 seconds. This means that

Figure 12. *Box plot of the mean of time for each condition in the Street.*

there are no outliers and the data is ready to be analysed.

Figure 13 shows the box plot of the Obstacles data. The box plot has one data point which is has an extreme value compared to the rest of the data and some which are higher. The Bonferroni Outlier test gives back ten potential outliers. The first one represents the extremely higher point in de box plot, with a significance value of $7.05e^{-165}$. The simulation took 176.0 seconds, which about five times as slow as the mean of time under that condition. The other nine points have some variation in p-value from $2.78e^{-21}$ to $7.74e^{-3}$. The times of the simulation range from 92.4 seconds to 59.9 seconds, which is about two to three times as slow as average. The data points are marked with a black circle in figure 13. All potential outliers also show relatively high collision counts. The robot probably got stuck, because of a malfunction of the simulation. Figure 14 shows the NavMesh of the robot with a watch distance of 30 in the Obstacles scene. On the left side of the left middle obstacle there is a passage for the robot, made clear by the red circle. However, if the robot chooses that path, the NavMesh sometimes does not bake the small passage anymore. Which causes the robot to get stuck. After a while the robot gets bumped into from behind by some deactivated human agents and the robot slowly turns and finds a path the other way around the obstacles. This causes the long time the simulation takes and the relatively high collision count. This means that the data point are caused by a malfunction of the simulation and they are outliers in the data, for that reason they should be removed. The data is now clean and ready to be analysed.

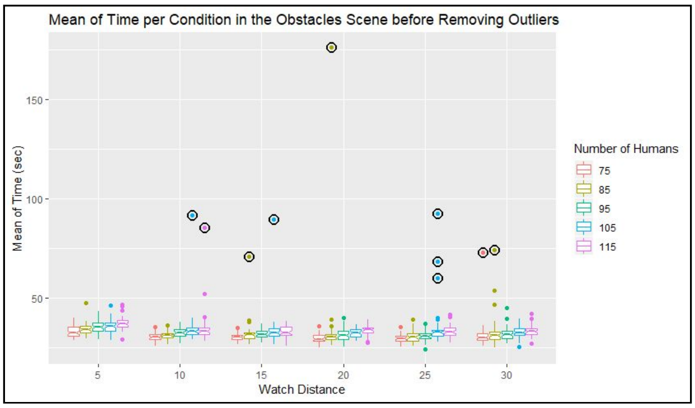

Figure 13. Box plot of the mean of time for each condition in the Obstacles scene. The outliers are marked with a black circle.

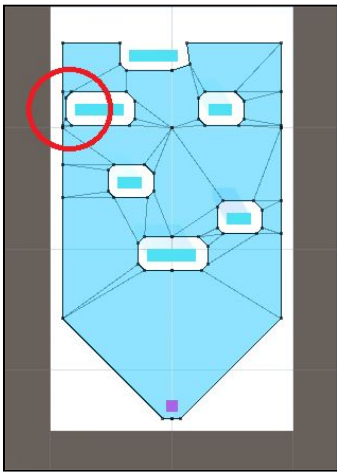

Figure 14. The NavMesh of the robot with a watch distance of 30 in the Obstacles scene. Marking the passage in the NavMesh which sometimes is not baked with a red circle.

3.2 Time Results

Figure 15 shows the bar plot of the time in the Office scene. The mean of times stays mainly the same with a higher watch distance. This can also be seen from the relatively straight lines in figure 16. It looks like that the watch distance does not influence the speed of navigation. This is also confirmed by an insignificant Friedman test, p-value = 0.57. The assumptions for normality of residuals(p-value = $3.13e^{-15}$) and homogeneity of variance (p-value = $1.74e^{-3}$) are both violated, thus a two-way ANOVA test is not allowed.

