

COMPANION: A Constraint-Optimizing Method for Person-Acceptable Navigation

Rachel Kirby, Reid Simmons, and Jodi Forlizzi

Abstract—This paper introduces the COMPANION framework: a Constraint-Optimizing Method for Person-Acceptable Navigation. In this framework, human social conventions, such as personal space and tending to one side of hallways, are represented as constraints on the robot's navigation. These constraints are accounted for at the global planning level. In this paper, we present the rationale for, and implementation of, this framework, and we describe the experiments we have run in simulation to verify that the method produces human-like behavior in a mobile robot. Our approach is novel in that it can express an arbitrary number of social conventions and explicitly accounts for these conventions in the planning phase.

I. INTRODUCTION

Mobile robots that encounter people on a regular basis must react to them in some way. Traditional robot control algorithms for path planning and obstacle avoidance treat all unexpected sensor readings identically: as objects that must be avoided. For a mobile robot that operates near and with people, however, these traditional methods may not follow human social norms. Even a simple convention, such as passing oncoming people in a hallway on the right, might not be honored by a naïve obstacle avoidance algorithm. However, people generally perceive robots—particularly assistive robots, which must move around people—as human-like, even when the robots are non-anthropomorphic [1]. When such robots behave counter to what is socially expected, breakdowns in human-robot interaction occur (see e.g. [2]). While some algorithms have been developed to produce social behavior around people, they typically do so in a local, reactive way, which may not be human-like.

In contrast, we have developed a framework for navigational algorithms for human-robot physical social interaction, such as navigating through crowds or standing in line. We call our framework COMPANION: a Constraint-Optimizing Method for Person-Acceptable Navigation. COMPANION is a generalized framework for representing social conventions as components of a constraint optimization problem, which is used for path planning and navigation. Social conventions, such as personal space and tending to the right, are described as mathematical cost functions that can be used by a heuristic path planner.

The remainder of this paper addresses related work and describes the COMPANION framework and implementation.

R. Kirby and R. Simmons are with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA {rachelg, reids}@cs.cmu.edu

J. Forlizzi is with the Human-Computer Interaction Institute and the School of Design, Carnegie Mellon University, Pittsburgh, PA 15213, USA forlizzi@cs.cmu.edu

We present results from our simulation experiments that demonstrate that our approach produces socially acceptable behaviors. Finally, we present our conclusions and future work.

II. RELATED WORK

Several methods have been developed to allow robots to navigate around people during specific, typically non-generalizable, tasks. Some of these tasks include tending toward the right side of a hallway, particularly when passing people [3], [4], standing in line [5], and approaching people to join conversational groups [6]. Museum tour guide robots are often given the capability to detect and attempt to deal with people who are blocking their paths on a case-by-case basis (e.g. [7]). Algorithms developed for the robot Grace allowed it to navigate a conference hall, ride an elevator, and stand in line to register for a conference, each as an individual behavioral module [8]. In contrast, we present a generalized framework for integrating multiple social conventions into a robot's behavior, thus producing natural and understandable robot movement without programming individual behaviors for every possible scenario.

Several groups have begun to address questions relating to planning complete paths around people, rather than relying on solely reactive behaviors. Shi and colleagues have developed a method for a robot to change its velocity near people [9]. While this method begins to address ideas of planning around people, it does not directly consider social conventions. In contrast, the Human-Aware Motion Planner (HAMP) [10] considers the safety and reliability of the robot's movement as well as "human comfort," which attempts to keep the robot in front of people and visible at all times. However, the paths that the planner generates may be very unnatural due to its attempts to stay visible to people.

Our approach is unique in that it is capable of expressing an arbitrary number of social conventions, it explicitly accounts for these conventions in the planning phase, and it is intended to produce sociable, human-like paths.

III. DESIGN

Human social conventions are *tendencies*, rather than strict rules. For example, people do not *always* maintain a specific buffer space around each other (i.e. personal space). Rather, social conventions are flexible and are followed opportunistically. By modeling social conventions as costs to a path planner, the robot is able to generate paths that *tend* to observe human social conventions, similar to the way that people tend to follow them.

A. Global Planner

We believe that, for a robot to navigate in a human-like manner, it must account for human social conventions not just at a reactive level, but at a global planning level. To understand why, consider walking down an office hallway and encountering someone walking toward you. In the United States, social convention dictates that you should move to the right side of the hallway; the other person will do similarly, thus allowing you to pass each other without incident. Consider, however, if your goal is an office down an intersecting hallway to your left. You may now choose to walk across the hallway in front of the oncoming person, effectively passing them on the left of the corridor. Doing so is not anti-social; rather, it represents a personal trade-off between social conventions and what we might call “task conventions”—namely, the desire to reach your goal in as little time as possible.

Thus, to produce human-like navigation in a mobile robot, the robot must use a global planner that is capable of optimizing among multiple constraints. For this, we use the heuristic planner A* with a cost function that accounts for both task and social conventions. We discuss details of our planner implementation in Section IV-A.

B. Constraints and Objective Functions

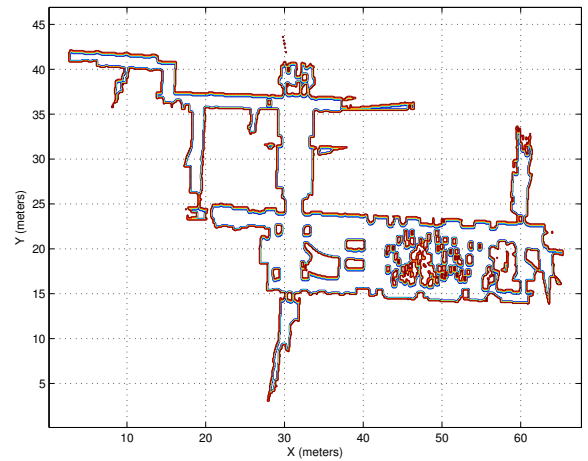
Constraints and objective functions are related mathematical concepts. *Constraints* limit the allowable range of a variable (e.g. “ x is constrained to be less than 100”). Constraints may be *hard* or *soft*; hard constraints provide an absolute limit, whereas soft constraints allow a variable to pass a given limit, but at an associated cost. A cost or *objective function* is a mathematical function that can be *optimized*—that is, maximized or minimized. Soft constraints and objective functions can be mathematically transformed into each other [11]; in the following, we will use the terms interchangeably.

We have identified the following constraints as particularly important for social behavior in hallway situations. The first two, minimizing distance and obstacle avoidance, relate to the *task* of traveling to a goal, whereas the remaining constraints of person avoidance, default velocity, and inertia all relate to the *social* aspects of traveling around people.

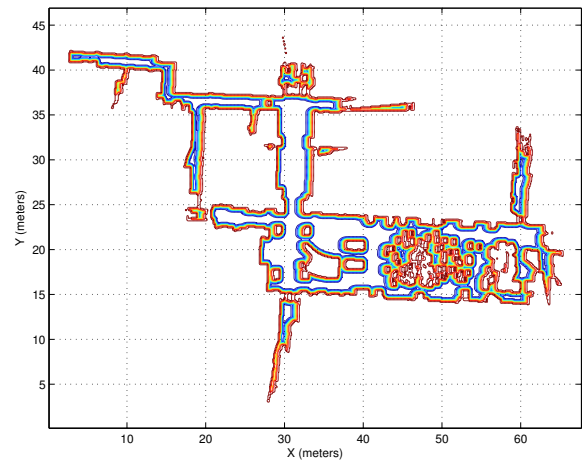
1) *Minimize Distance*: When walking to some goal, people tend to choose paths that minimize their energy expenditure [12], [13], taking shortcuts when available [14]. That is, at some level, people plan to take the shortest possible path to their destination. Thus, one part of the robot’s objective function should be to minimize the overall path length.

2) *Obstacle Avoidance*: Obstacle avoidance is comprised of two components: a hard constraint against trying to pass through obstacles, and an optimization function to avoid traveling too close to things.

a) *Hard Avoidance*: The robot has a hard constraint against collisions with static obstacles in the world. This is a standard constraint in robot path-planning algorithms.



(a) Obstacle buffer cost region for a given map and robot velocity of 0.5 m/s at $\theta = \pi/2$ (i.e. upward).



(b) Obstacle buffer cost region for a given map and robot velocity of 1.5 m/s at $\theta = 0$ (i.e. to the right).

Fig. 1. Obstacle buffer cost regions for two robot velocities and directions, shown as contour lines. For a faster speed (1(b)), the cost regions cover a larger portion of the map. Furthermore, the robot’s direction of travel influences the width of the cost region, so that the robot incurs a higher cost when driving directly toward an obstacle rather than along side one.

b) *Buffer Space*: The robot also attempts to keep a buffer space around static obstacles. It does this by incurring a cost when it approaches obstacles. The cost varies according to the robot’s speed and direction. The buffer is computed in advance, according to obstacles known on the map. While this cost is primarily for the robot’s safety, it also mimics human behavior. Figure 1 shows the cost regions for a given map and various speeds and directions of the robot.

3) *Person Avoidance*: As with static obstacle avoidance, the robot’s person avoidance is composed of both a hard constraint and several objective functions.

a) *Hard Avoidance*: While certain circumstances may arise in which a robot may be allowed to contact a person, the robot must never plan a path that would attempt to drive *through* a person. In planning, this can be achieved by rejecting robot actions that would cause the robot’s path and the person’s path to intersect.

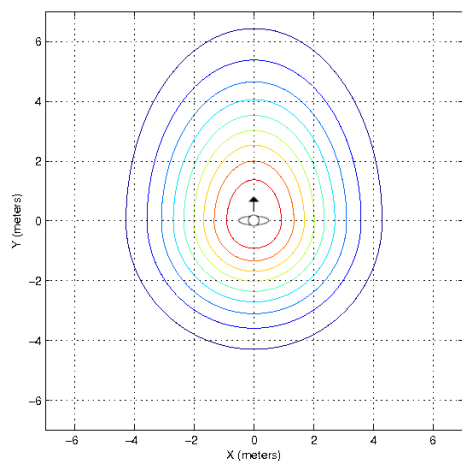


Fig. 2. Personal space cost function for a person moving along the positive Y-axis, with a relative velocity of 1 m/s toward the robot. Greatest cost is centered over the person at point (0,0). Cost is computed by composing two Gaussian functions. Below the X-axis, the function is a symmetric Gaussian with $\sigma_x = \sigma_y = 3v$ where v is the relative velocity between the person and robot; above the X-axis, $\sigma_y = 1.5\sigma_x$.

b) Personal Space: Personal space, or more broadly *proxemics*, is the “bubble” of space that people attempt to keep around themselves and others [15]–[17]. The shape of personal space is asymmetric—greatest to the front of a person—but its exact size is not constant and differs across cultures and familiarity groups [18]. Furthermore, the size of personal space can change based on walking speed as well as other factors [19].

Some attempts have been made to measure the robotic equivalent of personal space [5], [20]. In general, people tend to keep a similar space around the robot as if it were human, so the constraint to our planner should also respect human-like tendencies.

The personal space constraint can be modeled as two halves of 2D Gaussian functions: an elliptical function to the front of the person, and a symmetrical function behind. The size of the function is dependent on the relative velocity between the person and the robot. Figure 2 shows the cost function for a person moving along the positive Y-axis, with a relative velocity of 1 m/s toward the robot. This cost function was designed to roughly match the personal space kept in the United States. If future research indicates that people prefer a different amount of space between themselves and a robot, the cost function can be grown or shrunk accordingly.

c) Pass on the Right: When approaching a person who is traveling in the opposite direction, people typically avoid collision by moving to one particular side. In the United States, people tend to move to their right [13]. This tendency can be modeled by adding a region of increased cost to the right-hand side of people in the environment. In a head-on encounter, this will cause the robot to prefer to stay to its right (the person’s left). Modeling this convention in this way also accounts for the tendency of people to pass a slower-moving person headed in the same direction on the left. As with personal space, the convention to pass on the right can

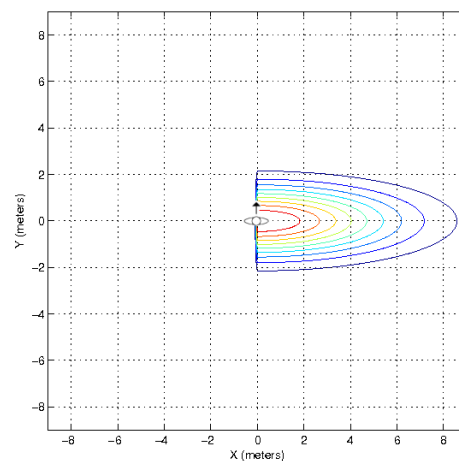


Fig. 3. Pass-on-the-right cost function for a person moving along the positive Y-axis, with a relative velocity of 1 m/s toward the robot. The person is shown at point (0,0). Cost is composed of two Gaussian functions; to the right of the Y-axis, $\sigma_x = v$ and $\sigma_y = 4v$ where v is the relative velocity between the person and robot; to the left of the Y-axis, $\sigma_x = v$ and $\sigma_y = 0.01$.

be modeled as a mixture of Gaussian functions, as shown in Figure 3.

4) Default Velocity: Just as people tend to keep a set pace (to minimize energy expenditure) [12], the robot should prefer to keep a constant velocity. Changes to the default velocity should result in a cost to the robot, so that, for example, the robot would have a cost trade-off between slowing down versus traveling a greater distance around an obstacle or person. We model this objective as proportional to the absolute difference between the chosen velocity and the default velocity; that is, both increases and decreases in speed incur a cost, and greater changes cause greater costs.

5) Inertia: The inertia constraint is similar to the default velocity constraint, except that it applies to rotational velocity. Again, just as people prefer to move in a straight line, the robot also should prefer to keep the same heading, rather than turning.

IV. IMPLEMENTATION DETAILS

In implementing our framework, we made several design decisions regarding the planner, person-tracking method, and robot navigation. In the following, we provide details of these decisions.

A. Planning

Robots that operate in the real world need to respond rapidly to changes in the environment. A plan to the robot’s goal, generated at the robot’s starting location, quickly becomes invalidated as the environment changes or the robot receives new information. Because sensors are imperfect, robots navigating in dynamic environments must replan whenever they receive new sensory data in order to ensure a safe, low-cost path.

We use the heuristic planner A* [21] to produce paths. However, A* alone typically cannot run repeatedly in real-time, which is necessary for a mobile robot operating in the

real world. While many variants of A* have been developed to operate in real-time (e.g. D* [22]), none are capable of replanning for a moving robot amongst dynamic obstacles, particularly when those obstacles have associated costs (e.g. personal space). Instead, our planning algorithm makes the following modifications to a vanilla A* planner: use of a variable-sized grid, and an objective function that accounts for multiple constraints.

1) *Variable Grid*: Rather than planning on a single resolution grid, our approach uses a variable grid that is composed of regions of regular grids of decreasing resolution, spanning outward from the robot’s position. The key idea behind this method is that, if the search can be done quickly enough, then the robot can regenerate plans at each timestep (as it gets new sensor information). Thus, the plan needs to be at a high resolution only near the robot; a rough path is sufficient further from the robot because it will generate a new plan before reaching those areas. The grid changes as the robot moves, keeping the finest-resolution cells centered over the robot’s position. This approach is described in detail in [23].

At each grid cell, the robot’s available actions will move it straight ahead one cell or drive diagonally to the left or right (ending at 45° angles). In addition, each action can be performed at any of three possible speeds: the default speed (0.5m/s), 50% faster (0.75m/s), or 25% slower (0.375m/s). Thus, in total, the robot can choose between nine possible actions (direction x speed). Though these actions are discretized, the robot’s navigation algorithm converts them to continuous arcs, allowing for smooth motion (see Sec. IV-C).

2) *Objective Function*: Given a set of constraints, they must be combined into a single objective function for the A* planner. We combine the various cost functions linearly, with a weighting for each constraint. That is, the total cost of some action a can be defined as:

$$a = \sum w_i \cdot c_i(a) \quad (1)$$

where $c_i(a)$ is the action cost of constraint i and w_i is the weight associated with that constraint. Additional constraints can be added in a similar fashion.

The relative weighting of constraints creates an implicit trade-off between potentially conflicting conventions. For example, the desires to pass oncoming people on the right and to maintain a buffer around obstacles may conflict if an oncoming person is not walking along the expected side of a hallway. The planner examines the spaces on either side of the person and chooses the side with the lower overall cost.

In our current implementation, the constraints of “shortest distance,” “default velocity,” and “inertia” each have a weight of $w_i = 1$; the “obstacle buffer,” “personal space,” and “pass on the right” constraints each have a weight of $w_i = 3$. These weights were chosen as values that produce relatively human-like behavior. More on the parameter selection is discussed in Section VI.

B. Person Tracking

We use a laser-based person-tracking method similar to that used in [24]. However, we modified the tracker in two

key ways: first, to use a map of the environment, and second, to better smooth the tracked velocities.

1) *Map-based tracking*: For this work, we make the simplifying assumption that the robot has an accurate *a priori* map of the environment.¹ The robot uses the map to match a given laser scan to its location in the environment. Non-matching segments of scans are segmented into person-sized blobs, which are tracked continuously using particle filters.

2) *Velocity smoothing*: Because several of the social constraints in our framework depend on the person’s direction of travel, the robot needs to have an accurate estimation of the person’s velocity. We do this by performing a linear least-squares regression on the person’s tracked position over time. In planning, the robot uses the most current estimation of the person’s velocity to predict his or her future location.

C. Navigation

Plans are generated rapidly, and the robot must be able to navigate along the paths as they change. The robot follows the generated plans using the Pure Pursuit path-following algorithm [25], which guides the robot back onto the path if it strays or if a new path is planned. Each plan is followed until a new plan is received. Since the planner typically runs as fast as new sensor data is received, the navigator will always have a high-resolution plan to follow.

V. RESULTS

Currently, all of our work has been done in simulation, using the Carmen simulator², though we have recently begun experiments on a physical robot. We simulate a circular robot with a diameter of 36 cm that uses a Hokuyo URG scanning laser rangefinder. People are generated in simulation by adding sets of “legs” to the simulated laser readings. The simulator intentionally adds noise to the readings to approximate a real-world sensor.

Two experiments were run in simulation to verify the behavior of the algorithm. Both experiments were run with the constraint weightings presented in Section III-B.

A. Experiment 1: Different goal locations

The first experiment tested to see how the robot’s behavior is influenced by the locations of the goal and of a person in the vicinity. We ran this study as a 3 goal x 3 person location experiment, as shown in Figure 4. In each trial, the robot began centered in a hallway. The goal location required the robot to either turn right or left down an intersecting hallway, or to continue straight past the intersection. Each trial simulated one person traveling in a straight line at 0.5m/s toward the robot, with the person either left, right, or centered in the hallway. The hallway was wide enough that the robot could pass on either side of the person in all cases. One hundred trials were run for each goal–person combination. The results are summarized in Table I.

¹This could be generalized by having the robot run a simultaneous localization and mapping (SLAM) algorithm.

²Carmen software is available online at <http://carmen.sourceforge.net/home.html>.

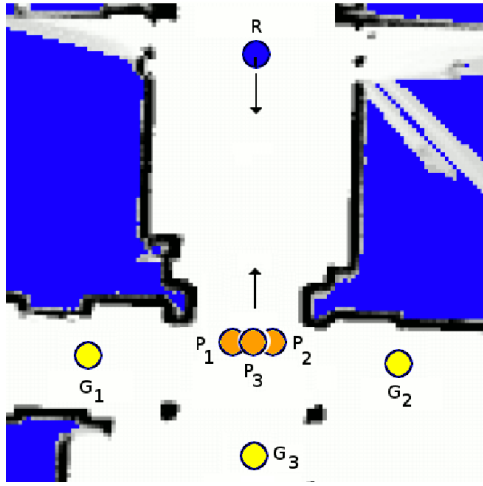


Fig. 4. Scenario for the first experiment. The robot (blue circle labeled “R”) starts centered in the hallway. Each trial selected one of three possible goals (yellow circles)—right turn (“ G_1 ”), left turn (“ G_2 ”), or straight ahead (“ G_3 ”)—and one of three person locations (orange circles)—somewhat to the left (“ P_1 ”), somewhat to the right (“ P_2 ”), or centered in the hallway (“ P_3 ”). The orange circles mark the approximate locations of the person at the point when the robot passed; the person’s starting location is not shown.

TABLE I

TIMES THE ROBOT YIELDED TO THE PERSON BY MOVING TO THE RIGHT, SO THAT THE PERSON PASSED ON THE ROBOT’S LEFT, IN EACH OF THE SCENARIOS DEPICTED IN FIGURE 4. EACH TRIAL WAS RUN 100 TIMES.

Goal Location	Person’s location in hallway		
	Left (P_1)	Right (P_2)	Centered (P_3)
Right turn (G_1)	85	100	100
Left turn (G_2)	7	95	44
Straight (G_3)	52	100	91

In general, when encountering an oncoming person, the robot chose to move to its right, as is socially expected. This behavior was dominant in six out of the nine conditions. In the case where the person was traveling on the left (“wrong”) side of the hallway and the robot needed to turn to its left, the robot stayed to the left of the hallway to avoid the person. More ambiguous behavior was seen in the final two cases: first, where the robot was traveling straight ahead while the person stayed toward the left of the hallway; and second, where the robot needed to turn left while the person traveled down the center of the hallway. In each of these cases, the robot tended to move right approximately half of the time.

B. Experiment 2: Crossing a person’s path

All of the person–robot encounters in Experiment 1 occurred at approximately the same location in the hallway, just as the robot began to enter the hallway intersection. This second experiment was designed to study how the robot’s behavior changes given different encounter locations. In particular, we studied the case where the robot must make a left turn while a person walks toward it, on the right side of the hallway. This scenario is shown in Figure 5. As in the previous experiment, each person traveled at 0.5m/s in a straight line, and each condition was run one hundred times. The results of this experiment are summarized in Table II.

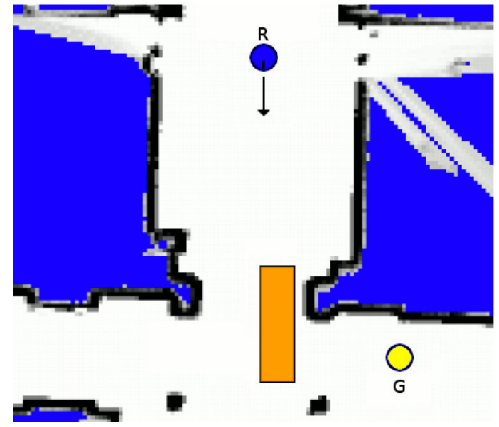


Fig. 5. Scenario for the second experiment. The robot (blue circle) starts centered in the hallway, with a goal (yellow circle) around a corner to the robot’s left. A person travels toward the robot, on the right side of the hallway. The person’s starting location is manipulated so that the person passes the robot when it is somewhere within the encounter region, depicted as an orange rectangle.

TABLE II

TIMES THE ROBOT TURNED ACROSS THE HALLWAY IN FRONT OF THE PERSON, CROSSING TO THE LEFT SIDE OF THE CORRIDOR, FOR VARIOUS ENCOUNTER LOCATIONS. EACH RESULT IS OUT OF 100 TRIALS.

Approximate Encounter Location (w.r.t. Robot)	Times Robot Turned in Front
Beyond the goal	99
Even with goal	78
Just before goal	40
Edge of intersection	13
Before intersection	8

As the results show, the person’s position does influence the robot’s behavior. Even though in all cases the robot had space to cross in front of the person to pass on the person’s right (which was the shortest path), the robot typically did so only when the encounter occurred in the intersection, rather than within the main hallway. In the hallway section, the robot more often chose to travel a greater distance in order to pass the person on the right side of the hallway.

VI. DISCUSSION

As our results show, the robot behaves in a predominantly “social” manner, passing people on the appropriate side given the situation. The generated paths differ based on many factors, including the goal location and the location of the person. Two sources of noise in the simulation can explain the trial-to-trial differences within each scenario: first, from the person tracking, and second, from the localization. Both of these modules are highly probabilistic, creating uncertainty in both the robot’s and the person’s positions. In particular, this noise can explain the ambiguous cases in Experiment 1. In both cases, social conventions allow the robot to travel on the left or right of the hallway for similar overall costs. As a result, minor discrepancies in the robot’s or person’s position can significantly change the path.

While these experiments used one set of constraint weights, we note that the robot’s behavior can be modified

by using different weightings. For example, weighting the “prefer right” cost significantly higher than others will result in the robot nearly always passing people by moving to the right side of the hallway, regardless of the robot’s final goal. Reducing its weight would cause the robot to choose the shortest path, instead. Tuning the weight parameters may be difficult; however, since social behavior between individual people varies widely, we expect that a wide range of constraint weights will produce acceptable social behavior.

In addition, the constraints presented here show a distinct North American bias. Personal space is very large, the robot tends to the right side of hallways, and occasionally cutting in front of people is acceptable behavior. However, modifying these behaviors to match other cultural norms is not difficult, and primarily involves modifying the size (but not basic shape) of constraints such as personal space and altering the relative weightings. More research would need to be done to fully understand what changes would need to be made.

Finally, it is interesting to note that the simulated people do not behave according to social norms, themselves. Rather, they travel in straight lines, and do not yield to the robot or respect its “personal space.” In the real world, this is likely to be different. In particular, if a person moves out of the way of the robot, the robot would be able to detect the person’s movements and react appropriately. Furthermore, we note that all of our tests presented here simulated exactly one person; further studies must be done with varying numbers of people moving around the robot.

VII. CONCLUSIONS AND FUTURE WORKS

We have introduced the COMPANION framework: a Constraint-Optimizing Method for Person-Acceptable Navigation. In this framework, social conventions are represented as constraints on a robot’s navigation, generating human-like paths. We have described the framework’s rationale and implementation, and we have presented the results from several experiments that demonstrate the robot’s social behaviors.

Currently, our work implements social conventions as if the robot were a person. While some evidence indicates that people do treat robots as human-like (see e.g. [2], [20], [24]), more work needs to be done to fully understand these social conventions as they relate to robots. To this end, we have begun testing our system on a physical robot. This will allow us to begin answering questions of how people expect and prefer the robot to behave. We expect that appropriate social behavior will be possible by simply adjusting the parameters within the COMPANION framework.

ACKNOWLEDGMENT

This research was partly funded by National Science Foundation Grant IIS-0624275.

REFERENCES

[1] R. M. Siino and P. J. Hinds, “Making sense of new technology as a lead-in to structuring: The case of an autonomous mobile robot,” in *Best Paper Proceedings of the Academy of Management*, New Orleans, LA, Aug. 2004.

[2] B. Mutlu and J. Forlizzi, “Robots in organizations: The role of workflow, social, and environmental factors in human–robot interaction,” in *Proceedings of Human–Robot Interaction (HRI)*, Mar. 2008, pp. 287–294.

[3] V. M. Olivera and R. Simmons, “Implementing human-acceptable navigational behavior and a fuzzy controller for an autonomous robot,” in *Proceedings WAF: 3rd Workshop on Physical Agents*, Murcia, Spain, Mar. 2002, pp. 113–120.

[4] E. Pacchierotti, H. I. Christensen, and P. Jensfelt, “Embodied social interaction for service robots in hallway environments,” in *Proceedings of the International Conference on Field and Service Robots (FSR)*, July 2005.

[5] Y. Nakauchi and R. Simmons, “A social robot that stands in line,” in *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, Oct. 2000, pp. 357–364. [Online]. Available: http://www.ri.cmu.edu/pubs/pub_3980.html

[6] P. Althaus, et al., “Navigation for human-robot interaction tasks,” in *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, Apr. 2004, pp. 1894–1900.

[7] W. Burgard, et al., “Experiences with an interactive museum tour-guide robot,” *Artificial Intelligence*, vol. 114, no. 1-2, pp. 3–55, 1999.

[8] R. Simmons, et al., “GRACE: An autonomous robot for the AAAI robot challenge,” *AAAI Magazine*, vol. 24, no. 2, pp. 51–72, Summer 2003.

[9] D. Shi, et al., “Human-aware robot motion planning with velocity constraints,” in *International Symposium on Collaborative Technologies and Systems (CTS)*, May 2008, pp. 490–497.

[10] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Siméon, “A human aware mobile robot motion planner,” *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, Oct. 2007.

[11] H. P. Williams, *Model Building in Mathematical Programming*, 4th ed. New York: Wiley, 1999.

[12] W. A. