Planning

Week 5:

~~Make individual planning~~

~~Task environment description~~

~~Finalising user requirements through proxemics~~

~~Looking at viability of Dynamic window approach and Social Force model.~~

Per approach: Try to understand it, give it a basic description, look at what user requirements are satisfied, if and how it might be extended / adapted for this application.

Decide on the best approach.

Week 6:

Find a way to simulate best approach (SFM)

Elaborate on (privacy) issues using this approach

Week 7:

Finishing simulations

Finish discussion of benefits and disadvantages for this approach

Topics for further research

Finalising report

Working on final presentation

Week 8:

22-10-2018 Final Presentation

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

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.

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 customer requirements in mind, 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. A simulation with a candidate object avoidance procedure will be done to test its working potential. Finally, advantages and disadvantages for this candidate procedure are given and topics for further research will be presented.

Task environment description of a supermarket

We will look at advantages and difficulties for robot collision avoidance in supermarkets.

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 not usable at all. Then also, how many extra ceiling cameras would be necessary and how much would that cost? Takaaki Sato et al. [BRON] [1] 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 desirable for a supermarket enterprise to invest in more cameras, when the cheaper option of only using local robot sensors 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 [BRON][2]. Since this does not directly fall into the category of collision avoidance, it will not be discussed here.

Difficulties for collision avoidance

1. Customers and staff members will be walking around supermarkets, either in groups or alone, maybe carrying a shopping cart. All these people need to be avoided in a way that is 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 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. 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)

---

[1] 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

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

inefficient some procedures become when large groups of people are in the robot's vicinity and how these inefficiencies can be overcome.

3. (Parked) shopping carts are present, which are objects that can move but not necessarily. 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.

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 prediction of movement is necessary. The location of static objects can be sent to the robot directly and path planning can be adapted accordingly.

Identifying user requirements

Proxemics

In order to find a desirable way in which robots avoid and move alongside humans, user requirements will be looked at.

The term collision avoidance in general will be used for the avoidance of all entities in a supermarket, being: humans, moving objects and stationary objects. 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. **[BRON]**[3] 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.

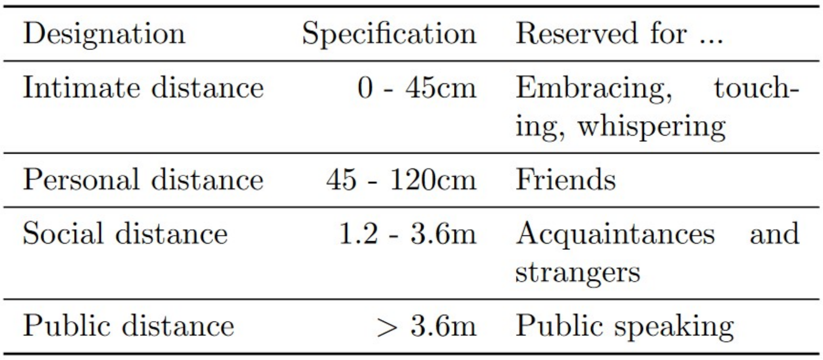
Several robot user requirements for avoiding collision with customers will be looked at now. Most of them come from surveys presented by the literature summary of Thibault, K et al. and from studies in the field of proxemics.

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.

---

[3] 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.

E. Hall **[BRON]**[4] 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 1 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 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.

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

This requirement 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) this is not necessarily a problem anymore. This requires that the robot is easily seen by surrounding people.

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

This cue should be as effective as possible in crowded environments, while also making sure no comfort is lost. No research could be found on cues that are desirable for humans.

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

Butler and Agah **[BRON]**[5] 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 **[BRON]**[6] and Satake

---

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

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

[6] 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.

[BRON][7] a speed that adapts to or resembles surrounding humans would be desirable for general movement.

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 of robots.

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

The notion of comfortable motion is expanded should also consider that robots should not be too noisy. Very little research on exact robot noise levels that are comfortable was found.

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

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.