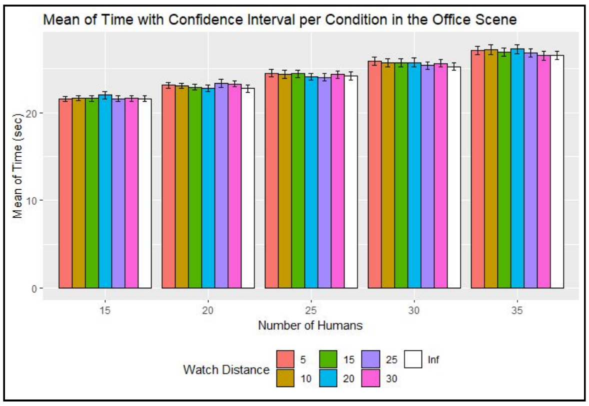

Figure 15. *Bar plot of the mean of time with confidence intervals for each condition in the Office. The coloured bars show the experimental watch distances. The white bars are from the baseline data.*

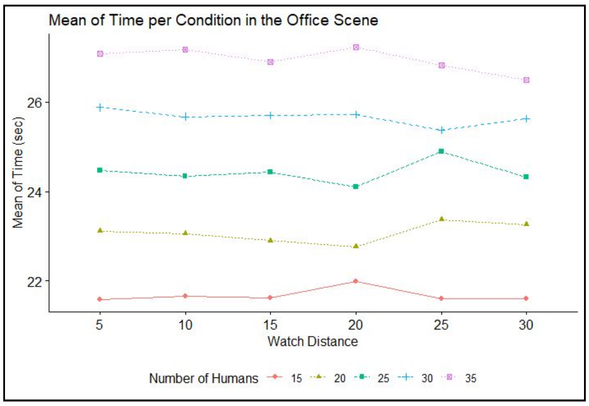

Figure 16. *Two-way interaction plot of the mean of time per condition in the Office scene.*

The time data of the Street is visualized in figure 17 and figure 18. Both graphs show a decreasing mean of time while the watch distance increases. The decrease is steep in the beginning and flattens around a watch distance of 20. The differences look significant from the 95% confidence interval in figure 17. The Friedman test confirms this significant difference with a p-value of $1.89e^{-4}$. A two-way ANOVA was not allowed, because the data has no homogeneity of variance, p-value = 0.0322.

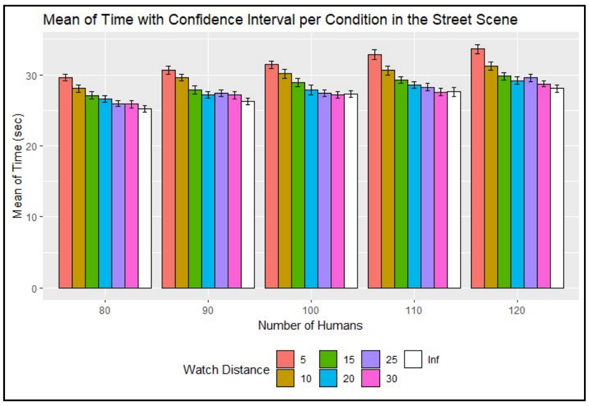

Figure 17. *Bar plot of the mean of time with confidence intervals for each condition in the Street scene. The coloured bars show the experimental watch distances. The white bars are from the baseline data.*

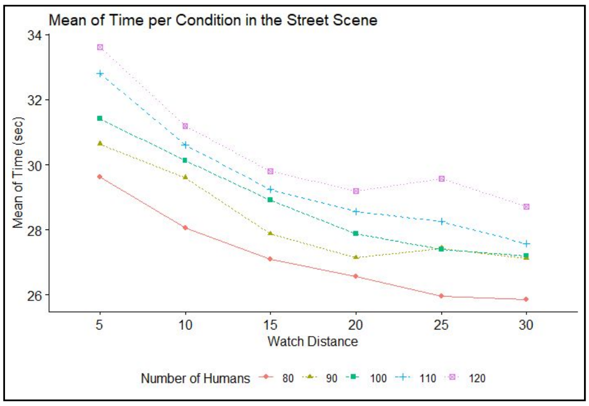

Figure 18. *Two-way interaction plot of the mean of time per condition in the Street.*

Figure 19 and figure 20 displays the time data of the Obstacles scene. They show that the simulations with a watch distance of 5 took about 3 seconds longer on average for each number of humans compared to the other distances. The times with watch distances between 10 and 30 fluctuate a little, but not significantly. The Friedman test shows that the difference between a watch distance of 5 and the other distances has a significant effect on the time, p-value = $2.52e^{-3}$. Again, a two-way ANOVA was not allowed, the residuals are not normally distributed, p-value < $2.2e^{-16}$ and the homogeneity of variance assumption is only slightly met, p-value = 0.0516.

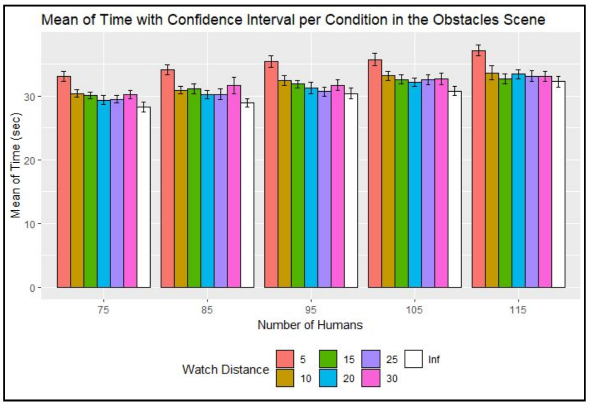

Figure 19. *Bar plot of the mean of time with confidence intervals for each condition in the Street scene. The coloured bars show the experimental watch distances. The white bars are from the baseline data.*

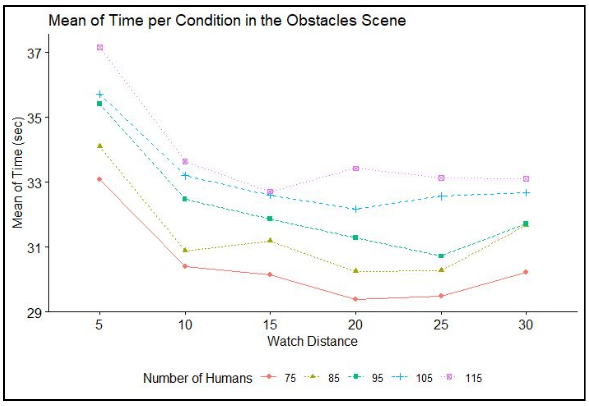

Figure 20. *Two-way interaction plot of the mean of time per condition in the Street.*

3.3 Collision Results

The mean number of all collisions in the Office is visualized in figure 21. The bars for each specific number of humans all appear to have a similar height, independent of the vision. The Friedman test also shows that there is no difference between the number of collisions for each watch distance, the p-value is 0.303. A two-way ANOVA test was not allowed, because the data is not normally distributed nor has it homogeneity of variance, both have a p-value smaller than $2.2e^{-16}$.

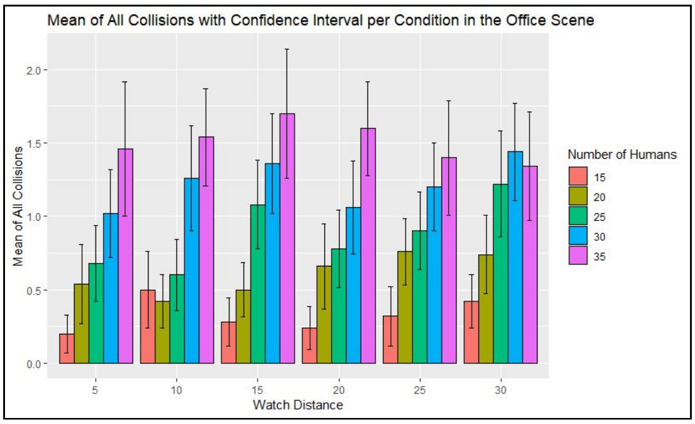

Figure 21. *Bar plot of the mean number of all collisions with confidence intervals for each condition in the Office.*

Figure 22 displays the collision data of the Street. The bars look very chaotic, there is no real structure between the bars. The Friedman test shows, with an insignificant p-value of 0.270, that the watch distance has no uniform effect on the number of collisions. The assumption of homogeneity of variance and normality are violated, p-value = $3.27e^{-3}$ and p-value < $2.2e^{-16}$ respectively, which causes a two-way ANOVA to be forbidden.

