User-centred design of a robot collision avoidance procedure for robots in supermarket environments

Project Robots Everywhere

0LAUKO

Tristan de Waard

0996117

Contents

Introduction & problem statement

Robot navigation and collision avoidance in crowded and dynamic environments is a challenging problem, not only from a technical point of view, but also when looking at how robots should behave in the proximity of (large numbers of) people. For various (social) robot applications, finding a method that is desirable for humans while maintaining movement efficiency is of great importance. Robot applications in supermarkets is rising in popularity, examples being Best Buy's Chloe robot[1] and Simbe Robotics' Tally robot[2] so it would be useful to develop methods that are specifically designed for a supermarket environment keeping customer and staff requirements in mind.

This research will focus on finding a solution for robot collision avoidance in a supermarket environment. A supermarket environment has aspects that make it unique from other crowded environments. To make this more concise, a description of this environment is given with advantages and difficulties for designing a robot collision avoidance. Furthermore, it will also become clear that users (staff and customers) will have certain requirements that relate to human robot interactions (HRI). Keeping both the environment and user requirements in mind, two state-of-the-art collision avoidance procedures will be assessed on application in a supermarket environment and possible additions to enhance them for this application will be investigated. The most fitting procedure will be chosen after which a quantitative way of testing user requirements in a simulation is given. Several simulations will be done to test its working potential and adaptations and extensions to the chosen approach are presented. A motivation is given on why the adapted approach is a superior approach for robot collision avoidance in a supermarket through simulation and further extensions and limitations to this approach are described for future research.

Task environment description of a supermarket

We will look at advantages and difficulties for robot collision avoidance in supermarkets. Through this analysis several aspects that need investigation will come to light.

Advantages for collision avoidance

1. It is assumed that there are several (security) cameras already mounted on the ceiling and that the robot already possesses an omnidirectional camera. By giving the robot access to ceiling mounted cameras, these can be used for collision avoidance as extra sensory input on top of the camera already present on the robot itself. This gives the robot a top down view of the area he is in, filling in blank spots in the robot's local sensing. This poses several questions; for one, security cameras usually make use of fish-eye cameras giving a distorted view of the environment, meaning that these images might need to be processed or are not usable at all. Then also, how many extra ceiling cameras would be necessary and how much would that cost? Takaaki Sato et al.[3] have proved that fish eye cameras can be used to make a (2D) bird's eye view of an environment to eliminate blind spots in a robot's local sensing. However, it needs to be investigated whether it is still desirable for a supermarket enterprise to invest in more cameras, when the cheaper option of only using local robot cameras might suffice.

2. Supermarket aisles have a static layout, with each aisle having distinct retail products ordered in a known layout. This semantic information stored in retail products can be used for robot localisation and navigation from point A to B. A detailed description of navigation using semantic techniques is given by Cosgun and Christensen[4]. Since this is not applicable for collision avoidance specifically but more important for navigation in general, it will not be discussed here.

However, this is a promising starting point for research in navigation algorithms for supermarkets that might be combined with collision avoidance procedures.

Difficulties for collision avoidance

1. Customers and staff members will be walking around supermarkets, either in groups or alone, maybe carrying a shopping cart. It might be useful if robot velocities are constrained in situations where people can unexpectedly come around corners potentially leading to collisions, which can happen at the ends of supermarket aisles. Then also, all people in the environment need to be avoided in a way that is predictable and perceived as safe by them. The robot should therefore act differently when humans, instead of (static) inanimate objects are to be avoided. To find out how a robot should act differently among humans, an investigation on proxemics for HRI needs to be done.

2. There are also peak times in the number of customers walking around (e.g. on Saturdays). Collision avoidance procedures on their own might then lead to the robot having no way to avoid masses of people or lead to computationally expensive situations where the robot loses reactiveness. A solution for robot collision avoidance in masses might be found when looking at how humans tend to cooperate to avoid each other in these situations. Conventional collision avoidance procedures might tell the robot to not come closer than X metres to humans to respect their personal space. However, humans sometimes tend to avoid each other crossing this line in crowded situations still maintaining to be polite and/or comfortable with each other. This is something that needs to be considered in collision avoidance procedures. Moreover, procedures might need to be adapted so that crowded areas are detected and then treated in a more computationally light way. In this situation it might also be necessary to add visible or audible cues that alert surrounding customers in a comfortable way to make sure the robot is noticed by surrounding humans to facilitate movement in crowded spaces. It should be investigated what kind of cues are desirable in these situations, how (computationally) inefficient some procedures become when large groups of people are in the robot's vicinity and how these inefficiencies can be overcome.

3. Shopping carts are present either in a parked or moving state, which will have to be avoided. For a parked shopping cart case, there should be some prediction about probability that it will move and in what direction. This probability should be depended on whether a human is close to that cart. These probabilities might be incorporated in a cost function for shopping carts specifically. It needs to be investigated if this motion prediction is worth the extra computational cost to the algorithm, possibly by doing simulations in combination with real-life experiments.

4. Miscellaneous items such as boxes, pallets or retail products fallen from shelves might be present as obstacles. Ceiling mounted cameras should be able to detect these obstacles. Since these objects are static, no movement prediction is necessary. The location of static objects can be sent to the robot directly or can be sensed by the robot itself and path planning can be adapted accordingly. It might also be beneficial if robot velocities are constrained when near shelves, so that slightly protruding misaligned products will not be hit that hard and potentially fall out of shelves, creating unwanted obstacles.

Overview of the environment

Because of the static layout of aisles in supermarkets, the environment of a supermarket is for this research simplified to one aisle that the robot needs to navigate through. This is done to fully leave out the navigation aspect for robots in the environment and focus solely on collision avoidance. During its path it will encounter static objects, moving objects as well as several humans standing around, walking and interacting the robot all needs to avoid in a reactive manner. Furthermore, it is assumed that a top-down view of the aisle is accessible to the robot by using images of several (fish-eye security) cameras mounted on the ceiling. This environment is schematically illustrated in the following figure.

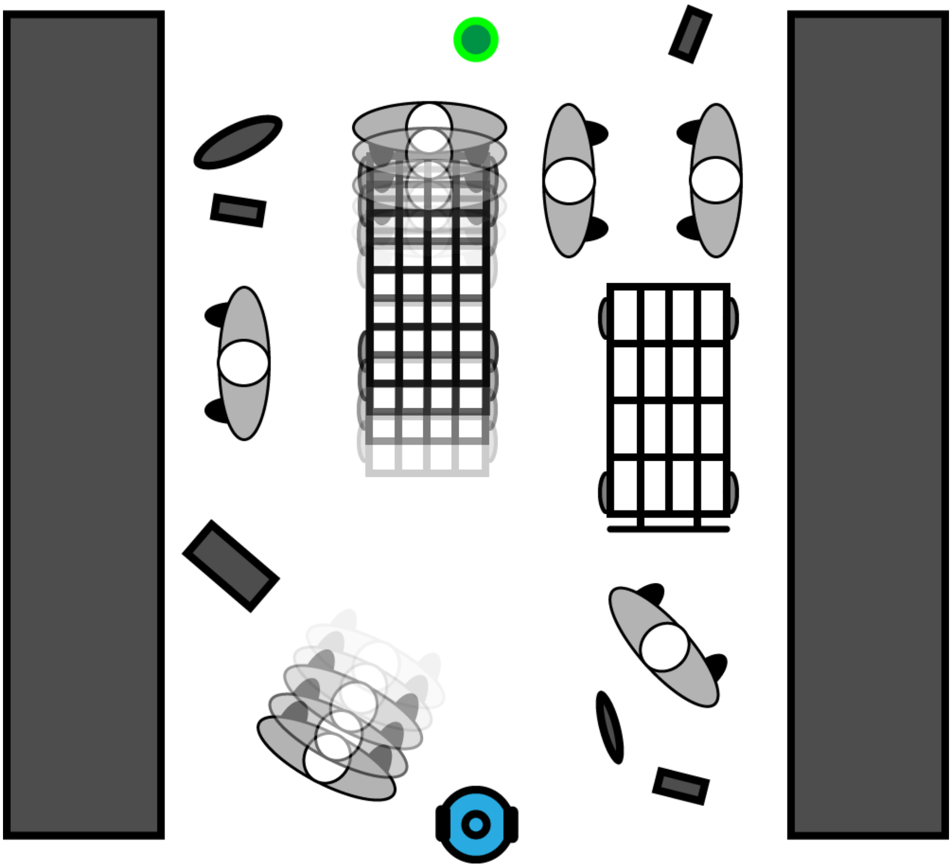


Figure 1 Schematic overview of the collision avoidance environment, the robot is represented in blue while the goal is represented in green.

Identifying user requirements

Proxemics and HRI

When avoiding or moving close to humans, it is important that humans do not feel any discomfort, harm or surprise. To make these and related terms more concise the definitions of Thibault, K et al.[5] will be used:

Comfort is the absence of annoyance and stress for humans in interaction with robots.

It should be noted that comfort is different than safety, in that a robot can move about safely but the surrounding people may feel unsafe. The opposite is also possible, when the human perceives a robot moving about safely it can still end up in a collision.

Naturalness is the similarity between robots and humans in low-level behaviour patterns.

Naturalness thus strives to a physical imitation of humans as much as possible. Examples are movement speeds and robot shapes that resemble humans.

Sociability is the adherence to explicit high-level cultural conventions.

Sociability is seen as constraints posed by society. Examples are the rule to walk on the righthand side and politely asking someone to move out of the way.

To find a desirable way in which robots avoid and move alongside customers and staff members, several robot user requirements will be looked at now. They are based on surveys presented by the literature summary of Thibault, K et al. and from studies in the field of proxemics.

1. Robots should never come too close to humans, even during object avoidance routines. It could frighten humans, possibly leading to sudden actions and human injury.

E. Hall[6] found designations for interpersonal distances for several human to human interactions:

Designation	Specification	Reserved for ...
Intimate distance	0 - 45cm	Embracing, touching, whispering
Personal distance	45 - 120cm	Friends
Social distance	1.2 - 3.6m	Acquaintances and strangers
Public distance	> 3.6m	Public speaking

Figure 2 Proxemics table from E. Hall

This table can be used to find a proper distance for robots during an avoidance or general movement that respects the personal zones of people. Generally, to make a person feel comfortable (safe) the robot should try to avoid the intimate and personal space of people, so a distance of more than 120 cm would be preferred during avoidance. Although this table does not incorporate the fact that a robot instead of a human is entering these personal spaces, current research still suggests that using these distances as a basis for robot navigation and collision avoidance is still a viable option. As stated earlier, however, this distance of 120 cm should not be implemented as a no-go zone for robots. If the robot is noticed by surrounding humans and has adapted its speed accordingly to some desirable approaching speed, it should be able to enter a person's intimate distance temporarily for collision avoidance. A personal space model that nicely incorporates this aspect is given by Barnaud, M.-L et al. and will be discussed in the next section.

2. Robots should not block a human's path, which may cause frustration.

This requirement is thus based on comfort and naturalness. It is rather straightforward, however, Thibault, K et al. describe that when humans actively try to avoid robots as well (when the robot's movement is perceived as safe and predictable) this is not necessarily a problem anymore. This requires that the robot is easily noticed by surrounding people.

3. In a case of a densely crowded area, the robot should provide humans with a visible or audible cue, possibly through language, to make collision avoidance possible or easier.

In this situation a robot should make clear to surrounding people that it wants to move in a certain direction and that some people might need to adapt their (walking) behaviour accordingly by, for example, making way for the robot. This cue should be as effective as possible in crowded environments, while also making sure that people do not lose comfort. The robot will thus attempt to persuade people to change their actions, which some people may dislike or even try to resist against. Ghazali, A. S. Et al.[7] describe that this phenomenon is described as psychological reactance, which can result from people perceiving these persuasive attempts as threats towards their freedom in decision making. This research concludes that using highly controlling language can lead to successful persuasions and that the cues presented by the persuading robot such as facial expressions do not have an impact. High controlling language uses explicit verbs, an example could be "You must..." etc. For this purpose of collision avoidance in a supermarket, it is thought that a verbal output using low controlling language, thus for example, "Could you please make way for me?" or something similar, is desirable to be used. According to Lohse, M. Et al.[8] People tend to have a positive attitude towards robots that communicate its intention through audio, compared to robots that do not communicate its intentions. Therefore, this is the preferred way of interacting with customers in crowded areas. However, it will still be necessary to do further research on which exact sentences using low controlling language are the most effective and fitting for a supermarket, minimising social reactance.

4. Robots should not move/approach too fast, which leads to discomfort for surrounding people.

Butler and Agah[9] found that approaching with 1 [m/s] turned out uncomfortable, while 0.5 [m/s] was acceptable. During avoidance the situation is slightly different, but the same velocities could be used. An important aspect of robot movement is the degree in which it is predictable, understandable or readable for humans (natural). According to Hayashi[10] and Satake[11] a speed that adapts to or resembles surrounding humans would be desirable for general movement. Humans tend to have a preferred walking speed of 1.4 [m/s][12], so it is thought that the robot's speed should always be lower than that, while a velocity of 0.5 [m/s] is admissible during the event that the robot (temporarily) enters someone's intimate space for collision avoidance.

5. Avoid erratic motions during movement, especially when close to humans.

This refers to the aspect of smoothness, which means that the geometry of the taken path and the velocity profile should be smooth. This would improve the naturalness and predictability of robots. To ensure a smooth velocity profile that resembles humans, it is necessary to impose constraints on the acceleration of the robot. Human pedestrian acceleration is found to have a maximum value of 1.44 [m/s^2] with an average of 0.68 [m/s^2][13]. The average acceleration of a human should then be set as a maximum for robot acceleration.

6. Robots should not make noises that cause distraction when coming close to humans, to increase comfort.

Comfortable robot motion should also pose a constraint on robot noise. Lohse, M et al. Did experiments where approach speed and sound volume of a robot were chosen according to four situations depicted in the figure below:

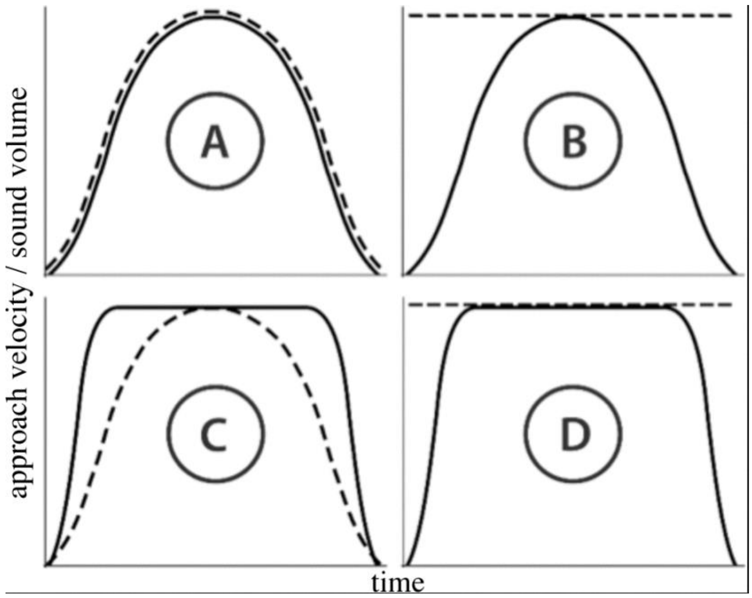


Figure 3 Four situations showing sound level (dashed line) and approaching velocity (solid line) profile of a robot, from Lohse, M. Et al.

It was found that robot sound volume has an impact on the perceived velocity of an approaching robot. They conclude that people tend to prefer a sound profile that almost matches the velocity profile (situation C), if we look at perceived safety. If we look at the likeability of a robot, sound profile A was preferred. In conclusion, no values of loudness were found in this research, but it is safe to say that some form of sound that scales with the velocity is preferred and would increase comfort levels and robot predictability for surrounding humans.

7. Behaviours disliked by society and the dominant culture should be avoided.

As described by Thibault, K et al., robots might need to prefer one side of the aisle for movement and/or avoidance, depending on country and culture. The robot might also need to ask or give cues to its environment if it wants to avoid a human or notices a human is blocking its path (as described under requirement 3). These aspects would make the robot more sociable, natural and predictable.

Describing cost functions

The most straightforward way to implement these user requirements and environment constraints is by making use of cost functions that can be implemented in avoidance procedures.

In order to find a path avoiding a human, in a sufficiently safe, comfortable, natural and legible way, a cost function can be used. This cost function assigns cost values to robot actions, depending mostly on environment and the robot's state. This cost function can be expanded to the environment's geometry, type and state, the person's age and gender, their current activity, the current interactions between people and interactions between people and objects. All this knowledge it has about its environment is stored in this cost function, which it tries to minimise when choosing a way to avoid collision. A visualisation of several cost function as a 2D map is seen in the following figure

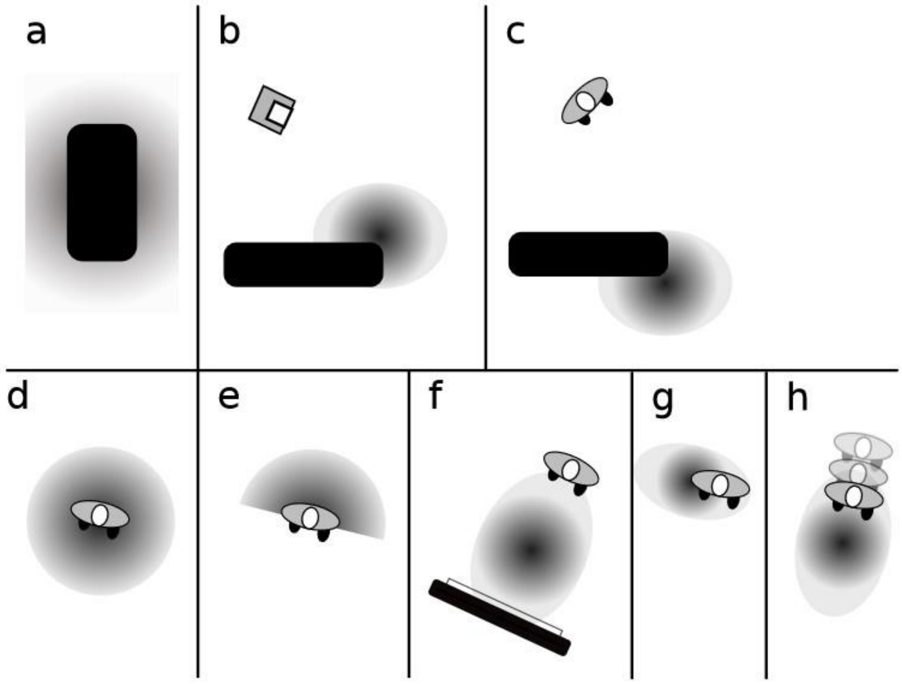


Figure 9: Visualizations of separate costmaps, thick black areas are obstacles, the square in b is a robot, human in h is moving. Areas shaded in grey have costs, meaning the robot should prefer to avoid those if possible. Shapes and sizes of cost function may vary and be context-dependent. All cost-functions can be combined.

Figure 4 visualisation of cost functions from Thibault, K et al.

Cost functions can incorporate the user requirements stated previously, by for example, modelling that moving closer to humans has less cost if done at low speed than at high speed. For planning algorithms, a problem would be that search space increases tremendously, resulting in a loss of robot reactiveness. A combination of the following cost functions is thought to be appropriate for a supermarket environment.

- Object padding (seen in figure 4a)

Object padding can be useful so that the robot does not move too close to or too fast near supermarket shelves, possibly causing misaligned products to fall out of shelves creating further complexities.

- Object occlusion and hidden zones (seen in figure 4b and c)

Due to the chaotic nature of the environment, people can come rushing around corners possibly leading to unexpected collisions with robots that are just behind line of sight for humans. The robot should know these locations and avoiding them or choosing a lower speed is desirable.

- Basic comfort distance (seen in figure 4d)

Following the previously described user requirements, every person’s personal space needs to be avoided as much as possible. An example of a procedure that incorporates this is given by Barnaud, M.-L et al.[14] who proposed a model that maps this personal space on the environment through a 2D normal distribution as a cost function, which can be used for collision avoidance. It was also found that interaction space, being the space in between two humans conversing or interacting in some way, was not necessary to model for these procedures. This model was successfully validated with experimental results with an actual robot. It showed that these procedures were perceived as safe by humans while also maintaining efficiency. This personal space model is used later.

- Passing people on their left (seen in figure 4g)

Passing people on their left is a social convention that should be preferred by the robot during collision avoidance. This is mostly a convention when a person avoids someone from behind. During face to face interactions, people tend to look in the direction they want to go. In these situations, this information should be used instead for collision avoidance as it is predictable and perceived natural for humans.

- Space ahead for moving (seen in figure 4h)

In general, robots should avoid moving in this space, as it hinders people. This cost function does require some form of motion prediction.

Most cost functions have growing costs as the distance to some entity decreases. This can of course be tweaked to exponential or other functions. Except for the personal space model, choosing linear cost functions might suffice. Combining these cost functions can be done via weighted sums. Cost function shape, combination and weighting can be tweaked manually or through machine learning.

Distinguishing between humans and objects

This distinction is needed, because humans will be avoided in a more advanced way than moving or static objects. This can be achieved through object recognition; however, this subject is beyond the scope of this research. Neglecting this aspect will make it so that only a distinction between moving and static objects will have to be made by sensors. By avoiding all moving objects in the same way as humans would be avoided, the main problem is slightly simplified. For real world applications this distinction can of course not be neglected, but the velocity detection discussed in the next section can easily be extended with the recognition of human beings.

Distinguishing between moving and static objects

The robot can make this distinction through object recognition as shown by Wei, Z et al.[15] This approach makes use of feature-line flows and distinguishes moving from static objects by computing residual errors. Although it will not be discussed here, research on how object recognition for navigation and collision avoidance might be used in supermarket environments is very important.

Assessment of possible collision avoidance procedures

Collision avoidance procedures will now be assessed on their application in a supermarket environment. Initially, the main aspects of the algorithm are described, then the degree in which these approaches can satisfy user requirements is looked at. Finally, a conclusion is drawn on how this approach might need to be adapted or extended to better fit the environment.

Dynamic window approach

The dynamic window approach by Fox, D. et al.[16] will be discussed

The dynamic window approach describes robot motion directly in the space of velocities. It reduces the search space to a dynamic window, which consists of the velocities reachable within a short time interval. These velocities are only admissible if the robot is also able to stop completely and safely in this time-span. It makes use of an objective function which measures the progress towards a goal location, forward velocity and distance to the next obstacle on the trajectory.

This approach models velocity as a piecewise constant function in time. It is thus assumed that robot trajectories consist of finitely many segments of circles. Intersection between circles and obstacles are used for collision checking. The approximate motion equations for x and y coordinates are described as follows:

For the x coordinate:

$$x(t_n) = x(t_0) + \sum_{i=0}^{n-1} \left(F_x^i(t_{i+1})\right) \qquad \text{(Eqn. 1)}$$

$$F_x^i(t) = \begin{cases} \frac{v_i}{\omega_i}(\sin\theta(t_i) - \sin(\theta(t_i) + \omega_i \cdot (t - t_i))),\ \omega_i \neq 0 \\ v_i \cos(\theta(t_i)) \cdot t,\ \omega_i = 0 \end{cases} \qquad \text{(Eqn. 2)}$$

And analogously for the y coordinate:

$$y(t_n) = y(t_0) + \sum_{i=0}^{n-1} \left(F_y^i(t_{i+1})\right) \qquad \text{(Eqn. 3)}$$

$$F_y^i(t) = \begin{cases} -\frac{v_i}{\omega_i}(\cos\theta(t_i) - \cos(\theta(t_i) + \omega_i \cdot (t - t_i))),\ \omega_i \neq 0 \\ v_i \sin(\theta(t_i)) \cdot t,\ \omega_i = 0 \end{cases} \qquad \text{(Eqn. 4)}$$

These equations make use of a discrete set of time steps (n).

v_i is the translational velocity at timestep i

ω_i is the rotational velocity at timestep i

$\theta(t_i)$ is the global orientation of the robot

These equations only depend on velocity, but these velocities can of course not be chosen arbitrarily. They need to follow from the dynamic situation the robot is in.

The search algorithm decides what velocities are admissible, which they are if the robot is able to stop before it reaches the nearest obstacle. Also, these velocities are restricted in that only velocities that can be reached in a short time interval (the dynamic window) will be chosen.

The robot then maximises the objective function, by picking a trajectory that maximises its translational velocity and the distance to obstacles but minimizing the angle to its goal relative to its own heading direction.

The main disadvantage of this approach is that it does not consider at all what kind of obstacles are in the environment and it only assumes static objects are present. There is no distinction made between moving and static objects, but more importantly, it does not consider that humans might need to be avoided differently. This approach also does not benefit much from the use of a top-down view as this approach is purely based on local reactive planning. An advantage of this approach is that it is very explicit about its movement trajectory through the functions for x and y that only depend on translational and rotational velocities.

Because of the disadvantages, the algorithm as it is presented here is not very viable for a supermarket environment. The restricted admissible velocities that result from this approach do make sure that erratic motion of the robot is prevented. This means that only user requirement 5 can be satisfied.

To make this approach more viable for a supermarket one will need to introduce the concept of moving obstacles, therefore needing an extension with motion prediction. If more user requirements are to be satisfied, this approach should also be extended with previously described cost functions and constraints formulated in the user requirements section. Seder and Petrovic[17] describe the dynamic window approach with motion prediction. Henkel and Xu[18] describe the extension with a cost function, but this cost function has nothing to do with enhancing the human robot interaction during collision avoidance. In conclusion, the dynamic window approach to collision avoidance requires too much adaptations for application in a supermarket. A different approach needs to be chosen.

Social force model

The Social Force model as described by Ratsamee, P. et al.[19] will be discussed now.

This is a very promising approach, since it aims to predict human motion through calculated social forces and then uses it in robot path planning. Social forces are described as inner motivation of a person to reach a certain goal. This path planning is perceived as human-like, because its path is natural, smooth and very much predictable for other human beings in the same environment. This approach specifically also distinguishes between objects and humans by analysing people's face pose. People tend to look in the way they want to avoid a certain obstacle or other person, so this is very valuable information when an avoidance that is predictable by humans needs to be executed. So, this approach considers the physical constraints of avoiding obstacles as well as social constraints.

This approach works by calculating a resulting force, $\sum F$, for changing the motion of individual humans or robots. This resulting force is calculated from F^{goal}, an attractive force that leads the human towards his goal, F^{object}, a repulsive force from other objects and F^{human}, a repulsive force from other humans:

$$\sum F = F^{goal} + F^{object} + F^{human} \qquad \text{(Eqn. 5)}$$

F^{object} and F^{human} are then calculated from a combination of social repulsive foces, f^{social} and physical repulsive forces $f^{physical}$.

For incorporating the face pose of surrounding humans, a new force is added:

$$F^{facepose} = FS \, * \, e^{\frac{r_{i,R} - d_{i,R}}{s_R}} \, * \, \overrightarrow{v_{i,R}} \, * \, (\lambda + (1 + \lambda)\frac{1+cos(\theta)}{2} \qquad \text{(Eqn. 6)}$$

In this formula, the following holds:

FS is a constant term that represents the strength of the face pose effect.

s_R is the range of the force

$d_{i,R}$ is the distance between the robot and human

$r_{i,R}$ is the sum of the radius of robot and human

$\mathbf{v}_{i,R}$ is the face pose vector from a human related to the robot. This describes the force direction.

θ describes the difference in angle between a human's face pose and the robot's.

λ is the anisotropic factor related to the difference in heading of a person and a robot (cosine term in $F^{facepose}$)

$F^{facepose}$ is summed with the other forces, resulting in:

$$\sum F = F^{goal} + F^{object} + F^{human} + F^{facepose} \qquad \text{(Eqn. 7)}$$

A path planning for robot R and a motion prediction for human H is then derived from the differential equation:

$$\frac{d}{dt}\vec{v} = \frac{\sum F}{m} \qquad \text{(Eqn. 8)}$$

The following figure shows an overview of calculated forces acting on a human (H) and a robot (R) during collision avoidance. F_i represents the resulting force ($\sum F$).

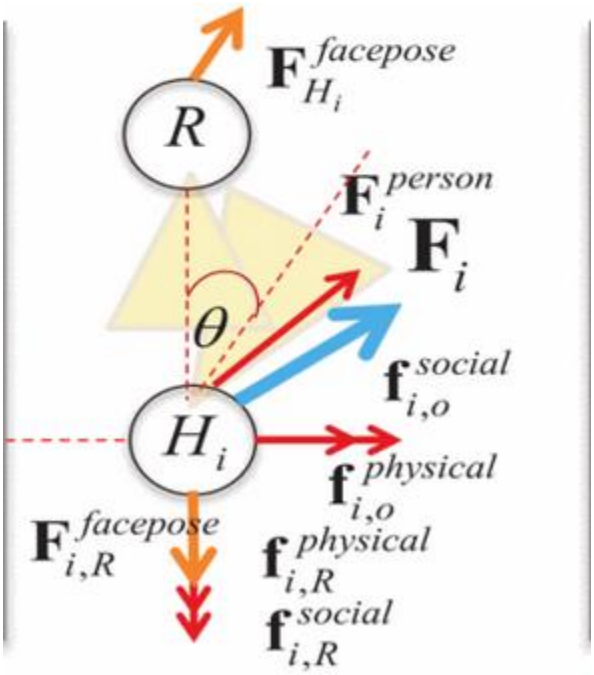


Figure 5 overview of forces from the social force model, by Ratsamee, P. et al.

