McGill University
School of Computer Science
Game Research at McGill

Navigating Social Spaces

GR@M Technical Report No. 2012-2

Bentley James Oakes   Clark Verbrugge
School of Computer Science, McGill University
Montréal, Québec, Canada, H3A 2A7
bentley.oakes@mail.mcgill.ca   clump@cs.mcgill.ca

September 17, 2012

Abstract

Behaviour of robots within a human-populated space can be disruptive, as robot motion does not necessarily conform to social norms. Typical movement models are oblivious to social expectations, and so easily violate *personal space* and other social rules, magnifying the unnatural behaviour of robot agents and causing discomfort to human occupants. This paper presents a navigation algorithm that incorporates human *proxemics* into a modified *Rapidly-exploring Random Tree* (RRT) algorithm. Our *Socially-Realistic RRT* algorithm (SRRRT) includes both a cost function based on a realistic model of human interaction distances, as well as a human motion model in order to produce movement patterns that better integrate with human social behaviour. We experiment with our algorithm in simulation, comparing it with both a naive RRT and an A* implementation in both static and dynamic movement contexts. SRRRT demonstrates quantifiably better paths in terms of social cost, while maintaining a simple and easily extensible implementation design. Inclusion of such a design in robot motion enables more socially transparent behaviour, improving the ability of humans and robots in real or virtual contexts to coexist.

1 Introduction

In the future, humans and robots will share the same personal space as they complete co-operative tasks. Therefore, it is beneficial to consider and model any psychological reactions that may occur in this space. Humans have instinctive and cultural rules that define how they experience and interact with the spaces around them, coined by Edward T. Hall as *proxemics*. For example, proxemic rules define socially acceptable distances and angles for participants in a conversation. If these constraints are violated, humans may feel uncomfortable. The motions and actions of robots must take these rules into account in order to work and collaborate with humans in harmony.

One area of developing interest is the use of socially aware navigation algorithms. These algorithms attempt to take proxemic rules into account when robots are navigating in the same environment as humans. The goal is to avoid or minimize the robot's intrusion into the personal space of humans. A major influence on these algorithms are techniques in collision avoidance, which tend to create a circular personal space to be avoided. Another main area of work has focused on robots approaching humans at a socially acceptable angle and distance to provide information or simulate conversation.

This paper presents an algorithm to combine the above two ideas to create an algorithm that can create a path that avoids the personal space of others while still appearing to be natural and human-like. This approach would aim to minimize the disruption caused by robots working beside humans by replicating social rules. Differing from past work, we aim for path-planning that is relatively socially invisible, or at least socially non-disruptive. The goal of our work is to produce a path that a human would consider respectful of personal space and similar to normal social navigation that humans accomplish everyday.

We have taken the *Randomly-exploring Rapid Tree* (RRT) algorithm from the robotics literature and modified it to be *socially realistic*. This was achieved by incorporating insights from the proxemics literature. We call this the Socially-Realistic Rapidly-exploring Random Tree algorithm (*SRRRT*), and apply it to virtual character navigation on a 2D plane. The SRRRT algorithm has been extended to have a cost function based on a model of personal space around simulation agents. This cost function can be visualized as a circular shape around an agent, with a fan shape outwards in the direction the agent is facing. This cost function allows the SRRRT algorithm to evaluate different paths around the space and avoid regions where proxemic rules make it costly to navigate. The modifications to the RRT algorithm are fairly simple, but offer large advantages in terms of social realism. It is our hope that this algorithm could therefore be implemented in real-world trials relatively quickly, as well as be supported by the large volume of RRT work already done.

We further improve the path found by the SRRRT to be more socially realistic by adding a human movement model based on human movement literature. This model incorporates an angle constraint on the nodes comprising the navigation path found by the algorithm. We argue that the smoother path created, with smoothness defined by the angle between nodes in the path, is more natural and human-like than those paths produced by A*, another path-planning algorithm.

Applications of this algorithm would be in any environment with a human or human avatar where a robot or simulated agent wishes to navigate through the space in a socially realistic manner. Specific scenarios may include co-operative office work, sharing a subway car, or as navigating background characters in a video game around the player.

Specific contributions of this work include:

- Our SRRRT algorithm is a novel path-planning algorithm which takes the personal space of human beings into account, while planning smooth paths based on a human movement model.
- In order to evaluate our system we define a social cost metric derived from the literature of proxemics and integrate it into two popular path-planning algorithms.
- Based on our model we gather simulation data evaluating and comparing SRRRT and A* with path smoothing and social cost metrics. A short discussion of these results offers insights into further algorithmic improvements.

The next section of this paper provides related work. Section 3 describes the algorithms and models created. Section 4 describes the simulator as well as a discussion of the collected data. We conclude with Section 5.

2 Related Work

2.1 Human Movement and Interaction

Perhaps the most well known feature of human interaction relevant to navigation concerns is the extent and nature of *personal space.* Human interaction space has significant complexity, with important social implications. Hall, for instance, divides human interaction distance into various zones of familiarity along with the acceptable and unacceptable actions that may occur therein [5]. For example, the intimate zone may only be entered by partners, and may only allow gestures like soft touching. Strangers entering this zone would make the person very uncomfortable. Another useful observation by Hall is that these distances change drastically by cultures. The numbers he offers are valid for North America, but may not be so for Japan or the Middle East.

Human beings, like robots, also have physical constraints on the paths they take through a space. This suggest that a navigation algorithm may also take properties of human movement into account in order to be socially realistic. In particular, human movement has a necessary inertia component which produces curved paths when plotted [2]. This is sharply in contrast to the A* path-planning algorithm commonly employed in simulation, which tends to produce piecewise linear paths. Smoothing reduces such artifacts, but this result with others seems to suggest that human motion would be best represented as non-holonomic, with the motion depending on the orientation of the trunk of the individual [1].

2.2 Human-Robot Proxemics

There is a growing body of literature on interactions between humans and robots that use increasingly realistic approximations of human behaviour. A framework has been proposed by Lam et al. that defines rules and fields to help humans and robots co-exist in a space [8]. The six rules define the relationship of robots to other robots and to humans, such as priority levels and the rule to leave a space when intruding. The framework was used successfully in an experiment to manage interactions between multiple robots and humans.

2.2.1 Psychological and Emotional Factors

Human psychology and robot motion can influence how the robot's motion is perceived by surrounding humans. Saerbeck and Bartneck elicited emotional states that humans placed on robots due to the robot's motion [15]. Strong factors to influence the perceived emotion was acceleration of the robot and the curvature of their path. Robots have also been used to interact with children with autism spectrum disorder in a socially assistive manner [4]. These robots provided motion and aural cues to the children to imitate human emotional states, which the children responded to by interacting or disengaging with the robot. The size and shape of the robot may also be a factor in how humans interact with the robot. In a study by van Oosterhout and Visser, a smaller robot received more interaction by younger participants, while adults tended to prefer the larger robot [20].

The physical distance that humans are comfortable with robots is also crucially important for study and modelling. When asked to approach or be approached by a robot in trials, the human's personality and familiarity with robots modified the distance they stood at [19]. For example, participants who were rated 'proactive' by Walters et al. tended to leave a larger distance between themselves and the robot [21]. It should also be noted that parcipiants tended to leave the same amount as Hall predicted for human-human interaction, although some participants moved closer to the robot then expected. This may signal that the participant did not feel that the robot was a social equal to be offered the same personal space.

2.2.2 Approaching Humans

In an effort to make robots more useful as information guides or assistants, effort has been made to identify and approach persons receptive to a robot's assistance [16]. This work also includes a model that determines proper placement of an informational robot within a group [22], robot manipulation of conversational formations to steer group movement [7], and how to use human pose estimation to identify the correct placement of the robot for interaction [18]. The angle of approach has also been reported to be important to the comfort of humans, as well as the criterion that the robot should be visible to the human as much as possible [3]. The use of face detection and tracking as well as verbal questioning for willingness to interact can also help robots co-operate to perform tasks with a human [17].

Rios-Martinez et al. have created a simulated robot which uses a *Risk-RRT* algorithm to navigate around virtual agents having a conversation [14]. This robot is also able to recognize formations in a conversation between simulated agents and approach the group in a natural way. Our approach shares ideas with this work, but adds the concept of a human movement model to the RRT algorithm in order to find a smooth path. A Dynamic Window Approach has also been used in order to plan and approach a human without pre-planning [6]. This approach uses a continuous model to navigate the robot into a trajectory that approaches humans slightly from the side to avoid threatening behaviour.

2.2.3 Passing and Pathing

When a robot and a human are heading towards each other, safety and comfort considerations dictate that the human be given ample space [23]. Experimental feedback can also offer rules for preferred speeds and at what distance the robot should begin to move to the side [11].

Psychological and geometrical proxemic rules can produce emergent behaviour, as found during an evacuation scenario in a conference hall [13]. An approximation of personal space is considered in a local potential field model in order for agents to physically push away agents in their path, potentially leading to agents falling down and becoming obstacles. This use of proxemic rules only considers an agent's personal space to extend the agent's physical space and not as a region to be avoided as in our work. It is interesting to note that in stressful situations, human beings may disregard proxemic rules out of concern for their own safety, possibly endangering robots that cannot react to the changing situation. Future work in our model will focus on how situational or individual changes in human perceptions of personal space affect appropriate path-planning.

An iterative approach to path around humans and reach the destination has been implemented by Pandey and Alami [12]. This work uses proxemic rules and task specific rules in order to produce a smooth path through a space while performing tasks such as guiding a human follower. Our work builds on this by providing an explicit social cost function in order to measure the cost of a path. A movement model for the agent is also integrated into the SRRRT to produce a socially realistic path.

2.3 Queueing

An interesting application of social properties in robot movement was demonstrated by Nakauchi and Simmons. In their work a robot has been developed that uses the concepts of personal space and queueing to wait in line to buy coffee [10]. Their design uses experimental data in order to construct an average personal space area shape for people waiting in line. The robot is then able to manoeuvre itself to the end of the line and queue naturally until reaching the coffee vendor.

3 Algorithms and Models Used

When deciding which algorithms to modify to explore navigating around personal space, we chose algorithms already known to the simulation and robotics community. This was done to build upon

Classification	Radius	Activities
Intimate	0.45m	Embracing, touching or whispering
Personal	1.2m	Interactions between family members or close friends
Social	3.6m	Interactions with acquaintances
Public	7.6m	Public speaking

Table 1: Hall's classification of distances

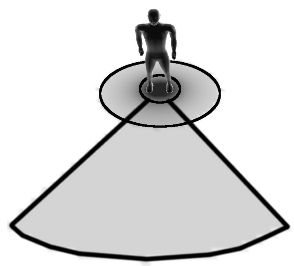


Figure 1: The personal space model around an agent

the large body of literature available, as well to encourage existing systems to implement benefits found here and in future work. As the RRT and A* algorithms are widely known by the robotics community, we assume the reader is familiar with them for the sake of brevity.

3.1 Proxemic Values

To create our model of personal space, we relied on the well-known proxemic values found in Hall's work [5]. His work defined four separate distance classification in which various actions are socially acceptable. These are briefly summarized in Table 1. For example, any interaction between persons with less than 1.2m of space between them but more than 0.45m would be classified as occurring in 'personal space'. It should be noted that these are imprecise measurements, and are based upon Hall's own culture of North America. Hall warns that various cultures may have a different set of limits or activities than seen here.

3.1.1 Social Cost Algorithm

The proxemic distances above are used to create a social cost region around an agent. Figure 1 shows the three regions of the model: the inner circle, the outer circle, and the fan shape. The boundaries of the circle are defined by the values above. The intimate distance is used for the inner circle, and the personal distance for the outer circle. The social distance defines the outer boundary of the fan shape, which has an arc of 60 degrees.

The cost algorithm works by evaluating a point p in space against all agents A in the space. For each agent $a \in A$ where a is not the navigating agent, d is set to the distance that p lies away from a. p is then classified according to Hall's rules based on the value of d. For example, p will be classified as being within the intimate distance if $d < dist_{intim}$, where $dist_{intim} = 0.5m$ in our paper.

Based on these classifications, the distance and angle from a is used to sum the cost. For instance, if p is within the intimate distance, the social cost of p is calculated by multiplying a constant k_{intim} by d_{intim}/d. In our experiments, k_{intim} was set to 500 based upon experimental results. Future work should be undertaken to assign units and real-world values to these constants. If $d_{intim} < d < d_{personal}$, the cost is $k_{personal} * d_{personal}/d$ plus an extra $cost_{fan}$ if p is within the fan area as defined above. This ensures that the area directly in front of the agent is penalized higher than the areas to the back and side. The last case is where $d_{personal} < d < d_{social}$, in which case the social cost is $cost_{fan}$.

Note that the social cost metric used below in our experiments is the social cost evaluated for each frame of the simulation, with p as the navigating agent's position.

3.2 A* Extension

The SRRRT algorithm is the focus of our paper. However, A* is a widely used path-planning algorithm in the field of robotic path-finding and is therefore useful to examine for comparison. Benefits of A* include the use of a heuristic function, which attempts to estimate how much farther the goal is from a particular node. In some cases, this heuristic can decrease the time taken to find a path dramatically. Here we present modifications to A* to accomplish similar goals to the SRRRT algorithm.

As A* is a well-known algorithm, we will not present the basic operation. However there are two points of interest that have been modified in this paper. The first is to change the known cost for each A* node to include the social cost for each node as well as the distance travelled from the root node. The social cost is described as above, with the position of the node as p. This allows the algorithm to avoid routes that pass through a region with high social cost.

The A* algorithm was also modified to instantly re-path if the navigating agent was experiencing a high social cost at any moment. This increased the sensitivity of the navigating character to moving agents and outdated information. Instead of continuing to move into another's personal space, a new path to the goal was computed that would, with high likelihood, avoid the agent intruded upon. However, the threshold for recalculation was arbitrarily set. In future work, an improved approach would be to dynamically set this threshold based on the expected social cost of the surrounding environment. An example of this may be a subway environment or other highly congested context, where some social cost is to be expected no matter the position due to such close proximity to other persons.

3.3 Socially Realistic Rapidly-exploring Trees

Rapidly-exploring Random Trees offer the algorithmic base for the Socially-Realistic RRT described in this paper. The RRT algorithm is a probabilistic roadmap creator that starts from a root node and iteratively picks target nodes to extend the tree towards. These target nodes are either random points in the space or the goal if there is one. As a full discussion of RRTs are beyond the scope of this paper, we point to the comprehensive guide provided by Lavalle and Kuffner for further details [9].

We have modified two functions of the RRT algorithm to create the SRRRT algorithm. These are the functions to choose the next node to expand the tree from, and the function that performs the expansion of the tree.

3.3.1 Choosing Node to Expand

In the RRT algorithm, typically the closest node to the target node is chosen to expand the tree from. Therefore, a distance heuristic is applied to all nodes in the tree, and the one with the lowest cost is selected. We propose that this heuristic be changed in order to also consider the social cost of the node. Equation 1 shows the proposed heuristic. The heuristic $h(n)$ is comprised of two parts: the Euclidean distance d to the target node and a cost term. This cost term is comprised of the social cost of the node's position, added to the cost of the node's parent. The parent's cost p_{cost} is added to discourage paths that lead through socially costly areas. Finally, this sum is multiplied by some value c. This weight determines whether the distance term dominates or the social cost term. Our trials found that a reasonable value for c was 2 based upon observation of tree branching and personal space avoidance. Future experiments will determine appropriate metrics to evaluate the effect of c.

$$h(n) = d + c * (socialCost(n) + p_{cost}) \tag{1}$$

3.3.2 Extending Towards Target Node

At the extension stage, the algorithm has picked a target node t and the node from which to extend the tree e. Typically, the RRT algorithm creates a new node on the line from e to t, depending on the movement model implemented. It is also desirable to check for obstacles that prevent the creation of the new node at this stage. In our work, the SRRRT algorithm checks for obstacles as well as selects the best position for the new node based on the social cost.

In our algorithm, the first step is to determine the vector from e to t. Then, this vector is slightly rotated in order to sample potential new node positions, which lie a fixed distance along the vector. These node positions are evaluated according to three criteria. First, these nodes may not lie within an obstacle, Secondly, they must be within the acceptable turning angle as defined by the movement model defined in our algorithm. Finally, the social cost for each node position is found.

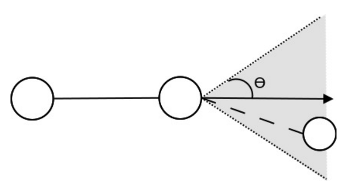


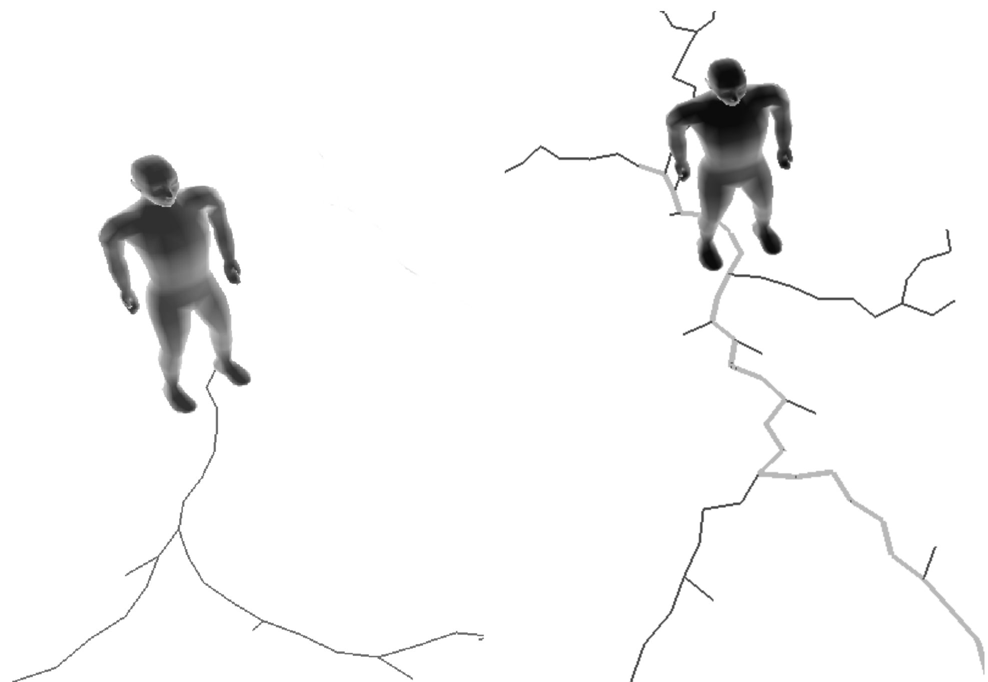
Figure 2: Movement model constraint

The second criteria is demonstrated in Figure 2. The node in the middle of the figure is e, while the node to the left is e's parent node. From these two nodes another vector can be determined. Our movement model only permits a new node to be created within a certain angle of this vector, which is the solid line to the right in the figure. If a potential node does not lie within this arc, it is discarded. In this way, the movement model directly controls the angle between nodes in the path, which we argue in our results is a important criteria for natural paths.

The third criteria is social cost, which can be easily calculated for each potential new node. At the end of the sampling process, the node with the lowest social cost is selected, and added to the tree.

3.3.3 Human Movement Model

It is our contention that a human-like movement model is critical for socially realistic navigation. Our movement model will produce paths that are smooth and natural, giving robots or simulated agents socially realistic navigation. A strong advantage to RRTs is that a movement model can easily be implemented in the above extension step by simply enforcing the constraints illustrated in Figure 2. As a result, paths produced using this movement model have much smoother turns. An example is shown in Figure 3a based on $\theta = 30°$, which can be compared to the original RRT path shown in Figure 3b. The $30°$ criteria was made experimentally to produce smooth paths without overly restricting the navigation tree created. It is our hope that further study of the human motion literature will inform our movement model, and lead to increasingly realistic motion.



(a) RRT path with movement model (b) RRT path with no movement model

4 Simulator and Results

The experiments performed were designed to simulate dynamic social situations in a crowded space, with zero, one, or many 'obstacle agents' who may be motionless or moving. For simplicity, these agents move by picking a random destination, rotating to face it, and then move along a straight line to their target. These obstacle agents provide the context with which to test the algorithms under various conditions. It was decided to use the experiment parameter of zero obstacle agents to provide a useful data baseline for error detection. Experiments of one agent were designed to capture the behaviour where the navigating algorithm faces a minimum of personal space, while experiments with five agents in the space are to provide an dense, cramped environment to navigate through. Future experiments could use data from human studies to determine realistic densities for scenarios such as a public square, metro car, or mall.

The experiments consisted of two parts. First, the A* algorithm was tested for all combinations of number of agents (0, 1, 5), agent movement (static, dynamic), and the A* heuristic (distance to target, distance to target squared). For the second experiment, the SRRRT algorithm was tested on the same combinations of number of agents and agent movement, but with the addition of the human movement model (used, not used). These combinations ensure that the performance of these algorithms is captured over a wide range of parameters. Each particular trial was repeated 10 times with a different random seed, and the results were averaged together. These experiments were carried out in a Java real-time simulation environment on a Intel Core i7-3820 CPU computer with 16 GB of RAM under Ubuntu 12.04.

The algorithms were tested in an environment consisting of a room with two main entrances and two exits, as seen in Figure 3. The navigating agent begun the experiment on one side of the room, and their goal was to manoeuvre to the square on the other side while avoiding the obstacle agents inside. This space was quite advantageous in testing the algorithm's ability to navigate around a constrained environment with dense, overlapping personal spaces. The algorithm could choose to enter through one of the entrances in order to intelligently minimize the social cost of the path. Our future work intends to remake this space into a subway car in order to study social dynamics under extremely crowded and stressful social situations.

The simulator created provides a real-time, efficient framework in order to visualize, measure, and compare socially realistic navigation algorithms. Along with standard timing and logging capabilities, this simulator offers a unique social cost visualization as shown in Figure 3. The upper agent is about to begin navigating through the room, which contains a stationary agent facing left. The personal space for the stationary agent is shown clearly as a fan shape connected with a circle, and the visualization is produced with the social cost algorithm defined above. Even as multiple agents move throughout the space, the social cost visualization offers an indication of socially expensive regions in real-time.

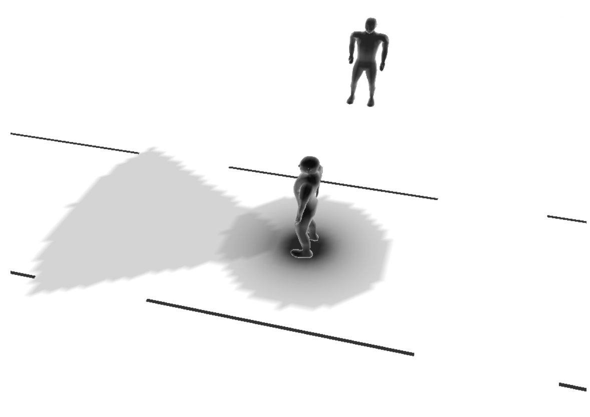


Figure 3: Room layout with entrances/exits

Algorithm	Max Angle	Avg. Angle	Num. of Sharp Angles
RRT With Movement Model	74.83	16.02	1.02
RRT Without Movement Model	93.89	26.90	27.10
A*	51.75	7.62	23.21

Table 2: Smoothness metric results

Mobility of Obstacle Agents	Num. of Agents	Social Cost
SRRRT		
Mobile	5	132.51
	1	32.41
Immobile	5	4.00
	1	0.0
A*		
Mobile	5	2194.8
	1	207.70
Immobile	5	0.0
	1	0.0

Table 3: Social cost by obstacle agent parameters

4.1 Smoothness Metrics

Table 2 shows various angle measurements taken on paths found by the navigating algorithms. To obtain these measurements, we examined each set of three neighbouring nodes along the path. Given these three nodes it is then trivial to find the angle at which the path is bending. We report the maximum angle found at any node in the path, the average angle for all nodes in the path, as well as the number of 'sharp angles'. These sharp angles are the number of angles found in the path that exceed the angle criteria set out in our human movement model. We argue that these three metrics define the smoothness of the path. It is apparent that when the SRRRT algorithm is used with the human movement model, the largest angle, average angle, and number of sharp angles decreases compared with the normal RRT algorithm. We submit that this result is desirable as a smooth path is human-like and more socially realistic than a sharply-turning path. Future work will further expand upon this movement model, as well as validate the data against the human movement literature.

It is also informative to compare these results to those obtained with the A* algorithm. While the A* path tended to have a smaller max angle and smaller average angle, the number of sharp angles was much larger than for the SRRRT algorithm. One likely reason for these results is that the A* algorithm tends to favour straight line paths with sharp corners, as in Figure 4.

4.2 Social Cost

Table 3 shows the results obtained from grouping the experiments based on the number and mobility of the obstacle agents within simulation space. The social cost metric defined above gives insight into the navigating algorithms. One expected observation is that a higher number of agents, with more mobility, causes more social cost.

However, an interesting result is the difference between the data for the A* and SRRRT algorithms. With five moving agents, the average social cost for a agent navigating using RRT was 132.51. Navigating around one moving agent produces about five times less social cost, as to be expected. On the other hand, it is curious how the SRRRT algorithm did experience some social cost while navigating around five immobile agents. Further work is needed to identify the cause of this. For one immobile agent, or for the trivial case where no obstacle agent was present, the social cost was zero.

Examining the social cost for the A* algorithm leads to a very surprising result. While the immobile agents produced no social cost in the navigating agent at all, the mobile agents produced a much larger amount. For example, an agent using A* to navigate around one mobile agent would experience around 1.5 times the social cost of if it had used the SRRRT algorithm to path around five mobile agents. The social cost difference between one mobile agent and five mobile agents for the two algorithms is also very interesting. While the SRRRT algorithm experiences around five times more social cost for five times the agents, the A* algorithm experiences over ten times the social cost as for one agent. As a direct comparison, the A* social cost is over 16 times the equivalent SRRRT cost for five mobile agents.

These results could arise from how the A* algorithm tends to produce paths that closely hug the personal space of other agents, as in Figure 4. Any movement by the obstacle agents may lead to a rapid increase in social cost as the navigating agent finds itself intruding upon personal space. We suggest that naive A* is not applicable to navigate around personal space. SRRRT incurs a much smaller social cost when used to path through a dense social environment, and should therefore be chosen in these situations.

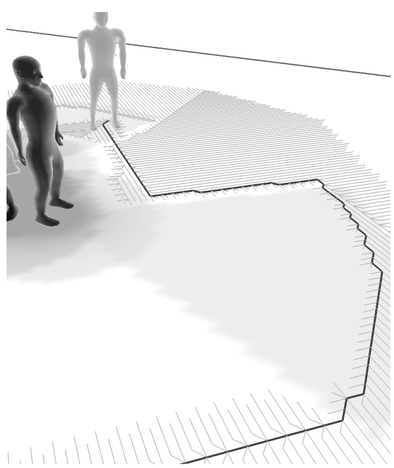


Figure 4: A* algorithm producing straight line paths

5 Conclusions and Future Work

Socially realistic models for robot navigation are a useful and interesting direction for better and more natural integration of robot movements in human society. In this paper, we have presented the Socially-Realistic Rapidly-exploring Random Tree algorithm to navigate through an environment while maintaining social realism. We have incorporated rules and models from the proxemic literature in order to evaluate social cost in a space, as well as created a human movement model to produce smooth paths. These incorporations are both straightforward and lead to clearly improved behaviour, as seen in the comparison of the SRRRT algorithm to an implementation of A*.

There are a number of directions for future work in our approach. A more detailed model of human social behaviour would allow for even better integration, and we are currently working on algorithm variations that will better accommodate heterogeneous social groupings (mixtures of intimate groups and strangers), the potential for eye-contact, the emotional states of individuals, cultural differences, and other social factors. Including actual, observed behaviours, such as partly addressed in [10], would enable our design to adapt to current and changing circumstances; the impact of highly congested contexts, for instance, would be particularly interesting, as they involve substantial changes in human behaviours. A current, direct application of our work is in the context of video games where computer controlled non-player avatars share many of the same navigation concerns as robots. To ensure the player's continued sense of immersion, it is critical that they are not disrupted by the socially awkward behaviour of virtual agents.

References

[1] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz. The nonholonomic nature of human locomotion: a modeling study. In *Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on*, pages 158 –163, feb. 2006.

[2] D. Brogan and N. Johnson. Realistic human walking paths. In *Computer Animation and Social Agents, 2003. 16th International Conference on*, pages 94 – 101, may 2003.

[3] K. Dautenhahn, M. Walters, S. Woods, K. Koay, C. Nehaniv, E. Sisbot, R. Alami, and T. Simeon. How may i serve you? a robot companion approaching a seated person in a helping context. In *Proceedings of ACM SIGCHI/SIGART 2nd Conference on Human Robot Interaction (HRI 06)*, 2006.

[4] D. Feil-Seifer and M. Mataric. Using proxemics to evaluate human-robot interaction. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 143 –144, march 2010.

[5] E. T. Hall, R. L. Birdwhistell, B. Bock, P. Bohannan, J. Diebold, A. Richard, M. Durbin, M. S. Edmonson, J. L. Fischer, D. Hymes, S. T. Kimball, W. L. Barre, S. J. Lynch, Frank, J. E. McClellan, D. S. Marshall, G. B. Milner, H. B. Sarles, G. L. Trager, and A. P. Vayda. Proxemics [and comments and replies]. *Current Anthropology*, 9(2/3):pp. 83–108, 1968.

[6] J. Kessler, C. Schroeter, and H.-M. Gross. Approaching a person in a socially acceptable manner using a fast marching planner. In S. Jeschke, H. Liu, and D. Schilberg, editors, *Intelligent Robotics and Applications*, volume 7102 of *Lecture Notes in Computer Science*, pages 368–377. Springer Berlin / Heidelberg, 2011.

[7] H. Kuzuoka, Y. Suzuki, J. Yamashita, and K. Yamazaki. Reconfiguring spatial formation arrangement by robot body orientation. In *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference on*, pages 285 –292, march 2010.

[8] C.-P. Lam, C.-T. Chou, C.-F. Chang, and L.-C. Fu. Human-centered robot navigation - toward a harmoniously coexisting multi-human and multi-robot environment. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 1813 –1818, oct. 2010.

[9] S. Lavalle and J. Kuffner. Rapidly-exploring random trees: Progress and prospects. In *Algorithmic and Computational Robotics: New Directions*, pages 293–308, 2000.

[10] Y. Nakauchi and R. Simmons. A social robot that stands in line. In *Intelligent Robots and Systems, 2000. (IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, volume 1, pages 357 –364 vol.1, 2000.

[11] E. Pacchierotti, H. Christensen, and P. Jensfelt. Human-robot embodied interaction in hallway settings: a pilot user study. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 164 – 171, aug. 2005.

[12] A. K. Pandey and R. Alami. A framework for adapting social conventions in a mobile robot motion in human-centered environment. In *ICAR 2009*, 2009.

[13] N. Pelechano, J. Allbeck, and N. Badler. Controlling individual agents in high-density crowd simulation. In D. Metaxas and J. Popvic, editors, *Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, 2007.

[14] J. Rios-Martinez, A. Spalanzani, and C. Laugier. Understanding human interaction for probabilistic autonomous navigation using risk-rrt approach. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2014 –2019, sept. 2011.

[15] M. Saerbeck and C. Bartneck. Attribution of affect to robot motion. In *Proceedings of the 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI2010), Osaka.*, 2010.

[16] S. Satake, T. Kanda, D. F. Glas, M. Imai, H. Ishiguro, and N. Hagita. How to approach humans?: strategies for social robots to initiate interaction. In *Proceedings of the 4th ACM/IEEE international conference on Human robot interaction*, HRI '09, pages 109–116, New York, NY, USA, 2009. ACM.

[17] E. Sisbot, A. Clodic, L. Marin U., M. Fontmarty, L. Brethes, and R. Alami. Implementing a human-aware robot system. In *Robot and Human Interactive Communication, 2006. ROMAN 2006. The 15th IEEE International Symposium on*, pages 727 –732, sept. 2006.

[18] M. Svenstrup, S. Tranberg, H. Andersen, and T. Bak. Pose estimation and adaptive robot behaviour for human-robot interaction. In *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, pages 3571 –3576, may 2009.

[19] L. Takayama and C. Pantofaru. Influences on proxemic behaviors in human-robot interaction. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 5495 –5502, oct. 2009.

[20] T. van Oosterhout and A. Visser. A visual method for robot proxemics measurements. In *Proceedings of Metrics for Human/Robot Interaction*, pages 61–68, 2008.

[21] M. Walters, K. Dautenhahn, R. te Boekhorst, K. L. Koay, C. Kaouri, S. Woods, C. Nehaniv, D. Lee, and I. Werry. The influence of subjects' personality traits on personal spatial zones in a human-robot interaction experiment. In *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pages 347 – 352, aug. 2005.

[22] F. Yamaoka, T. Kanda, H. Ishiguro, and N. Hagita. How close? model of proximity control for information-presenting robots. In *Human-Robot Interaction (HRI), 2008 3rd ACM/IEEE International Conference on*, pages 137 –144, march 2008.

[23] M. Yoda and Y. Shiota. Analysis of human avoidance motion for application to robot. In *Robot and Human Communication, 1996., 5th IEEE International Workshop on*, pages 65 –70, nov 1996.