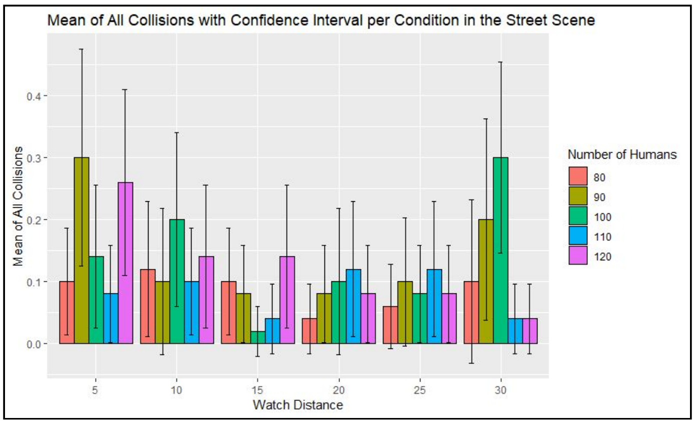

Figure 22. *Bar plot of the mean number of all collisions with confidence intervals for each condition in the Street.*

Figure 23 shows the mean number of all collisions per simulation in the Obstacles scene. Comparing all bars for a specific number of humans shows a slight upwards slope, however there are quite big confidence intervals. The collisions data is not normally distributed (p-value < $2.2e^{-16}$) and has no homogeneity of variance of residuals (p-value = $1.23e^{-5}$), thus a two-way ANOVA cannot be used to see if the vision has an effect. However, the Friedman test comes back significant (p-value = $6.43e^{-3}$), which shows that the watch distance has an effect on the total number of collisions in robot navigation through a crowd. Figure 24 indeed shows a slight upwards slope in the mean number of all collisions per simulation.

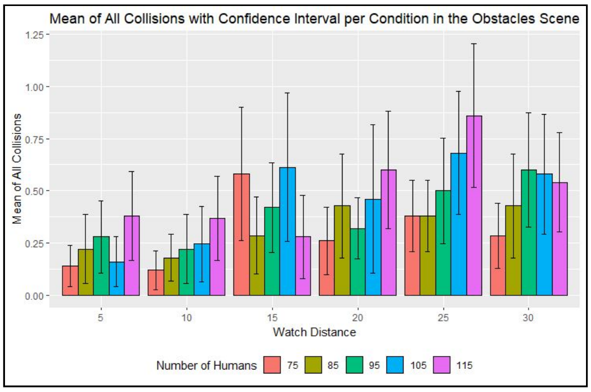

Figure 23. Bar plot of the mean number of all collisions with confidence intervals for each condition in the Obstacles scene.

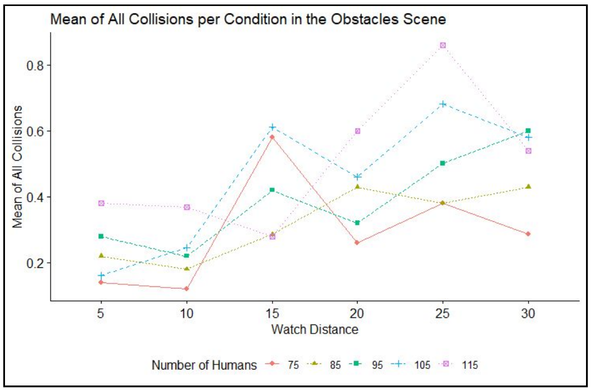

Figure 24. Two-way interaction plot of the mean number of all collisions per condition in the Obstacles scene.

3.4 Baseline Results

The Bonferroni test gives back four mean-shift outliers which are highly significant. All four point are from the Obstacles scene. The times of the simulations vary from 63 to 70, which is at least twice as high as the average times. Figure 25 shows a box plot of the baseline data with the outliers marked with a black circle. The four points are removed from the baseline data.

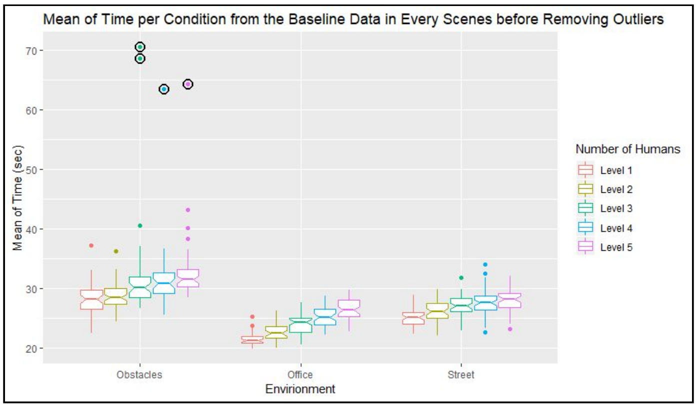

Figure 25. Box plot of the mean of time for each condition from the baseline data in all the scenes. The outliers are marked with a black circle.

Figure 15, 17 and 19 show the mean of time from the baseline data as white bars next to the experimental data for each scene. The baseline data looks to be mostly in line with the experimental data in all scenes. The baseline is often a bit lower as experimental data and sometimes slightly higher.

		Watch Distance					
		5	10	15	20	25	30
Number of Humans	15	0.793	0.471	0.625	0.146	0.869	0.741
	20	0.115	0.168	0.432	0.929	0.0613	0.0320
	25	0.264	0.912	0.586	0.521	0.590	0.847
	30	0.123	0.261	0.162	0.267	0.907	0.287
	35	0.146	0.135	0.381	0.065	0.478	0.632

Table 2. The p-values of the Wilcoxon rank-sum test comparing the times of each watch distance of the Office data to the times of the baseline data for each number of humans. Significant values are marked gray.

Section 3.2 shows that the experimental data violates the assumption of normality, for that reason the non-parametric Wilcoxon rank-sum test is used for the comparisons. Each piece of baseline data is compared to six watch distances, thus the Bonferroni correction says that a p-value is significant if it is smaller than 0.05/6 = 0.0083. Table 2, 3 and 4 show the p-values for all comparisons of the baseline data to the Office data, Street data and Obstacles data respectively. The significant values are marked gray. None of the comparisons with the Office data are significant. Almost all comparisons with the low watch distances are highly significant for the Street and the Obstacles data. Some of the higher watch distances show a significant results and some do not. All comparisons between the baseline data and a watch distance of 30 in the Street are insignificant.

	Watch Distance					
	5	10	15	20	25	30
Number of Humans 80	2.93E-16	4.18E-11	8.86E-07	5.51E-05	0.00718	0.0873
90	3.67E-15	2.11E-13	4.04E-05	0.0198	0.00179	0.0264
100	4.69E-15	5.61E-10	3.22E-05	0.202	0.672	0.934
110	3.95E-14	2.58E-09	6.38E-05	0.00771	0.131	0.951
120	4.13E-16	8.75E-11	1.11E-05	0.00422	2.20E-04	0.0709

Table 3. *The p-values of the Wilcoxon rank-sum test comparing the times of each watch distance of the Street data to the times of the baseline data for each number of humans. Significant values are marked gray.*

	Watch Distance					
	5	10	15	20	25	30
Number of Humans 75	1.76E-12	7.67E-06	4.48E-05	0.0161	0.00461	1.15E-04
85	6.70E-14	1.80E-05	2.85E-05	0.00411	0.0202	7.68E-05
95	7.18E-11	1.17E-04	0.00104	0.0963	0.290	0.0107
105	2.69E-11	3.32E-05	0.00125	0.00648	0.00159	0.00198
115	2.22E-11	0.0132	0.0903	0.00552	0.0552	0.0340

Table 4. *The p-values of the Wilcoxon rank-sum test comparing the times of each watch distance of the Obstacles data to the times of the baseline data for each number of humans. Significant values are marked gray.*