Now that this approach is briefly described, it is important to look at which user requirements are satisfied and if this model possibly needs adaptations or extensions for use in the supermarket collision avoidance environment described previously.

First of all, this model can nicely incorporate the notion of a human's personal space through f^{social} and $f^{physical}$ that can be defined so that f^{social} grows in size when entering someone's personal space. This roughly satisfies user requirement 1, although the force definition greatly influences how well this user requirement is satisfied. For example, these repulsive forces might be too strongly present resulting in the robot never entering someone's personal space which in turn could lead to having no way to avoid crowded spaces. User requirements 2, 4 and 5 can also be satisfied. Because the robot tracks the face pose of nearby humans, it is able to plan a predictable and non-erratic path around a human that also adapts it velocities accordingly. Therefore, the following problems are taken care of by the model:

Blocking a human's path as described under user requirement 2 is evidently avoided because the approach will detect the human beforehand and plan a path around it. This is of course under the assumption that either the ceiling mounted camera or the robot's local camera senses this human.

A discomforting velocity as described under user requirement 4 is prevented, because the velocity is adapted according to the previously described differential equation which considers all social and physical repulsive forces. These forces need to be calibrated well, so that moving closer to human does indeed result in lower robot velocities. There are however extensions needed to constrain these velocities; under every condition a robot's speed should have a maximum value of 1.4 [m/s]. This can be constrained by adding frictional forces. Also, a minimum velocity of 0.5 [m/s] should be chosen when the robot happens to be (briefly) in someone's intimate space and not obstructed by someone's physical space. In that way, user requirement 4 should satisfied.

Erratic motions as described under user requirement 5 are avoided if all the forces calculated do not change significantly in a short time span leading to robot paths changing rapidly. Correct placement of sensors on the robot or the environment can prevent this. To further prevent erratic velocity profiles, a maximum robot acceleration based on the average pedestrian acceleration of 0.68 [m/s^2] should be added.

User requirement 7 might also be partially satisfied as the path planning algorithm makes use of a person's gaze through $F^{facepose}$. It is thought that this makes the path planning very predictable and readable for surrounding humans. This would make the approach more easily accepted for humans in general, probably also in a variety of countries and cultures. Empirical evidence needs to be found that this model for robots is indeed predictable and generally accepted by humans, however, Wang, P.[20] says that the social force model already is consistent with psychological findings regarding for example (interpersonal) stress.

In conclusion, this approach needs no extension with motion prediction of moving objects, because the algorithm presented works on both humans (for prediction) and robots (for path generation) and combines both to form a robot path. Overall, both humans and objects can be avoided in a desired way in conformity with most user requirements after some adaptations. However, it might be desirable to add cost functions to the static environment like object padding for the aisles and object occlusion (hidden zones). This approach can also benefit from the use of ceiling mounted cameras in the environment, because then blank spots or errors in the local sensing of the robot, possibly causing erratic calculations of forces, can be avoided. Static objects already in the environment can also be detected by these cameras, which is beneficial for this approach when calculating object repulsive forces.

A disadvantage of this approach might arise in the case of peak customer times where some supermarket aisles can be densely crowded. When large groups of people are walking around or standing in an aisle, there is a significant increase in the amount of forces that need to be calculated in real time, which might lead to a decrease in robot reactiveness to the environment. It needs to be investigated how many calculations are admissible to keep robot reactiveness.

It might thus be beneficial to add a concept of human groups to the model. Because these people generally want to stay inside the group there should be extra attractive forces added so that group cohesion is maintained as much as possible. It was found that the Headed Social Force model introduced by Farida, F. Et al.[21] incorporates this aspect.

Another disadvantage inherent to the social force model lies in the fact that humans and robots are defined as particles. Radii of the agents is incorporated in repulsive force calculation, but there is no physical constraint for agents so that they might be able to move through each other in this model. In a supermarket application this simplification cannot be made, because avoiding actual physical collision should of course be considered as a top priority.

A solution might be found when looking at Zeng and Bone's work[22]. This approach to collision avoidance also makes use of repulsive and attractive forces, just like the social force model. However, it clearly defines critical regions around humans, dynamic objects and static object. This area is defined in the following figure:

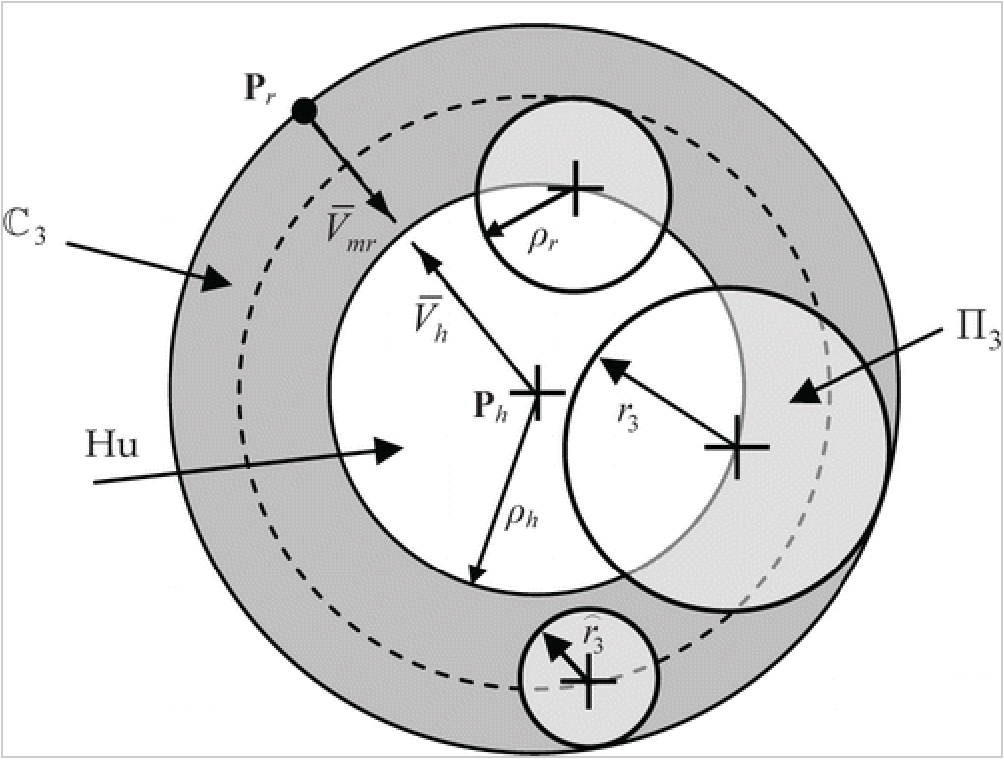


Figure 6 The critical region for a human as defined by Zeng & Bone.

Since the robot will move on the floor, a human should be modelled as the projected shape of its body on the floor. This projected shape is dependent on the pose of the human, but by modelling a human body as a cylinder the different poses are neglected, while still maintaining safety. Since the average step length of a human is 0.8 m [23], the radius of this cylinder ρ_h is taken as 0.4 m. The robot's radius is defined as ρ_r, so when the robot's centre is located along the dashed line, it makes contact with the human. This approach states that the robot should decelerate with maximum deceleration when it enters C_3. Since this critical region will only be entered by the robot in a worst-case scenario, another region should be defined where collision avoidance can take place. Zeng & Bone define this as follows:

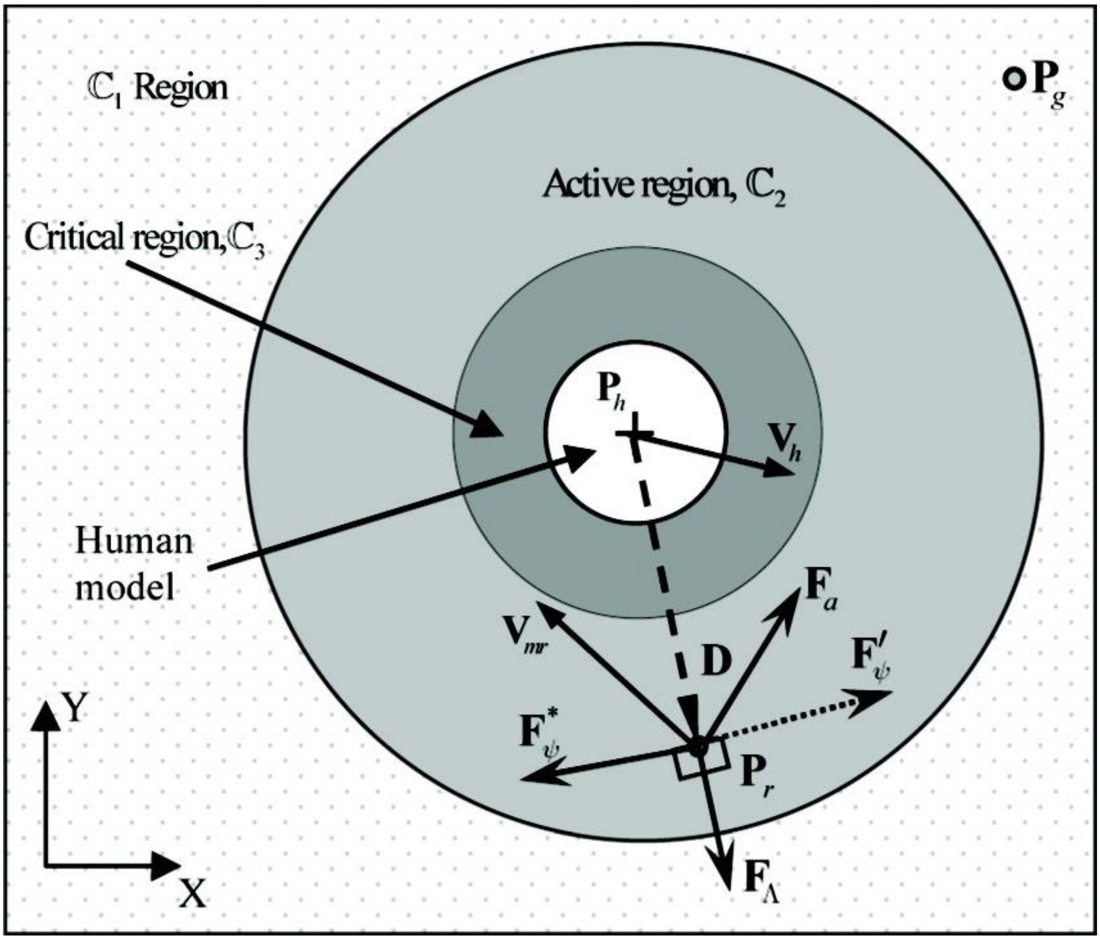


Figure 7 The active region of a human where collision avoidance should take place

This active region is where social repulsive forces should become larger than zero, so the trajectory and velocity of the robot can be adapted for collision avoidance.

It can be concluded that the most viable approach is given by the Social Force model, mainly because the Dynamic Window approach only takes into account static objects, while the environment of a supermarket consists of many moving objects. A distinction between inanimate objects and humans is also necessary, which the DWA does not consider. Significant adaptations need to be made to make this approach viable here, so it is considered inferior. The social force model on the other hand can relatively easy be extended with extra forces or environment cost functions and constraints to better fit the user requirements for a supermarket.

Now that the Social Force model is considered as the most viable collision avoidance option and necessary extensions are described, a simulation with this extended approach is necessary to test its working potential.

Candidate procedure simulation

The main aim of the simulation is to show that the adapted social force model is indeed suitable for application in a supermarket environment, while the standard social force model is not. The collision avoidance environment previously described will be made explicit by giving its exact geometries, describing all agents and obstacles present in the environment. Furthermore, formulae will be presented by which this algorithm can be tested in how well certain user requirements are satisfied.

Testing user requirements

The model should track how many problems with regards to HRIs (and thus user requirements) have occurred.

By calculating the amount of erroneous actions taken by the robot in this environment and relating this to a negative score, a conclusion can be drawn on whether user requirements are satisfied better with the adapted SFM compared to the standard SFM.

Therefore, the simulation needs to keep track of the following:

- How many times actual physical collisions happen and at what velocity (related to user requirement 1).

Since this is the most undesired, this action will be highly weighted in calculations. Higher collision velocity should add more negative points. This can be implemented as follows:

A physical collision occurs:

If: the distance from the centre of the robot to the centre of the nearest human equals $\rho_h + \rho_r$. (This distance can never be less than $\rho_h + \rho_r$)

Then: the robot's velocity relative to the human at this time step ($v_{r,h,i}$) needs to be stored.

The total negative points P_c for the robot due to physical collision can then be calculated with:

$$P_c = w_c * \sum_{i=0}^{n} v_{r,h,i} \qquad \text{(Eqn. 9)}$$

Where,

w_c is a weight factor for this calculation, which should be substantially higher than the other negative points calculations. Taking the value for w_c 10 times higher than for other calculations should implement the undesirability of physical collisions happening.

$v_{r,h,i}$ is the relative robot velocity at physical collision i

n is the total number of physical collisions between human and robot that happened during simulation.

- For how long and to what degree a robot has entered someone's personal space and with what velocity. (related to user requirement 1 and 4)

The degree in which this is undesired for humans is dependent on what velocity the robot used while moving in a human's personal space and how far into someone's personal space a robot moved in. A high velocity results in more negative points P_p.

A human's personal space is quite large, having a radius of 1.2 [m] around a human. The cost function approach by Barnaud, M-L. et al. which was already described can be used here. The personal space cost will then be modelled as two 2D normal distributions joined together seamlessly. The normal distributions have an independent front variance σ_h and rear variance σ_r but have the same side variance σ_s. Their research showed that values for σ_h, σ_r and σ_s of 0.9, 0.1 and 1.5 (respectively) showed the best fit with experimental results. This means that this normal distribution is elongated to a human's heading.

So, if a robot enters someone's personal space from behind the cost function is defined as:

$$\mathcal{N}(x, y \; ; \; [x_0, y_0], \Sigma_r)$$

If the robot enters someone's personal space from the front the cost function is:

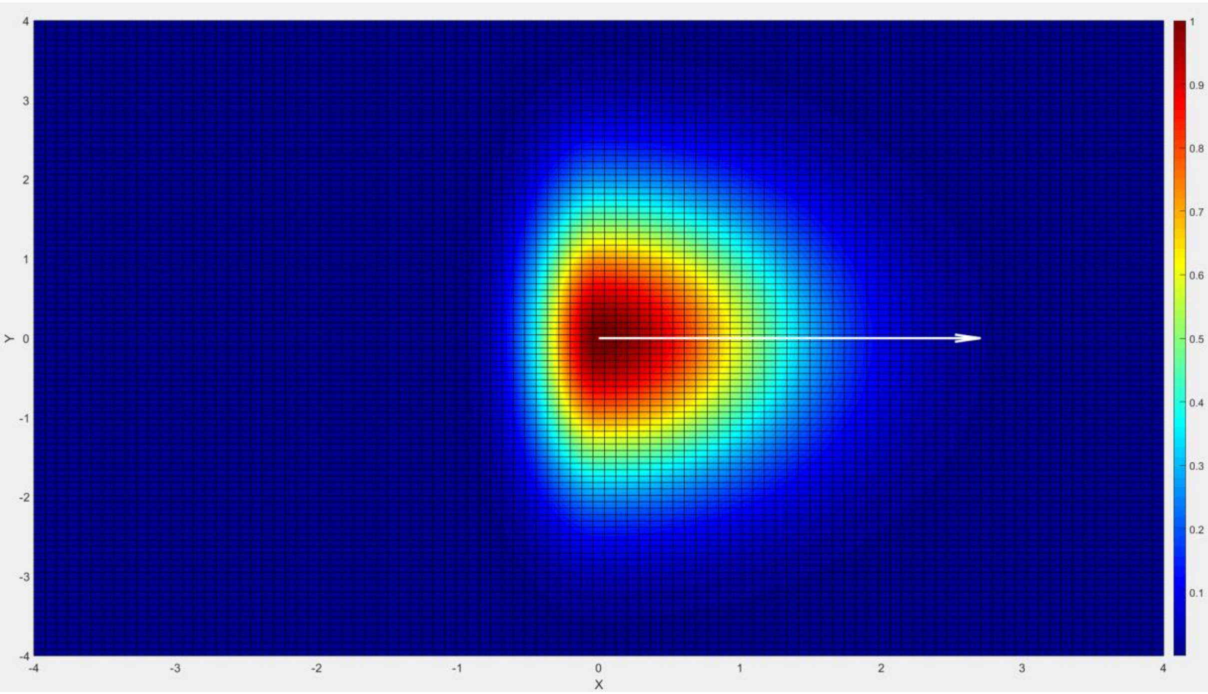
$$\mathcal{N}(x, y \; ; \; [x_0, y_0], \Sigma_h)$$

Where x_0 and y_0 are the coordinates of the centre of a human, thus the centre of the 2D normal distribution, x and y the local coordinates w.r.t. a human and with covariance matrices:

$$\Sigma_h = \begin{pmatrix} \sigma_h & 0 \\ 0 & \sigma_s \end{pmatrix}, \; \Sigma_r = \begin{pmatrix} \sigma_r & 0 \\ 0 & \sigma_s \end{pmatrix}$$

Calculations with this model will be stored in PS and are initially set to 0.

This 2D normal distribution is visualised in the following figure, with a white arrow indicating the direction of a human's heading:



To improve simulation performance and make sure every non-zero value of PS is considered, PS should be evaluated when the distance between the robot and a human ($d_{R,H}$) is less than $\rho_h + \rho_r + 2$ [m]

So,

While $d_{R,H} < \rho_h + \rho_r + 2$

PS(x,y) should be evaluated, with x and y the robot coordinates in the local coordinate system of a human, and multiplied by the robot's relative velocity to the human for every time step. This value then needs to be added to PS.

We can then derive the following formula for calculating the total negative points P for entering a human's personal space during the entire simulation:

$$P_p = w_p * \sum_{i=0}^{n} v_{r,h,i} * PS_i(x, y) \qquad \text{(Eqn. 10)}$$

$PS_i(x,y)$ is the evaluation of the 2D normal distribution at time step i and local coordinates x and y of the human. n is the total number of time steps the robot was present in someone's personal space.

w_p is the weight for this calculation

- Acceleration values of the robot (related to user requirement 5)

User requirement 5 simply states that the acceleration of a robot must not be higher than 0.68 [m/s^2]. This requirement should only hold when a robot is in someone's personal space, so if $PS_i(x,y) > 0$.

This can be checked by storing the robot's acceleration if it is higher than 0.68 [m/s^2] for every time step i. The difference in acceleration with 0.68 [m/s^2] (a_i) will then be summed for the entire simulation to calculate the negative points due to acceleration P_a:

$$P_a = w_a * \sum_{i=0}^{n} a_i \text{ (if } PS_i(x,y) > 0) \qquad \text{(Eqn. 11)}$$

w_a is the weight for this calculation

n is then the total number of time steps in the simulation

- How many times a human's path is blocked and for how long (user requirement 2)

This can be checked by storing, for every human, how many time steps (t_i) a human's (absolute) velocity (v_H) is equal to zero. Negative points for one human can be calculated as follows:

$$P_b = w_b * \sum_{i=0}^{n} t_i \text{ (If } v_H = 0) \qquad \text{(Eqn. 12)}$$

This should of course be tracked for every human in the simulation, except for any stationary humans.

The total negative points for the entire simulation is then:

$$P = P_c + P_p + P_a + P_b \qquad \text{(Eqn. 13)}$$

User requirements 1, 2, 4 and 5 can now be quantitatively assessed for the standard and adapted SFM in a simulation by comparing their P-values. These calculations were implemented in MATLAB function files.

Standard SFM simulation in Netlogo

Simulation with a standard SFM was done in Netlogo to see how well it performs. Static obstacles and a robot with a goal was added. It makes use of the standard interactive (repulsive) force, given by:

$$\vec{f}_{i,j}^{int} = -Ae^{\frac{r_{i,j}-d_{i,j}}{D}} \vec{\boldsymbol{e}}_{\boldsymbol{i,j}} \qquad \text{(Eqn. 14)}$$

Where,

$d_{i,j}$ representes the distance between a robot and a human or between two humans

$r_{i,j}$ represents the sum of the radius of a robot and a human or the radii of two humans, which is set to 0.5 [m] in the simulation.

A is the strength of the repulsive force, which is set to 1 [N] in the simulation

D is the characteristic distance of the repulsive force, which is set to 1.5 [m] in the simulation

The preferred velocity of the robot for reaching the goal was set to 1.4 [m/s]

The sum of the radii ($r_{i,j}$) is set to 1 [m], so that $\rho_h + \rho_r = 1$ [m].

Furthermore, the previously described scoring system was introduced with w_p set to 10 and all other weighting factors to 1. Unfortunately, because of limitations in Netlogo the personal space model (PS) had to be simplified by changing it to a function that is inversely proportional to the distance between human and robot, instead of the described 2D normal distribution. So, PS was changed to:

$$PS = \frac{1}{d_{i,R}} \qquad \text{(Eqn. 15)}$$

Where $d_{i,R}$ represents the distance between human i and the robot.

The environment was set to an infinite box (the environment wraps) where all orange humans are walking downwards, while all blue humans were walking upwards. This is used, because in a supermarket aisle, most people will walk in the direction of the aisle. A total of 500 humans were used, with a 50/50 distribution orange/blue, this done to test a variety of scenarios with high crowd density. A robot agent (white arrow) was implemented with a goal, represented by a green patch. Initially, no obstacles are present to only regard the avoidance of humans. The initial condition is given in the following figure:

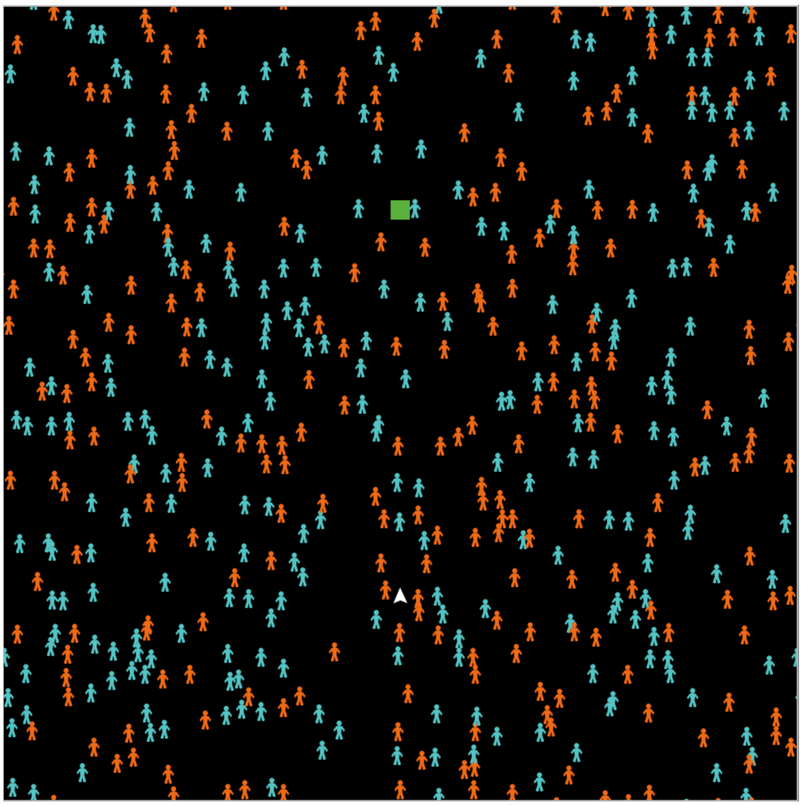


Figure 8 Initial configuration of Netlogo simulation

After a while the first collision avoidance occurs, which is successfully performed after which the goal is reached:

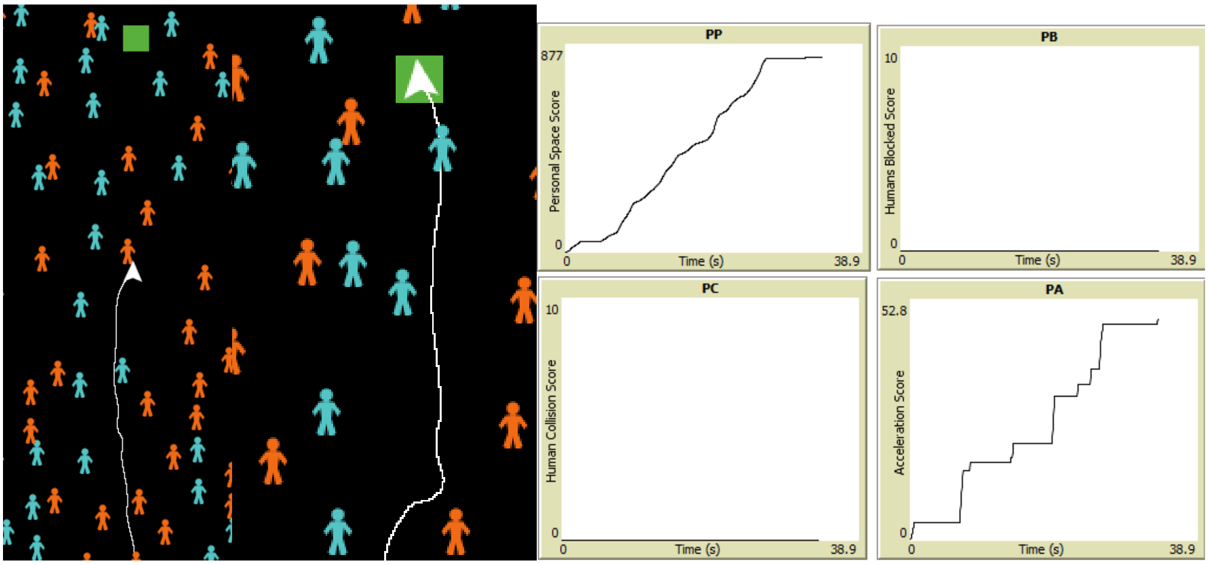


Figure 9 Start of collision avoidance, reaching the goal and score

Then if random static obstacles are added to the environment, the goal is reached, but it takes a very inefficient path and it gets stuck for a while:

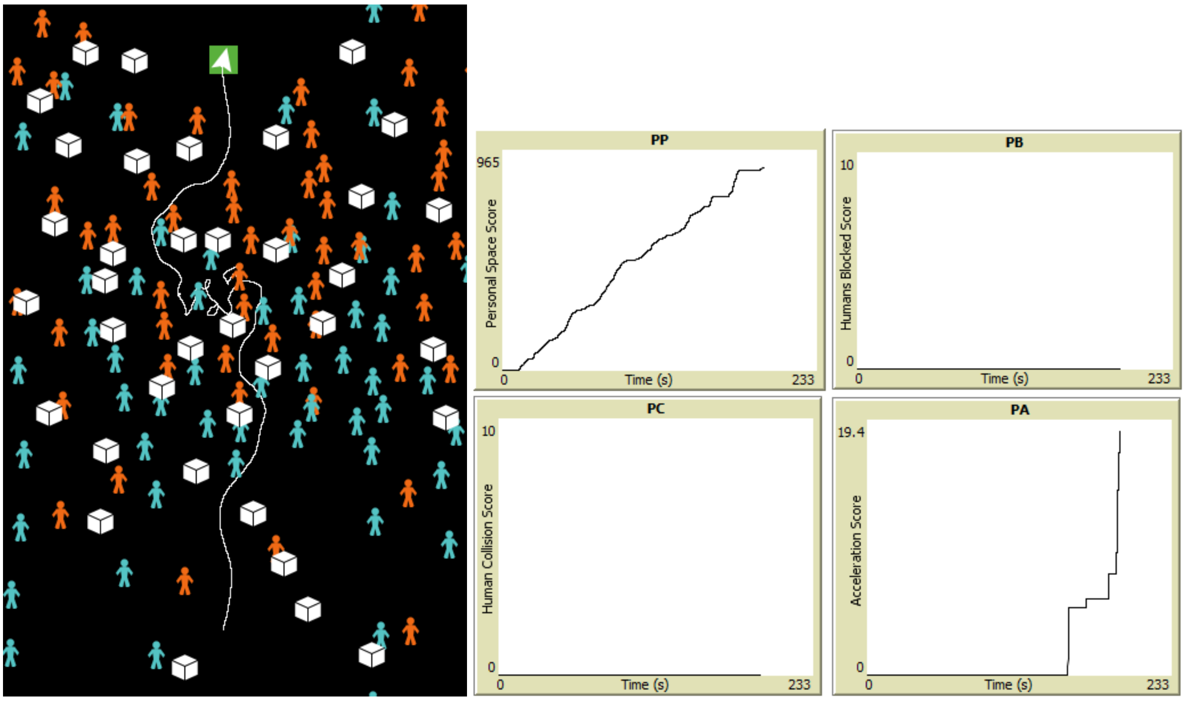


Figure 10 Collision avoidance path with random static obstacles and score

The SFM model was slightly adapted by letting the blue humans start behind the robot, and the orange humans all start at the top of the environment. Without any obstacles this lead to the robot having a difficult time avoiding the humans, and a high amount of personal space entries were detected.

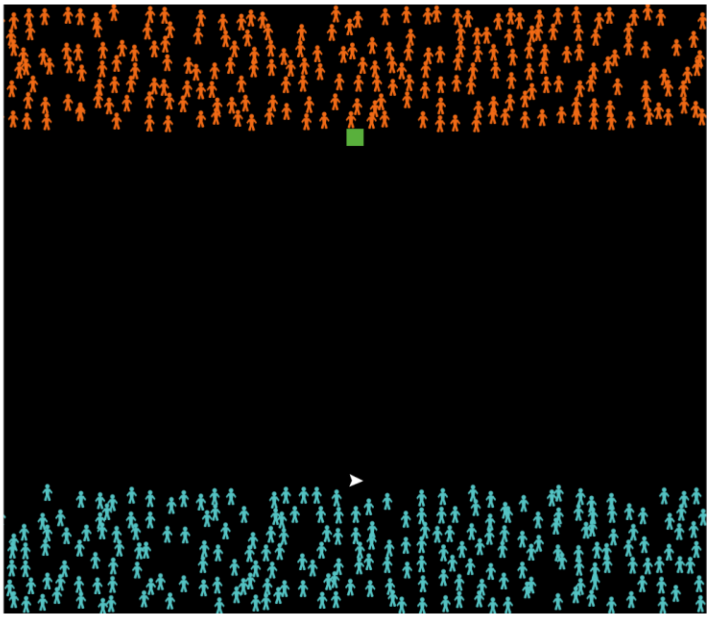


Figure 11 Initial configuration

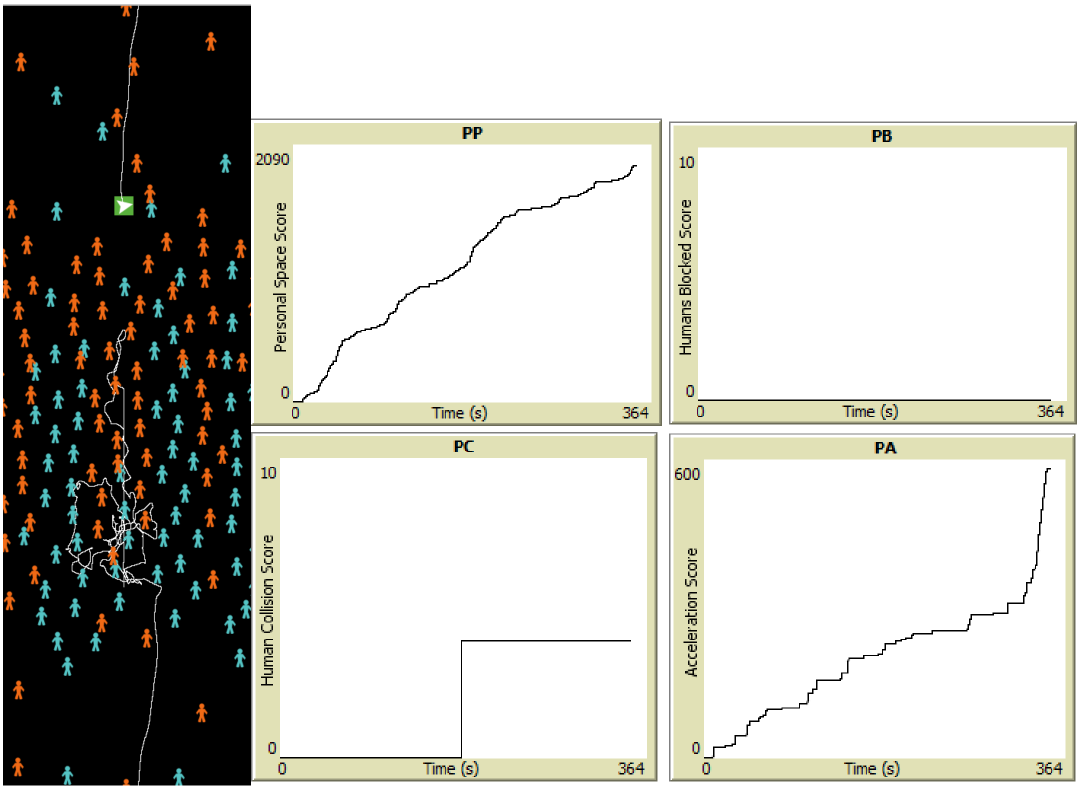


Figure 12 Netlogo SFM simulation with scores

Furthermore, it was found that to minimise P_p and P_a, a higher value for A and a lower value for D might be useful. Simulating with for example, A = 3.5 [N] and D = 0.9 [m] resulted in the following:

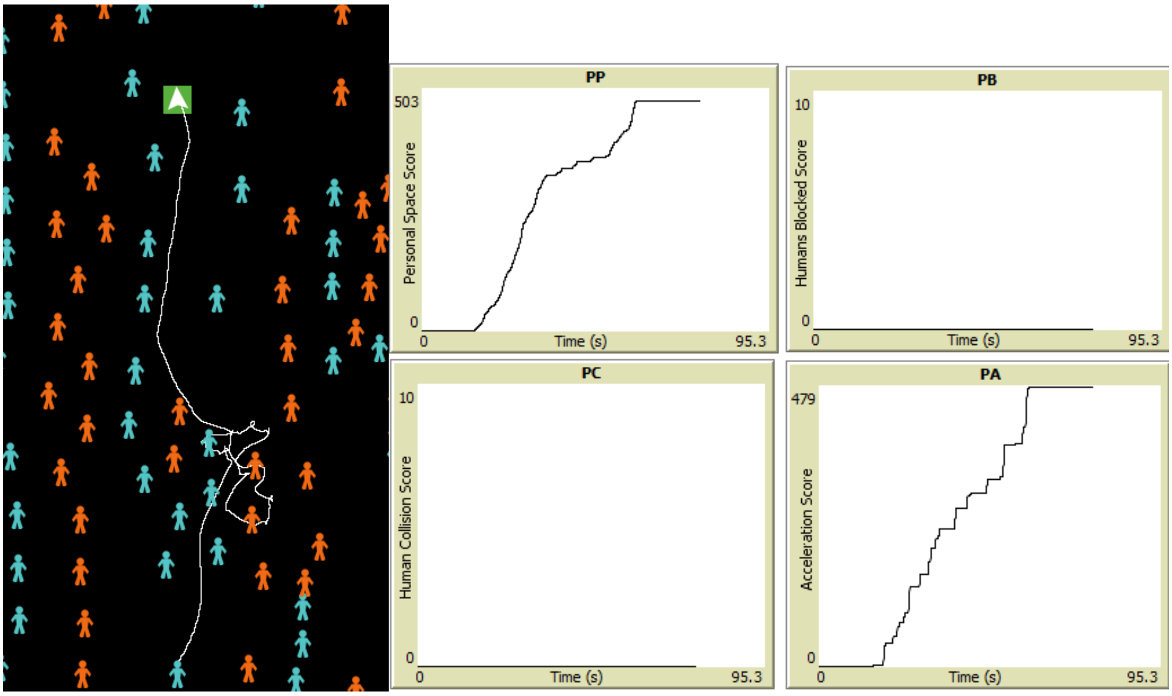


Figure 13 Simulation with higher A = 3.5 and D = 0.9 with scores

Environment simulation with HSFM

The headed social force model (HSFM) by Farina, F. Et al will be used as a basis for this simulation, since a MATLAB implementation is readily available, and it already incorporates the modelling of humans as cylinders instead of particles. Furthermore, it adds the concept of a human's heading which is in this case used to describe the direction in which humans tend to walk. This is necessary because it has been empirically shown[24] that people prefer to walk forward most of the time, while lateral displacements are rarely seen. This model also adds forces so that human group cohesion is maintained as much as possible. Also, velocities are constrained by making use of frictional forces. Unless stated otherwise, the model parameters from Farina, F et al. will be used.

The simulation environment is defined as follows:

- A corridor with a width of 3 [m], which is slightly above the average width of a supermarket aisle according to Steenblock, S., A.[25]
- The length of the corridor is taken sufficiently large: 10 [m]
- Four small obstacles of variable dimensions to account for fallen retail product, located near the corridor walls.
- Three parked shopping carts, with approximate dimensions W x L of 0.60 x 1 [m], which are the dimensions of shopping carts sold by shoppingcartmart.com[26]. They can be arbitrarily placed near the walls, so the shopping carts will all be placed near the left wall to make the environment less harsh. The geometry with dimensions given in inches is seen in following figure and is also used in the simulation:

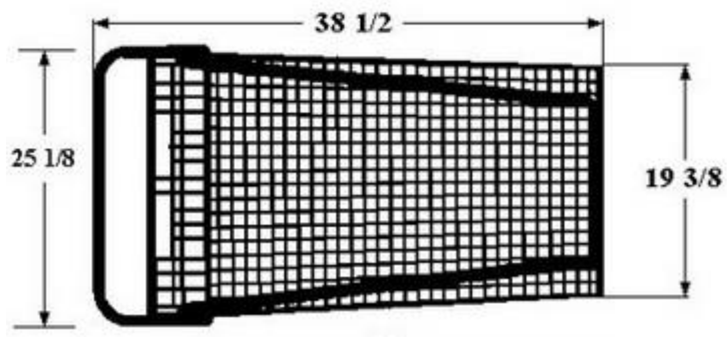


Figure 14 Dimensions in inches of a shopping cart

Implementation in the HSFM by Farida, F. et al gives the following environment:

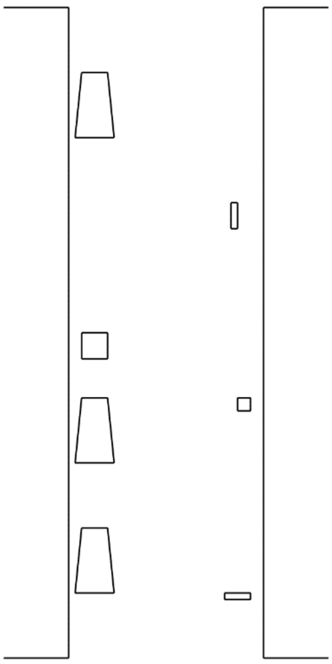


Figure 15 The simulation environment of a supermarket aisle in the HSFM

The agents present in the environment and their goals are described now.

All humans in the simulation will have the following characteristics: their radius ρ_h varies between 0.35 and 0.45 [m], their mass varies between 60 and 90 [kg] and their desired speed will vary between 0.5 and 1.4 [m/s].

In this model, the robot could have about the same characteristics as humans, because changing the characteristics of the robot does not have a great impact on the simulation. However, the robot can have a significantly smaller radius than humans, so taking ρ_r = 0.2 [m] should suffice.

To simulate an environment where several moving and stationary humans need to be avoided by the robot, the following agent groups should be added to the environment:

- Two groups, each consisting of 3 humans, each group entering the corridor from opposing direction will walk through the environment. Their goal thus lies outside of the corridor.
- Two groups of humans, both consisting of 2 humans will enter from either side having a distinct goal present at the walls of the corridor and will then move outside the corridor.
- Two humans will be stationary somewhere in the corridor.

- One robot will enter the corridor from the bottom and has its goal at the top end, to simulate a robot navigating through.

A (technical) problem with this model is that it makes use of groups of people that need to have at least two humans in them. Because of this, the robot needs to be programmed in separately, which was not achieved. It was found that the first three agent groups can be added to the model without much adaptation.

Simulating only the first two agent groups gave the following result:

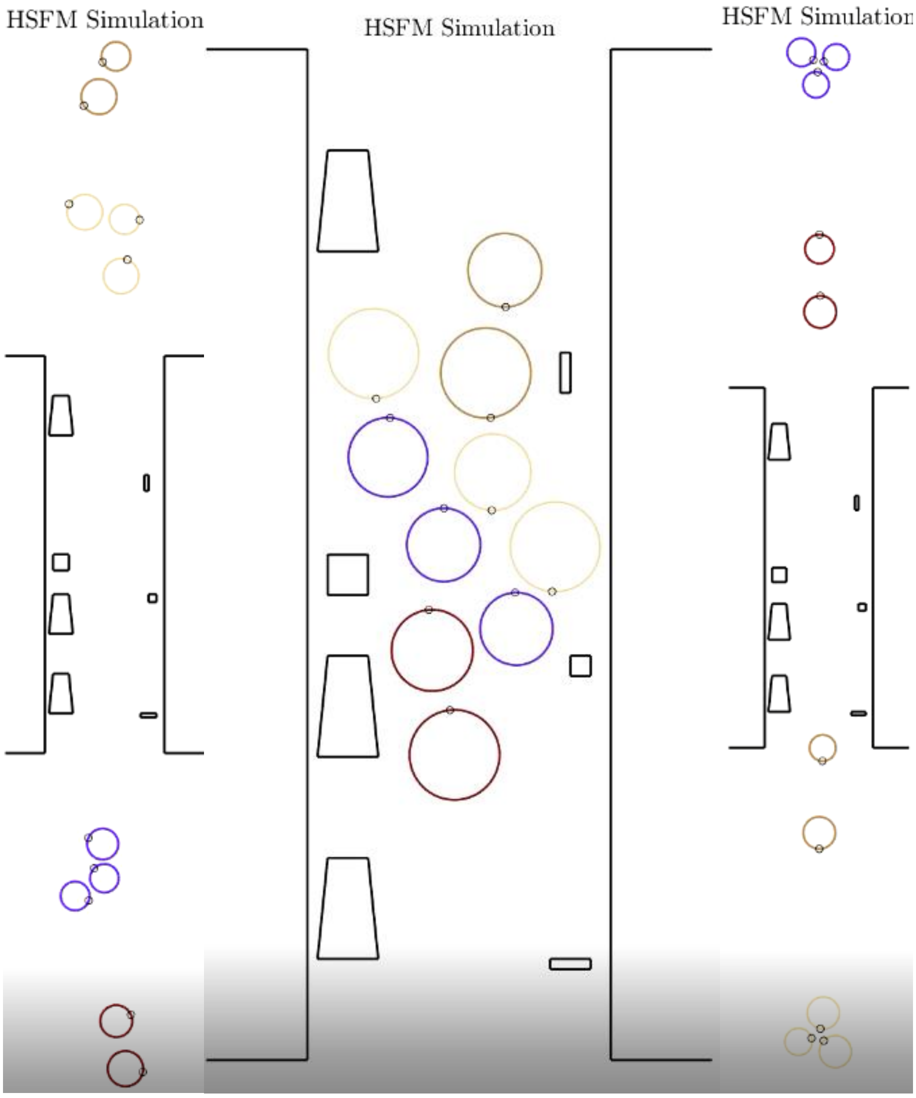


Figure 16 from left to right: Initial condition - humans avoiding each other - final condition (40 time steps)

Adding the third agent group leads to the following two final conditions:

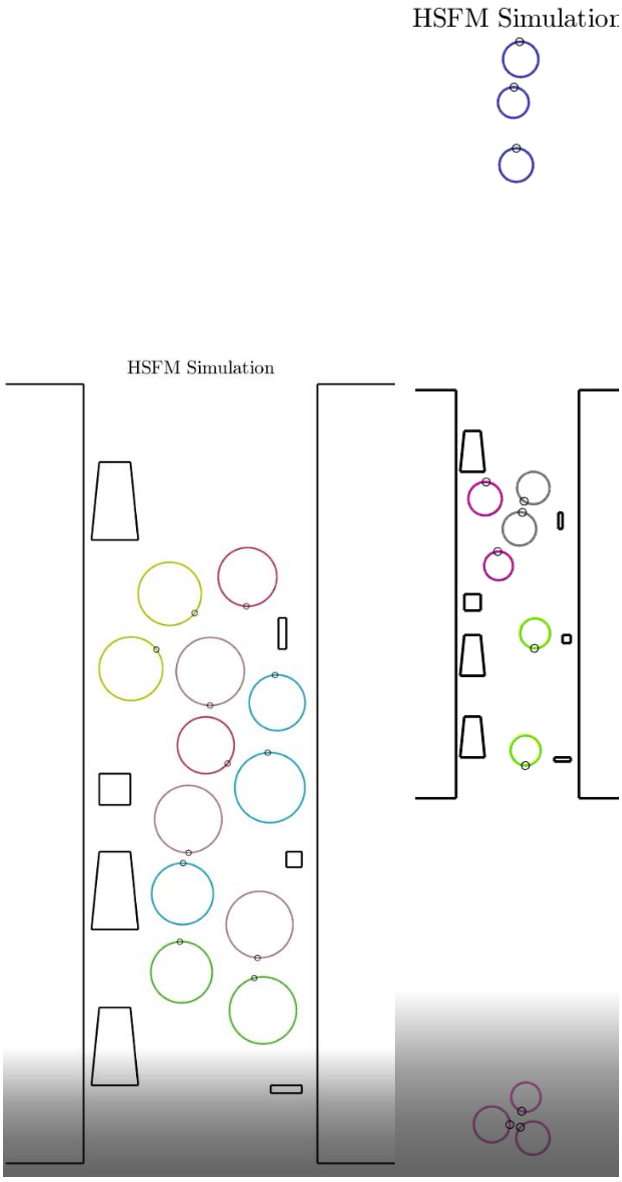


Figure 17 the final conditions after 40 time steps, simulating with the first three agent groups

A video of these simulations is also available on the wiki. Unfortunately, the user requirement tests could not be implemented in this model.

HSFM extensions

The first extension that needs implementation is adding $F^{facepose}$ to the HSFM. In the HSFM, θ_i is the heading, which is defined as the angle between x-axis of the body frame centred at the human's position and the x-axis of the global environment.

Then, the distance between human and robot $d_{i,r}$, the sum of radii of a human and a robot $r_{i,R}$ and this θ_i need to be used to calculate $F^{facepose}$, according to the formula of $F^{facepose}$ (Eqn. 6).

Where θ is now defined as the difference in heading of a human $\theta_{i,H}$ and a robot $\theta_{i,R}$:

$$\theta = \theta_{i,H} - \theta_{i,R} \qquad \text{(Eqn. 16)}$$

In the HSFM, f_i is denoted as the total force acting on an individual, $f_{i,0}$ as the attractive force to the goal and $f_{i,e}$ describes the repulsive and interaction forces. $F^{facepose}$ should thus be added to $f_{i,e}$:

$f_i = f_{i,0} + f_{i,e}$ (Eqn. 17)

And with:

$f_{i,e} = f_{i,p} + f_{i,w} + F^{facepose}$ (Eqn. 18)

The next extension is adding (static) cost functions to the environment. It is thought that the object occlusion, hidden zone and object padding cost functions are only necessary to be implemented in the HSFM. The other cost functions proposed previously would not be necessary to add, because they are already described well through the HSFM. Small static obstacles on the ground are already taken care of by the HSFM, but for the walls (supermarket shelves) cost functions need to be added.

These cost functions should have a direct impact on velocities and should thus skip the force calculations.

For object padding a cost C_p is defined, which is inversely proportional to the robot's distance to the walls $d_{i,w}$:

Only if the $d_{i,w}$ is smaller than or equal to 2 times the robot radius ρ_r, this cost function should be evaluated so that movement further away from the walls is not affected:

$C_p = \frac{1}{d_{i,w}}$ (if $d_{i,w} \leq 2\ \rho_r$) (Eqn. 19)

For object occlusion and hidden zones, a cost C_o is defined which is now inversely proportional to the distance to the centre of a certain critical zone $r_{i,c}$. In this environment critical zones are the ends of the corridor, where the robot might not expect people coming around corners.

The distance to the centre of this critical zone is defined as $d_{i,c}$. 2 ρ_r is now taken as the radius of this critical zone, leading to:

$C_o = \frac{1}{d_{i,c}}$ (if $d_{i,c} \leq 2\ \rho_r$) (Eqn. 20)

Where

$d_{i,c} = \|r_{i,R} - r_{i,c}\|$ (Eqn. 21)

$r_{i,R}$ represent the coordinates of the robot.

The coordinates of this critical zone's centre ($r_{i,c}$) should of course be programmed in manually.

The two cost functions can then be summed, applying weight factors:

$C = w_{padding} * C_p + w_{occlusion} * C_o$ (Eqn. 22)

The robot velocity $v_{i,R}$ should then be reduced, according to:

$v_{i,R} = v_{i,R} - C$ (Eqn. 23)

To make sure this does not stop the robot completely, the minimum velocity of 0.5 [m/s] should always be chosen if C gets too high.

A visualisation of the proposed cost functions to the HSFM simulation environment is given in the following figure:

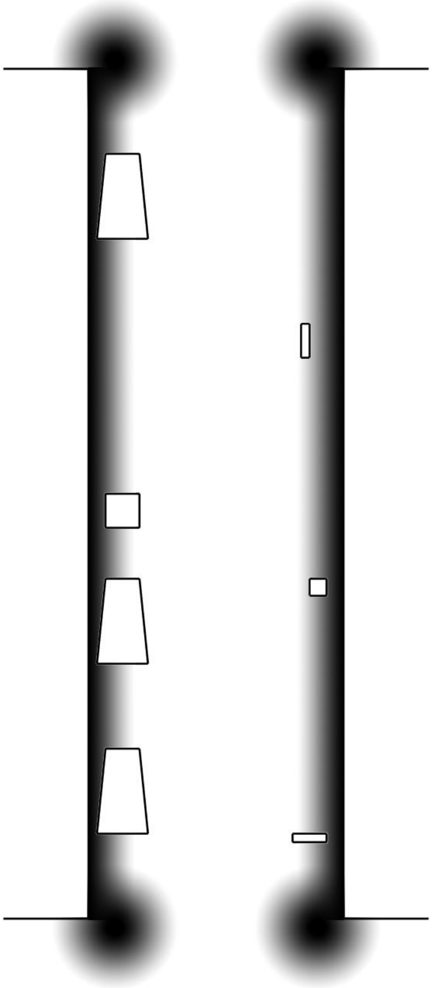


Figure 18 cost function mapping to the simulation environment

A simulation that implements the user requirement tests, with the extended HSFM should prove that this approach is superior for collision avoidance in a supermarket.

Extended SFM Simulation in Netlogo

Unfortunately, the HSFM could not be extended due to time restrictions. However, to test whether the HSFM extensions presented previously do have a positive impact on collision avoidance, the standard SFM in Netlogo was extended instead. So, $F^{facepose}$ was added to the repulsive forces and the object padding cost function (with $w_{padding} = 1$) was implemented. The occlusion cost function was left out, as the ends of the aisle are not represented in the simulation. To test a correct implementation of the cost function, the robot was initially placed near the left wall, with the goal also near the wall. The desired velocity was set to 1.4 [m/s].

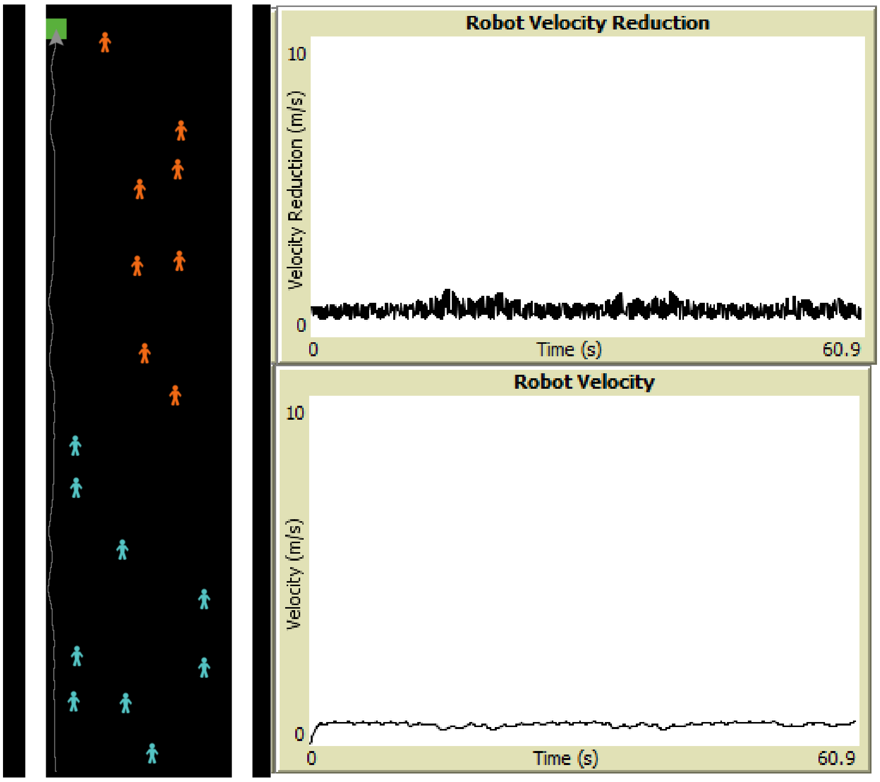


Figure 19 Simulation with environment cost function with a velocity reduction and robot velocity graph

As seen from the figure above, the velocities were adapted accordingly with a velocity range between 0.39 and 0.61 [m/s] and a mean velocity of 0.54 [m/s]. Doing the same test, but with robot goal and initial position in the middle of the corridor leads to no velocity reduction.

To show the difference in simulating with $F^{facepose}$, compared to the standard SFM the following example simulations were performed. The robot trajectory efficiency is assessed by visually expecting the taken path of the robot, which is drawn on the environment. Multiple simulations were done per simulation set up, the results presented here merely give a good representation of the average behaviour of the model.

Now, a simulation is performed with 41 humans, a desired velocity of 1.4 [m/s] was chosen and A and D were set to 3.5 [N] and 0.9 [m] respectively, which were the parameters that gave the best results with the standard SFM model. No $F^{facepose}$ or obstacles were added. The following results were obtained:

Figure 20 Simulation without $F^{facepose}$ and no obstacles

The robot reached the goal after 316 seconds and with relatively low P_p and P_a scores, compared to the previous simulations with the standard SFM. The trajectory of the robot was however very inefficient.

Adding randomly placed obstacles to the environment gave the following results:

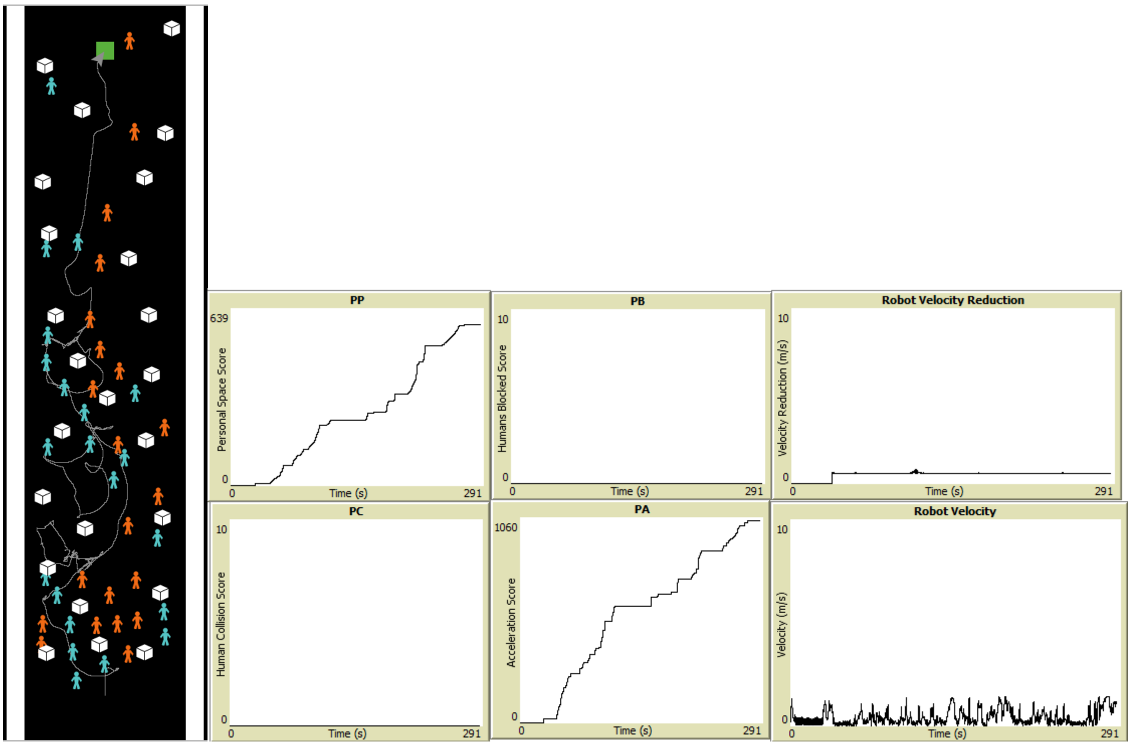


Figure 21 Simulation without $F^{facepose}$ and with obstacles

The robot reached the goal after 287 seconds, with a value for P_p around 2 times as high and a value for P_a around 4 times as high compared to the simulation without obstacles. The trajectory of the robot was again very inefficient.

Now adding $F^{facepose}$ with FS = 2 [N] and λ = 0.6 [-] and setting A to 1.5 [N] so that the total magnitude of the repulsive forces remains 3.5 [N] as before, we obtain the following results. The velocities had to be scaled down in some areas as velocities higher than 1.4 [m/s] were detected. When these higher velocities were detected, the speed in x and y direction were scaled down both in the same way according to:

$$v_x = \frac{v_x}{\sqrt{v_x^2+v_y^2}} \text{ and } v_y = \frac{v_y}{\sqrt{v_x^2+v_y^2}} \qquad \text{(Eqn. 24)}$$

The results were as follows:

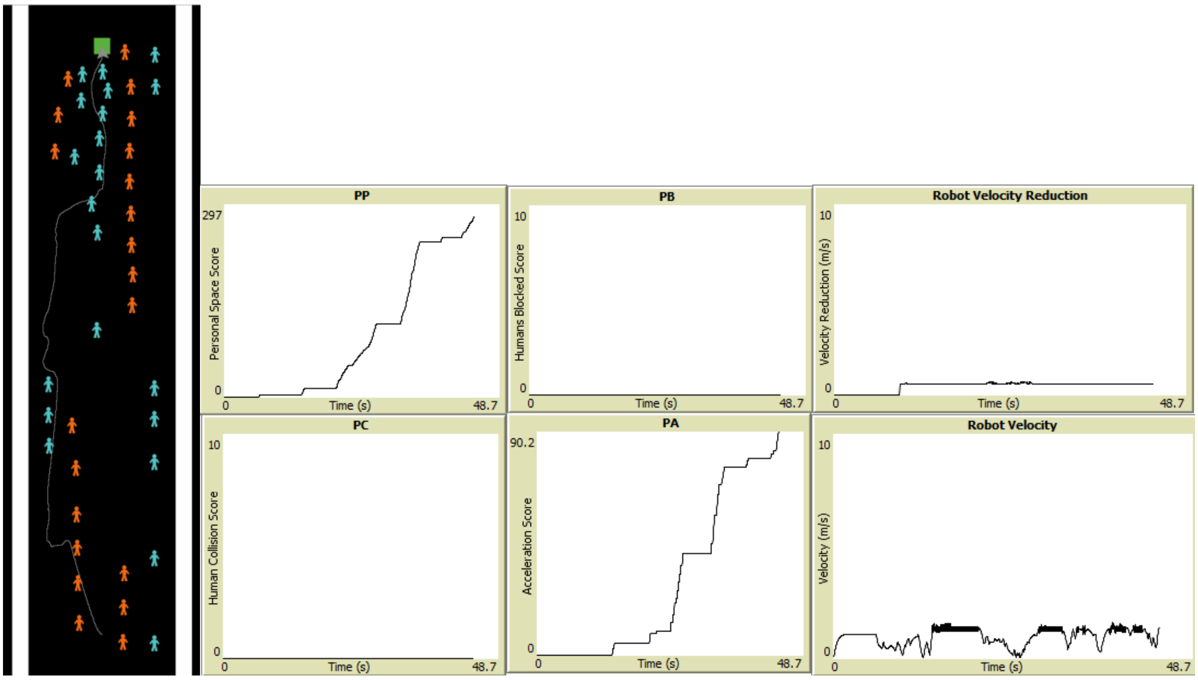


Figure 22 Simulation with $F^{facepose}$ and no obstacles

The robot reached the goal after 44 seconds with a similar P_p value and P_a almost 3 times as low compared to the simulation without $F^{facepose}$. Also, in this comparison the path is more efficient.

Adding randomly placed obstacles to the environment gave the following results:

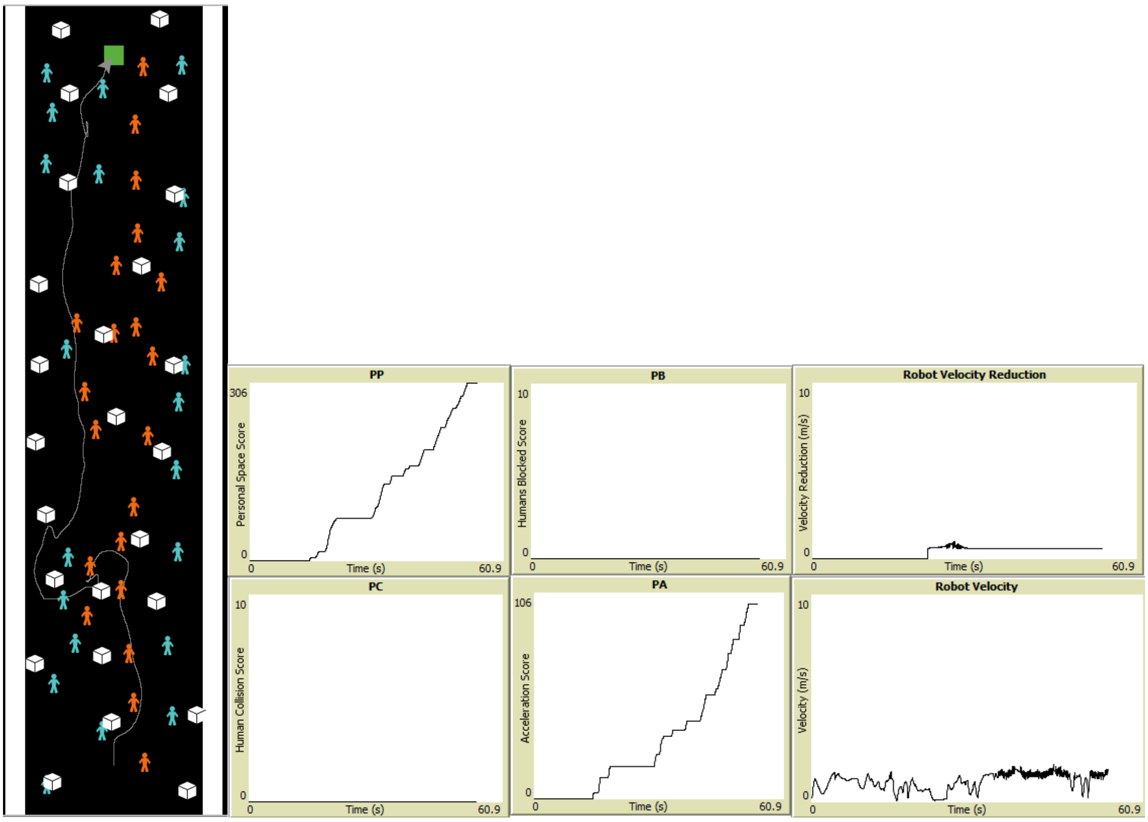


Figure 23 Simulation with $F^{facepose}$ and obstacles

The robot reached the goal after 54.5 seconds with similar values for P_p and P_a compared to the simulation without obstacles. As seen, the path is similar in efficiency. Comparing it with the standard SFM simulation with obstacles, P_a was around twice as low and P_a around 10 times as low. The trajectory was also more efficient.

Furthermore, there is a problem which is inherent to the standard SFM and this extended SFM. That happens when obstacles in U-shapes are placed in the environment. These result in so-called local minima where agents get stuck in these situations sometimes only getting un-stuck after long simulation times. An example can be seen in the following figure:

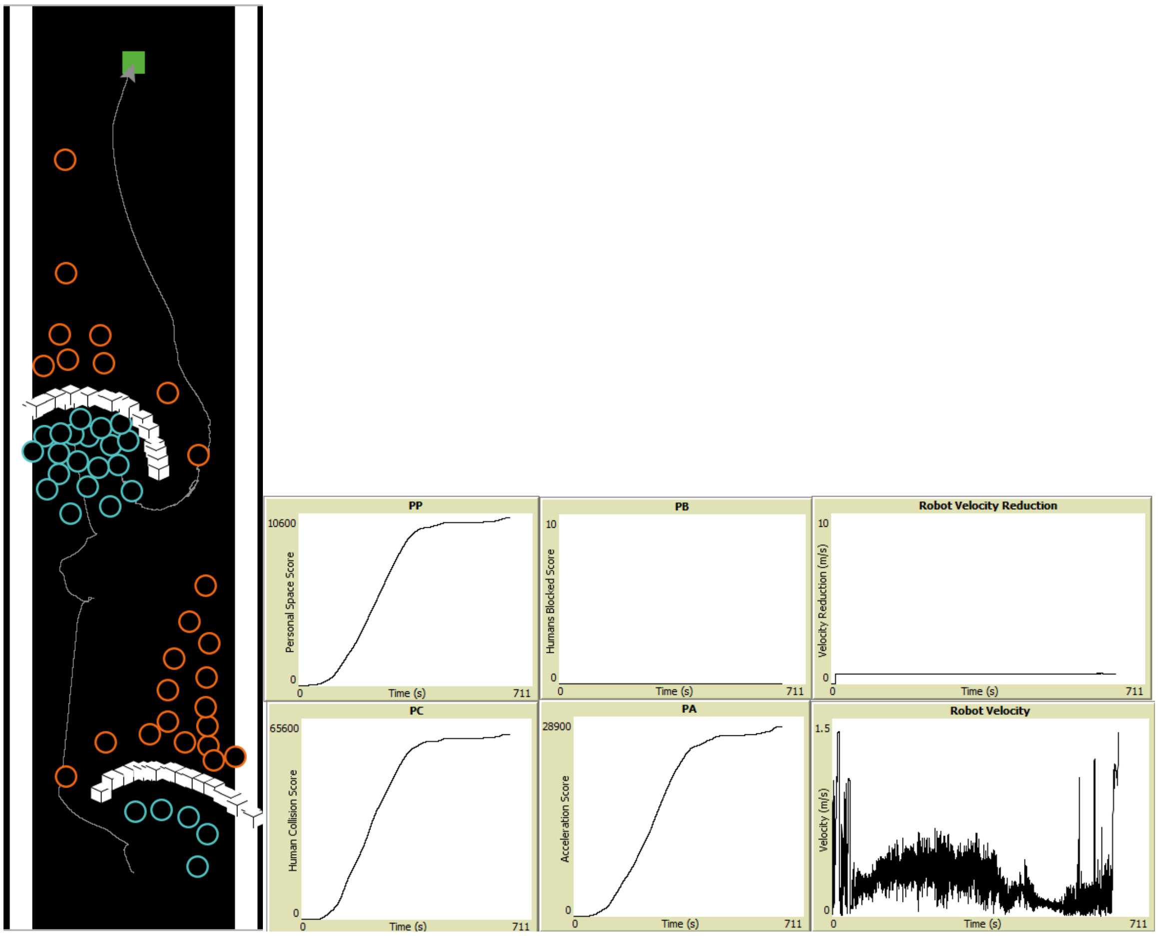


Figure 24 Simulation with U-shaped obstacles

After 643 seconds the robot reached the goal, being stuck for a majority of that time period. Also, the humans in the environment have a hard time evading these obstacles. The negative scores also skyrocket during these simulations.

Discussion

Standard SFM simulation in Netlogo

Comparing the first and second simulation, in both simulations no physical collision occurred, and no humans were blocked enough so that their velocity became zero. As for personal space entries (P_p), the second simulation (with obstacles) showed a higher score than the first simulation (without obstacles) but the profile over time looked similar. This means that the velocities and the degree in which a personal space was entered was similar, but it occurred more during the entire simulation. There was a drastically higher score for P_a in the first simulation, because the obstacles in the second simulation lead to lower velocities and accelerations by the robot. The path taken was also very inefficient in this simulation, compared to the first. It can be concluded that obstacles severely reduce efficiency and increase the time to reach the robot goal.

The third simulation showed deviating results. The robot had severe difficulties getting to the goal. Eventually, the repulsive forces lead to the robot being 'pushed' down far enough for it to use the shorter path of going down. There was also one physical collision that occurred with a velocity of around 0.32 [m/s], which is undesirable. The scores for P_p and P_a were also drastically increased compared to the first and second simulation. The fourth simulation showed better results for P_p and P_a, so choosing a stronger repulsive force while lowering the characteristic distance is beneficial for collision avoidance.

All in all, because the second and third simulation showed inefficient paths and relatively high negative scores on user requirement tests, this standard SFM is not viable for application in robot collision avoidance in a supermarket environment.

Environment simulation with HSFM

Looking at the first simulation with the standard SFM model, there is already quite a complex situation for robot collision avoidance when only 10 humans are moving in and out of the corridor. It should also be noted that the environment is not very 'strict'; a wider than average corridor is chosen, and the parked shopping carts are all relatively out of the way. If a robot would be introduced with smaller radius but equal characteristics as humans, it can probably already lead to the robot getting stuck and some user requirements might be violated. However, all groups do reach their destination after 40 time steps.

Then adding two stationary humans in the second simulation leads to the model having a hard time to get all groups to their destinations in 40 time steps. Sometimes no destinations were reached at all, or only 2 out of 5 groups reached their destination. Also, computation times increase with a factor of 6.

What we can conclude from this simulation is that humans do not really avoid each other by adapting their heading beforehand. Humans simply get really close to each other (around 0.05 to 0.1 m) and are then slowly directed around another human due to the repulsive forces. There is no actual collision happening because there is a constraint on the minimum distance between humans. Also, a lot of velocity is lost during these avoidances, so they are very inefficient. If a robot would take this approach to collision avoidance it would be very inefficient, because it should take a path around a human that still maintains some velocity and does not come too close to humans.

Additionally, user requirements would also be violated: Robots will come too close to humans, because with this approach someone's intimate space is always entered while this should be avoided if possible. It is also possible that robots will block a human's path, because in the second simulation all agents in the environment can get stuck and show no sign that they can get un-stuck eventually. Approaching speed is also too high, since agents only decelerate right before they get to the 0.05 to 0.1 m in-between distance. Erratic motions are however prevented, because this model adds an inertia term, a heading and group cohesion forces that even out velocity and path profiles.

Although it is not quantitatively measured, it is safe to say that this HSFM is not a viable approach to robot collision avoidance for a supermarket environment, because it is thought to be very inefficient in this environment and it violates user requirements 1, 2 and 4. However, the extension with agent headings, group cohesion forces, physical radii and an inertia term will be beneficial for implementation in robots for this application. Also, choosing different model parameters after robot implementation can lead to significantly better results with this model.

Extended HSFM

Although it cannot be quantitatively measured, the extended HSFM is thought to be superior than the regular SFM for robot collision avoidance in terms of efficiency in a supermarket environment because of the following points:

- It defines a heading for every agent so that only more realistic forward motions are performed by agents. These headings can then be used to describe a more efficient robot path around humans and make the environment more realistic.
- It adds cost functions to the static environment, so the robot will adapt its velocity accordingly in critical situations near the shelves to prevent it from nocking over misaligned retail products. Furthermore, its velocity will be constrained near the ends of shelves, so that unpredictable collisions with humans coming around the corner can be avoided. This would thus make the robot move more efficiently and more safely in the environment.
- It can define groups of humans that have cohesion forces that make them want to remain close to each other. This makes the collision avoidance environment more realistic as these groups can be present during more crowded situations in supermarkets.
- With the addition of an extra repulsive force $F^{facepose}$, the robot maintains more distance to the human, so it can move with a slightly higher speed, increasing efficiency.

The extended HSFM would also satisfy the user requirements more, because of the following points:

- It adds the concept of a physical radius for all agents in the environment so that physical collisions can be modelled, which is essential as the main goal of robot collision avoidance is to prevent physical collisions. The standard SFM is based on particles that have radii but does not define this as a physical constraint and is thus not viable for this application.
- Because of $F^{facepose}$, robots will prefer to pick a path that respects a human's personal space more. This path is also more predictable for humans. This should result in a better score for P_c and P_p.
- The acceleration of the robot is also easy to constrain due to the added frictional forces and inertia terms incorporated in the HSFM, so P_a should also give a better score.
- Again because of $F^{facepose}$, the robot will take a path around a human earlier in time because of this repulsive force, which makes sure less humans will be blocked in their activities. This should result in a better score for P_b.

Comparing simulations with the extended HSFM and the standard SFM would prove that the extended HSFM is indeed better for robot collision avoidance in a supermarket environment. This was unfortunately not achieved and needs to follow from future research.

Extended SFM Simulation in Netlogo

The conclusions drawn from these simulations are not very strong, since it would be preferred if all simulations were repeated more and all results were processed automatically and averaged out to give more reliable outcomes. Due to time restrictions and Netlogo limitations this could not be achieved.

However, if we take look at the results we can conclude that adding $F^{facepose}$ to the standard SFM is beneficial for both the tested user requirements and the efficiency of the robot trajectory. Overall, the time it took the robot to reach the goal was reduced, and the scores show better results.

There are some limitations to this extended SFM. Adding another repulsive force to the model resulted in some situations that higher than preferred velocities were chosen. Having velocities higher than 1.4 [m/s] is detrimental for human comfort and safety in general, so the velocities had to be scaled down. The approach to scaling down is rather unrealistic, so it is not desired to implement this. It is thought however that the inertia or friction term in the HSFM should provide a better solution to implement this. Another limitation for both standard and extended SFM happens when large U-shaped obstacles are present in the environment. These models cannot efficiently evade those, so it is important to find a solution for this.

User requirement tests

The negative point calculations described can of course only measure the degree in which user requirement 1, 2, 4 and 5 are satisfied. This point system with a chosen weighting is relatively arbitrary so absolute values are meaningless, but it should provide a good measure to compare different collision avoidance approaches. The remaining user requirements can unfortunately not be quantitatively compared and will need to follow from other simulations or experiments. User requirement 3 can only be experimentally tested by looking at how different robot cues affect collision avoidance in crowded situations. Testing user requirement 6 can be done by adding the concept of a sound source to every robot, but more investigation is needed what effect exact volume levels have on the perceived safety and likeability of the robot. This would need real life experiments with a robot as well. User requirement 7 can only follow from real life experiments. In different countries and cultures, the perceived safety and likeability of the robot for a collision avoidance approach should need to be investigated.

Cost function extension

The minimum velocity constraint used in the environment cost functions is the admissible approaching speed for robots interacting with humans described earlier. Since these cost functions are applied to the static environment and not humans, other velocities might be desired to increase movement efficiency.

Also, the weighting factors ($w_{padding}$, $w_{occlusion}$ in eqn. 22) used in the cost functions can be tweaked manually. Only simulations or even real-life experiments with these cost functions can give optimal values for these parameters to increase environment safety. It might also be necessary to investigate if non-linear cost functions are better suited for collision avoidance in a supermarket.

Topics for further research

Related to the environment

The topic of using ceiling-mounted cameras was briefly described but was not elaborated on. Although it was found that fish-eye cameras can indeed be used to make a top-down view of the environment, it needs to be investigated if this is beneficial for collision avoidance compared to using only local cameras/sensors on the robot. If it is indeed found that using ceiling-mounted cameras is beneficial, the exact amount and costs of installing these cameras needs to be found, and whether the benefits then outweigh the costs or not.

Moreover, it is important to investigate how robots should distinguish between humans and inanimate objects as well as between moving and static entities, and how this can exactly be implemented in collision avoidance procedures. This, in combination with finding a good way to make use of cameras in the environment as well as cameras on the robot itself are important topics for developing a complete solution for robot collision avoidance in a supermarket environment.

Related to HRI and USE aspects

For the situation of crowded environments, the topic of using robot cues to alert people was shortly described. It should be investigated what kind of cues are desirable in crowded situations. Moreover, it will still be necessary to do further research on which exact sentences using low controlling language are the most effective and fitting for this context, minimising social reactance. This can be achieved with more real-life experiments.

More (real-life) experiments need to be done to verify that the extended SFM applied to a robot in a supermarket environment does indeed lead to more desirable collision avoidances, from a user perspective. This should elaborate on how predictable and generally accepted the robot behaviour is perceived by people. In general, all user requirements presented in this research are important to do real-life experiments with, where the reaction of participants on robot behaviour is key. Moreover, since most user requirements presented here are generally applicable to any humans in public spaces it may be useful to conduct surveys on supermarket customers and staff members to acquire additional user requirements especially important for these people.

Related to simulations

An actual implementation of the proposed simulation (with user requirement tests) with the extended HSFM is necessary to conclude whether it outperforms the other approaches in a supermarket environment.

Calibration and validation of SFM-based collision avoidance approaches is necessary, with an example given by Tang, M et al.[27] It needs to be found what optimal modelling parameters must be chosen for a supermarket environment. For example, changing the parameters A, D, FS and λ have significant impact on simulation results as well as how well user requirements are satisfied. The model parameters in the HSFM thus also need calibration. In addition to this, a better way of constraining the robot velocity needs to be found. A solution might be given by adding frictional forces (e.g. implemented by Yang, X. et al.[28]) or inertia terms (as seen in the HSFM).

Furthermore, it needs to be looked at how (computationally) inefficient this collision avoidance procedure becomes when large groups of people are in the robot's vicinity and how these inefficiencies can be overcome. They can follow from simulations, but preferably from actual implementations in robots.

As stated in the environment description, it needs to be investigated if motion prediction for shopping carts is worth the extra computational cost to the algorithm, possibly by doing simulations in combination with real-life experiments. A first step would be to add agents (shopping carts) to the simulation environment that humans can interact with and move.

Finally, it is important to look at how the problems with local minima in any SFM can be overcome. More HSFM extensions will then be necessary. It might be beneficial to add navigational forces to the HSFM as presented by Karamouzas, I et al.[29] that can aid the reactive part of the procedure with the addition of a part that is planned beforehand. It is then also important how this collision avoidance approach can be incorporated in a complete navigation solution for supermarket environments.

Conclusion

This research described the difficulties and advantages for robot collision avoidance in a supermarket environment and then formulated design requirements based on this environment as well as the users, being customers and staff members. A basic idea was presented that these user requirements can be incorporated in cost functions. Then, collision avoidance through the Dynamic Window Approach and the Social Force Model (SFM) were assessed on how well they would satisfy user requirements and how fitting they were for the supermarket environment. The Social Force Model was chosen as the most viable option. Quantitative tests for several user requirements were formulated to be used during simulation after which the standard SFM and the HSFM were simulated. Required extensions were given and then partially implemented in the standard SFM. Through simulation it was found that the extensions had a positive impact on efficiency and user requirements. Additionally, some limitations inherent to any SFM-based approach came to light regarding velocity scaling and local minima problems. All in all, the extended SFM is thought to be a step closer to be a desirable collision avoidance procedure in a supermarket. While the extended HSFM also proposed here is thought to be even better in this regard, future research must provide actual evidence.

References

[1] Robots in Retail - Examples of Real Industry Applications -. (2017, July 06). Retrieved from https://www.techemergence.com/robots-in-retail-examples

[2] Tally shelf-scanning robot adds RFID, machine learning. (2018, July 20). Retrieved from https://www.therobotreport.com/tally-robot-rfid-machine-learning

[3] Sato, T., Moro, A., Sugahara, A., Tasaki, T., Yamashita, A., & Asama, H. (2013). Spatio-temporal bird's-eye view images using multiple fish-eye cameras. Proceedings of the 2013 IEEE/SICE International Symposium on System Integration, 753–758. doi: 10.1109/SII.2013.6776674

[4] Akansel Cosgun, Henrik I. Christensen. (2018). Context-aware robot navigation using interactively built semantic maps. De Gruyter Open. Paladyn, J. Behav. Robot. 2018

[5] Thibault Kruse, Amit Pandey, Rachid Alami, Alexandra Kirsch. Human-Aware Robot Navigation: A Survey. Robotics and Autonomous Systems, Elsevier, 2013, 61 (12), pp.1726-1743.

[6] E. Hall. The hidden dimension. Anchor Books, 1966.

[7] Ghazali, A. S., Ham, J., Barakova, E. I., & Markopoulos, P. (2017). Pardon the rude robot: Social cues diminish reactance to high controlling language. 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 411–417. doi: 10.1109/ROMAN.2017.8172335

[8] M. Lohse, N. Van Berkel, E. Van Dijk, M. Joosse, D. Karreman, V. Evers. The influence of approach speed and functional noise on users' perception of a robot - IEEE Conference Publication. (2018, October 13). Retrieved from https://ieeexplore.ieee.org/abstract/document/6696573

[9] J. T. Butler and A. Agah. Psychological effects of behavior patterns of a mobile personal robot. Autonomous Robots, 10(2):185–202, 2001.

[10] K. Hayashi, M. Shiomi, T. Kanda, and N. Hagita. Friendly patrolling: A model of natural encounters. In Proceedings of Robotics: Science and Systems, Los Angeles, CA, USA, June 2011.

[11] D. F. G. M. I. H. I. N. H. S. Satake, T. Kanda. How to approach humans? strategies for social robots to initiate interaction. In HRI, ACM/IEEE, 2009.

[12] Browning, R. C., Baker, E. A., Herron, J. A., & Kram, R. (2006). Effects of obesity and sex on the energetic cost and preferred speed of walking. Journal of Applied Physiology. Retrieved from https://www.physiology.org/doi/full/10.1152/japplphysiol.00767.2005

[13] Korhonen, T., & Heliövaara, S. (2011). FDS+Evac: Modelling Pedestrian Movement in Crowds. SpringerLink, 823–826. doi: 10.1007/978-1-4419-9725-8_81

[14] Barnaud, M.-L., Morgado, N., Palluel-Germain, R., Diard, J., & Spalanzani, A. (2014, September 14). Proxemics models for human-aware navigation in robotics: Grounding interaction and personal space models in experimental data from psychology. Retrieved from https://hal.archives-ouvertes.fr/hal-01082517

[15] Wei, Z., Zhu, H., & Wang, P. (2007). An Object Recognition Method for Indoor Robot Based on Feature-Line Flows. 2007 IEEE International Conference on Automation and Logistics, 591–596. doi: 10.1109/ICAL.2007.4338633

[16] Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23–33. doi: 10.1109/100.580977

[17] Seder, M., & Petrovic, I. (2007). Dynamic window based approach to mobile robot motion control in the presence of moving obstacles. Proceedings 2007 IEEE International Conference on Robotics and Automation,

[18] Henkel, C., Bubeck, A., & Xu, W. (2016). Energy Efficient Dynamic Window Approach for Local Path Planning in Mobile Service Robotics. IFAC-PapersOnLine, 49(15), 32–37. doi: 10.1016/j.ifacol.2016.07.610

[19] Ratsamee, P., Mae, Y., Ohara, K., Takubo, T., & Arai, T. (2012). Modified social force model with face pose for human collision avoidance. 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), 215–216. doi: 10.1145/2157689.2157762

[20] Wang, P. (2016). Understanding Social-Force Model in Psychological Principles of Collective Behavior. arXiv, 1605.05146. Retrieved from https://arxiv.org/abs/1605.05146

[21] Farina, F., Fontanelli, D., Garulli, A., Giannitrapani, A., & Prattichizzo, D. (2017). Walking Ahead: The Headed Social Force Model. PLoS One, 12(1), e0169734. doi: 10.1371/journal.pone.0169734

[22] Zeng, L., & Bone, G. M. (2013). Mobile Robot Collision Avoidance in Human Environments. Int. J. Adv. Rob. Syst., 10(1), 41. doi: 10.5772/54933

[23] Martin, P., Marsh, A. (1992) Step length and frequency effects on ground reaction forces during walking, Journal of Biomechanics, Vol. 25, No. 10, 1992, pp. 1237–1239

[24] Arechavaleta G, Laumond JP, Hicheur H, Berthoz A. On the nonholonomic nature of human locomotion. Autonomous Robots. 2008;25(1–2):25–35. doi: 10.1007/s10514-007-9075-2

[25] Shelley A. Steenblock. (2010). User centred design evaluation of the grocery store environment. Graduate Theses and Dissertations. 11345. doi: 10.31274/etd-180810-2791

[26] Supermarket Shopping Cart. (2018, October 10). Retrieved from https://www.shoppingcartmart.com/supermarket-shopping-cart.html

[27] Tang, M., & Jia, H. (2011). An approach for calibration and validation of the social force pedestrian model. Proceedings 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), 2026–2031. doi: 10.1109/TMEE.2011.6199614

[28] Yang, X., Dong, H., Wang, Q., Chen, Y., & Hu, X. (2014). Guided crowd dynamics via modified social force model. Physica A, 411, 63–73. doi: 10.1016/j.physa.2014.05.068

[29] Karamouzas, I., Heil, P., van Beek, P., & Overmars, M. H. (2009). A Predictive Collision Avoidance Model for Pedestrian Simulation. SpringerLink, 41–52. doi: 10.1007/978-3-642-10347-6_4