Describing cost functions

The most straightforward way to implement these user requirements 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 is 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 figure XX.

---

[7] 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.

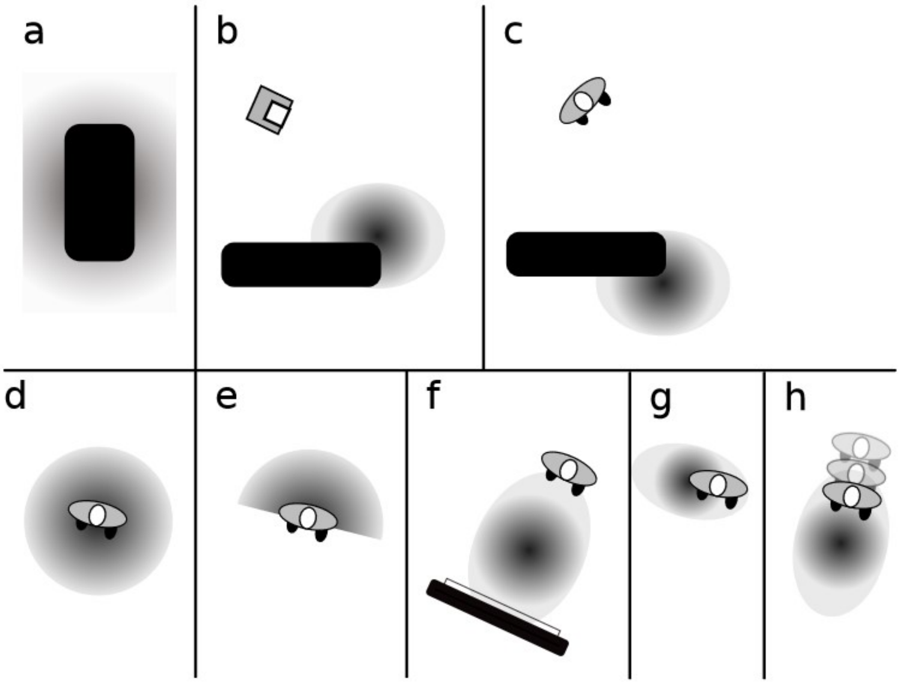


Figure 9: Visualizations of separate costmaps, thick black areas are obstacles, the square in b is a robot, human in h is moving. Areas shaded in grey have costs, meaning the robot should prefer to avoid those if possible. Shapes and sizes of cost function may vary and be context-dependent. All cost-functions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

Figure 2 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. A problem with this might 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 XXa)

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

- Object occlusion and hidden zones (seen in figure XXb and c combined)

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

- Basic comfort distance (seen in figure XXd)

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. [BRON][8] 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

---

[8] 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

validated with experimental results with an actual robot. It showed that these procedures were perceived as safe by humans and also maintained efficiency.

- Passing people on their left (seen in figure XXg)

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 for collision avoidance as it is perceived natural for humans.

- Space ahead for moving (seen in figure XXh)

In general, robots should avoid moving in this space, as it hinders people. This does require some form 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. For this application it is probably not necessary to change this parameter. 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 concept 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 (possibly) moving and static objects

The robot can make this distinction through object recognition as shown by Wei, Z et al. [BRON][9]. This approach makes use of feature-line flows and distinguishes moving from static objects by computing residual errors.

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 user requirements are satisfied is looked at. Finally, a conclusion is drawn on how this approach might need to be adapted or extended to better fit the environment.

The environment of a supermarket is for this assessment 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. During its path it will encounter static objects, moving objects and several humans standing around, walking and interacting. 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

---

[9] 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

ceiling. This environment is illustrated in the following figure.

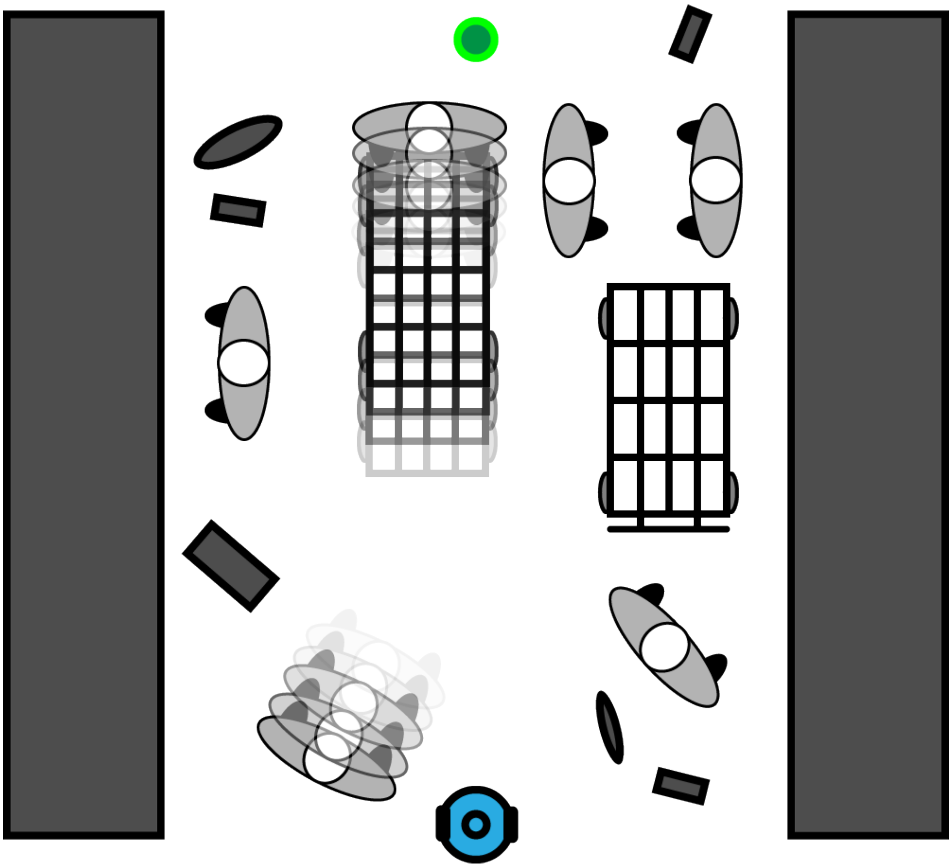


Figure 3 a schematic overview of the collision avoidance environment, the robot is represented in blue while the goal is represented in green.

Dynamic window approach

The dynamic window approach by Fox, D. et al. [BRON][10] 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:

---

[10] 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

For the x coordinate:

$$\begin{array}{rcl} x(t_n) & = & x(t_0) + \sum_{i=0}^{n-1} \left(F_x^i(t_{i+1})\right) \\ F_x^i(t) & = & \begin{cases} \frac{v_i}{\omega_i}(\sin\theta(t_i) - \sin(\theta(t_i) + \omega_i \cdot (t - t_i))),\ \omega_i \neq 0 \\ v_i \cos(\theta(t_i)) \cdot t,\ \omega_i = 0 \end{cases} \end{array}$$

And analogously for the y coordinate:

$$\begin{array}{rcl} y(t_n) & = & y(t_0) + \sum_{i=0}^{n-1} \left(F_y^i(t_{i+1})\right) \\ F_y^i(t) & = & \begin{cases} -\frac{v_i}{\omega_i}(\cos\theta(t_i) - \cos(\theta(t_i) + \omega_i \cdot (t - t_i))),\ \omega_i \neq 0 \\ v_i \sin(\theta(t_i)) \cdot t,\ \omega_i = 0 \end{cases} \end{array}$$

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 closest obstacle. Also, these velocities are restricted in that only velocities that can be reached in a short time interval 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 is 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 be extended with previously described cost functions.

[BRON][11] --> extension with motion prediction possible

[BRON][12] --> extension with a cost function

Social force model

The Social Force model as described by Ratsamee, P. et al. [BRON][13] 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 making use of 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: $\sum F = F^{goal} + F^{object} + F^{human}$. 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:

$$F^{facepose} = FS \ast e^{\frac{r_{i,R} - d_{i,R}}{s_R}} \ast \overrightarrow{\boldsymbol{v_{\iota,R}}} \ast (\lambda + (1+\lambda)\frac{1+cos(\theta)}{2}$$

In this formula, the following holds:

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

$s_{i,R}$ is the range of the force

$d_{i,R}$ is the distance

$r_{i,R}$ is the sum of their radius

$\mathbf{v}_{\mathbf{i,R}}$ 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.

---

[11] 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,

[12] 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

[13] 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

λ is a constant related to the cosine term in $F^{facepose}$

$F^{facepose}$ is summed with the other forces. A path planning for robot R and a motion prediction for human H is then derived from the differential equation $\frac{d}{dt}\vec{v} = \frac{\sum F}{m}$.

The following figure shows an overview of calculated forces for human (H) and robot (R).

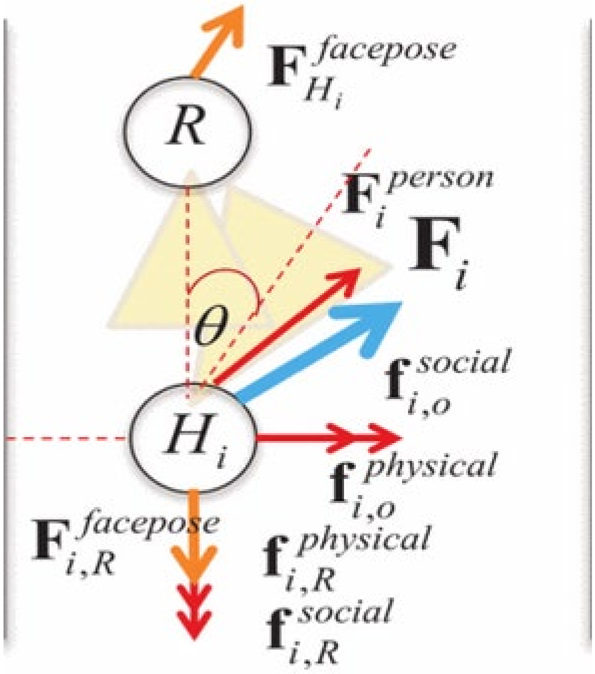


Figure 4 overview of forces from the social force model, by Ratsamee, P. et al.

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 $f^{physical}$, so user requirement 1 based on proxemics is satisfied well. It is evident that no cost function approach is needed anymore here. User requirements 2, 4 and 5 are also 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 forces.

Erratic motions as described under user requirement 5 are avoided if all the forces calculated do not change significantly in a short time span. Correct placement of sensors on the robot or the environment can prevent this.

User requirement 7 might also be partially satisfied as the path planning algorithm makes use of a person's gaze. It is thought that this makes the path planning very predictable and readable for surrounding humans. This would make the approach more easily excepted for humans in general, probably also in a variety of countries and cultures. This is validated by Source needed for social acceptance of this approach.

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 planning) and combines both to form a robot path. It is also thought that this approach does not really need an extension with cost functions that relate to HRI, because both humans and objects can be avoided in a desired way in conformity with most user requirements. 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.

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 leading to a decrease in robot reactiveness to the environment.

A solution for this might be to lump social force calculations when groups of people get larger. By using the ceiling mounted cameras, groups of people can be identified as follows. When every person in this aisle is represented by a circle and the moment more than X circles are close to each other individual social force calculations are lumped by calculating only one set of forces for the entire group.

https://ieeexplore.ieee.org/document/6199614 --> calibratie SFM

https://ieeexplore.ieee.org/document/6654008 --> relatieve velocities

https://ieeexplore.ieee.org/document/6696576 --> meer validatie

It can be concluded that the most viable approach is given by the Social Force model, because it needs little adaptation for the environment and already satisfies the most important user requirements.

Candidate procedure simulation

Conclusion

Possible privacy issues

Discussion & topics for further research