4 Discussion

Looking at the mean of time for each condition, time is clearly influenced by the watch distance in the Street and the Obstacles scene. This is visible in figures 17, 18, 19 and 20 and also from the Friedman test. If the watch distances gets bigger, the average times of the simulations get lower. In the Street, the effect appears to stabilize around a watch distance of 15 or 20. In the Obstacles scene, the effect is only there between a watch distance of 5 and 10. The effect of the watch distance is not visible in the Office, clearly shown in figure 15. The means of time for all watch distances for a specific number of humans are approximately the same. The differences in the effect show that the effect is dependent on the environment. In a narrow environment, such as the Office, vision has no effect. The space is so small, that the robot does not really have a choice for the direction it goes. If two agents approach the robot in the Office, there is no space to avoid them, even if you already saw them approaching earlier. In a broader environment, the robot can go to another direction if he sees them approaching him. With a longer vision, the robot can step out of the way earlier, which spares him time. Time he cannot spare in narrow environments, which causes the watch distance to have no effect in the Office. The difference between the Street and the Obstacles scene are the static obstacles present in the Obstacles scene, but not in the Street. These obstacles also explain the difference in the effect. With a watch distance of 5, the obstacles are only visible when the robot is very close to them. At that point, the robot has to choose which direction to go and doubts a lot, which takes some time. With a watch distance of 10 and higher, the robot sees the obstacles sooner and hence can take a decision earlier, which spares time. When the robot reaches the side of the obstacle, he often waits for the humans coming from the other side. This waiting happens for all watch distances and it always takes approximately the same time, because the same number of humans come by. This causes that there is no difference between the higher watch distances. In the Street, there are no obstacles. The robot only has to respond to the humans. He has to decide which side of the human is the best to pass. The side where the least humans are, is the best. With more vision, the robot knows for a bigger surface where the least humans are. If there are, for example, five humans to the left on 3 to the right in the first five meters and 1 to the left and 10 to the right in the second five meters. Then the robot would go left with a distance of 5, but right with a distance of 10. The right side actually has the least number of humans and will be faster. The effect is very clearly visible in the results of the Street. The Street and the Obstacles scene demonstrate that the watch distance has an effect on the time that the navigation takes. The effect is most visible with short distances. The Office shows no effect of watch distance, from which we can conclude that a broad environment is necessary to obtain the effect.

The graphs about the time data all show that more humans increases the average time a simulation takes. The robot has to wait and slow down more often, because more humans are approaching him. However, the effect of the watch distance on the time is not influenced by the number of humans. Figures 16, 18 and 20 show that the different means of time for each number of humans run approximately in parallel. Which means that the differences in time are equivalent for all watch distances. The effect of the watch distance is the same for all number of humans.

Comparing the times of the experimental data with the baseline data mostly corresponds to what is found from the experimental data. Figure 15 and table 2 show that the baseline data is perfectly in line with the results from the Office. The bar plot in figure 15 displays that the mean of time of the baseline data is approximately equal to the other data. The Wilcoxon rank-sum test also provides that there is no significant difference between any of the bars. This is in line with the finding that the watch distance does not influence the navigation in the Office. Full world knowledge is as good as partial knowledge in a narrow environment. The baseline data in the Street appears to be in line with the experimental data in figure 17. The average times of the baseline data is approximately equal to the average times of the simulations with the larger watch distances. Almost all comparisons between the baseline and the three largest watch distances are insignificant can be seen in table 3, showing that the performance does not give an significant difference. All comparisons between the baseline and the lower watch distances are significant, demonstrating that the lower watch distances perform worse as knowing everything. Comparing the Street data to the best possible case, knowing everything, shows that the robot does not need to know everything to navigate optimally. The findings in the Obstacles scene are less strongly confirmed by the baseline data. Figure 19 shows that the means of time of the baseline are all lower as in the experimental data. In table 4 can be seen that not all differences are significant, but most of them are significant. From the experimental data is concluded that the watch distances from 10 to 30 are approximately equal and a watch distance of 5 is on average 3 seconds slower. However, from the baseline data it looks like full world knowledge is faster than partial knowledge. Knowing the position of all obstacles gives a benefit over knowing only some of them. The baseline comparisons leads to some doubts about how the watch distance influences the time of the navigation of a robot through an environment with many obstacles. The influences of the watch distance on the time a robot needs to navigate through a crowd in open spaces and narrow environments is confirmed by the baseline comparison.

The Friedman test concludes that the watch distance has no effect on the number of collisions in two of the three environments. Only in the Obstacles scene there is an effect according to the Friedman test. However, figure 23 shows that the confidence intervals of the data are really big. Which means that the results are very variable. Figure 24 shows that the effect is quite small. The difference in collisions is about three collisions more in ten simulations. The effect is so small that it can be neglected. Which concludes that the number of collisions is not influenced by the watch distance. Seeing the graphs, there is no relation between the number of collisions and the watch distances. Figure 21, from the Office, and figure 23, from the Obstacles scene, show that the number of collisions stays quite equal for every watch distance. But, in figure 22, from the Street, the number of collisions appears to be unrelated to the watch distance. This difference can be explained from the scenes. In the Office and the Obstacles scene, the robot has less freedom to navigate, because of the obstacles and walls. In the Office, the corridor is so small that only three agents fitt next to each other. If two agents come towards the robot, the chance to bump into them is quite high, because there is not much space to avoid them. A longer watch distance does not make the path broader, thus it does not lower the chance of bumping into them. In the Obstacles scene, there are only a small number of paths which the robot can take around the obstacles. The agents also have to go around the obstacles. If suddenly an agent comes around an obstacles, the robot is very likely to bump into him. An agent which suddenly comes around an obstacle is not influenced by a bigger watch distance. Both in the Office and the Obstacles scene, the watch distance does not influence the number of collisions and there are certain places where the robot can bump into something with a considerably high chance, which causes the number of collisions to be equals for all distances. However, in the Street, there are no places with a relatively high chance to bump into an agent, because there are no obstacles or walls. The collisions only happen because of unexpected behaviour of the agents, which is random and leads to no relations between the collisions and the vision. All the environments reveal that the number of collisions is independent of the vision. This means that based on the number of collisions, the navigation of the robot is not improved by the amount of knowledge.

The time and collision analyses show that the time of the navigation is influenced by the watch distance and that the number of collisions is not influenced. This means that overall, the navigation of a robot through a crowd is improved by the watch distance.

5 Conclusion

From the experiment can be concluded that navigation of a robot through a crowd is improved with more knowledge. The improvement can only be found in the field of time and not for collisions. The number of collisions appeared to be independent of the knowledge.

The navigation is faster with more knowledge. However, the effect on the time is only found in broad environments. With narrow spaces, where only a few agents fit next to each other, more knowledge does not improve navigation, because there is not much space to avoid the agents. With broad environments, more knowledge does improve the navigation. However, the additional benefit of more knowledge is quite minimal. Adding more knowledge is only beneficial in the beginning. From the Street and the Obstacles scene, can be concluded that it is best to have a watch distance of 20 or 15. After that distance, the mean of time does not significantly improve anymore with more knowledge. When the watch distances are higher, the navigation takes approximately the same time as with full world knowledge of the baseline.

It is also found that the effect of the knowledge is not influenced by the size of the crowd. A bigger crowd does increase the time the navigation takes, but this time difference is equivalent for all amounts of knowledge.

The experiment shows that the amount of knowledge needed for a robot to optimally navigate through a crowd depends on the environment. In narrow environments, a simple sensor will already be enough. However, in broader environments, a very simple, cheap sensor will probably not suffice for proper navigation through a crowd. A robot needs to know a little more as only the closest agents and obstacles. However, a super complex sensor which perceives the whole environment is also not necessary. Further research should find the right middle way between evaluating almost nothing and everything.

5.1 Further Research

There is quite some room for further work that still can be done. The simulation works, but some flaws were found during the experiment. For example, the robot could get stuck. Further research can be performed with a simulation where the robot can turn around when he gets stuck. This would be a more realistic situation and prevents outliers which now appeared in the data. Another aspect is that the experiment and simulation are quite simplified compared to the real world. The experiment was a starting point for research to find out if knowledge influences navigation. This influence is found, but the details are not examined yet. The knowledge of the robot in the experiment cannot be replicable in real life. Further research can be done with more advanced simulations and even real life experiments, where the knowledge of the robot is obtained from sensors on the robot. It is also possible to give the robot 360° view, such that he sees behind himself, to investigate how that influences the navigation. It should allow the robot to turn around and not get stuck. Lastly, further research can be done with more advanced evaluation of results. This will give a more accurate overview between the different navigations. It can, for example, be taken into account how often the robot stops or when it accelerates. This will show if the robot waits-and-sees or just goes for it. Further research in different environments would also be useful. This can examine what exactly in an environment influences the robot's navigation. The type of sensors and amount of knowledge a robot requires to navigate very in different situation. Further research can make clear what a robot requires in every situation to navigation optimally.

Acknowledgment

I would like to thank my supervisor Serge Thill for all the insight into my thesis and all the help during the process. I also would like to thank Lars Doornbos for reading my thesis over and over again, because I kept changing everything. And also Rianne-Margot Koelewijn for reading the final version and commenting on almost every sentence. Lastly, I would like to thank the Brackeys YouTube channel for all the good videos on how to work with Unity.

References

- Barnouti, N. H., Al-Dabbagh, S. S. M., & Naser, M. A. S. (2016). Pathfinding in strategy games and maze solving using A search algorithm. *Journal of Computer and Communications, 4*(11), 15.
- Bera, A., Randhavane, T., Prinja, R., Kapsaskis, K., Wang, A., Gray, K., & Manocha, D. (2019). The Emotionally Intelligent Robot: Improving Social Navigation in Crowded Environments. *arXiv preprint arXiv:1903.03217*.
- Cabibihan, J. J., Javed, H., Ang, M., & Aljunied, S. M. (2013). Why robots? A survey on the roles and benefits of social robots in the therapy of children with autism. *International journal of social robotics, 5*(4), 593-618.
- Chiu, Y. P., & Shiau, Y. C. (2016). Study on the application of unity software in emergency evacuation simulation for elder. *Artificial Life and Robotics, 21(2)*, 232-238.
- Codd-Downey, R., Forooshani, P. M., Speers, A., Wang, H., & Jenkin, M. (2014). From ROS to unity: Leveraging robot and virtual environment middleware for immersive teleoperation. In *2014 IEEE International Conference on Information and Automation (ICIA)* (pp. 932-936). IEEE.
- Gelbal, Ş. Y., Altuğ, E., & Keçeci, E. F. (2016). Design and HIL setup of an autonomous vehicle for crowded environments. In *2016 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)* (pp. 1370-1375). IEEE.
- Hornung, A., Wurm, K. M., & Bennewitz, M. (2010). Humanoid robot localization in complex indoor environments. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 1690-1695). IEEE.
- Joosse, M., & Evers, V. (2017). A guide robot at the airport: First impressions. In *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (pp. 149-150). ACM.
- Koceski, S., & Koceska, N. (2016). Evaluation of an assistive telepresence robot for elderly healthcare. *Journal of medical systems, 40*(5), 121.
- Li, M., Jiang, R., Ge, S. S., & Lee, T. H. (2018). Role playing learning for socially concomitant mobile robot navigation. *CAAI Transactions on Intelligence Technology, 3*(1), 49-58.
- Manti, M., Pratesi, A., Falotico, E., Cianchetti, M., & Laschi, C. (2016). Soft assistive robot for personal care of elderly people. In *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)* (pp. 833-838). IEEE.
- Park, C., Ondřej, J., Gilbert, M., Freeman, K., & O'Sullivan, C. (2016, October). HI Robot: Human intention-aware robot planning for safe and efficient navigation in crowds. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 3320-3326). IEEE.
- Savkin, A. V., & Wang, C. (2014). Seeking a path through the crowd: Robot navigation in unknown dynamic environments with moving obstacles based on an integrated environment representation. *Robotics and Autonomous Systems, 62*(10), 1568-1580.
- Shiomi, M., Iio, T., Kamei, K., Sharma, C., & Hagita, N. (2015). Effectiveness of social behaviors for autonomous wheelchair robot to support elderly people in Japan. *PloS one, 10*(5), e0128031.
- Trautman, P., & Krause, A. (2010). Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 797-803). IEEE.

- Trautman, P., Ma, J., Murray, R. M., & Krause, A. (2015). Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation. *The International Journal of Robotics Research, 34(3)*, 335-356.
- Unity Technologies. (2019). Unity. Retrieved from https://unity3d.com/unity.
- Wood, J., Margenet, M. C., Kenneally, P., Schaub, H., & Piggott, S. (2018). Flexible Basilisk Astrodynamics Visualization Software Using the Unity Rendering Engine. In *AAS Guidance and Control Conference, Breckenridge, CO*.
- Zikky, M. (2016). Review of A*(A star) navigation mesh pathfinding as the alternative of artificial intelligent for ghosts agent on the Pacman game. *EMITTER International Journal of Engineering Technology, 4*(1), 141-149.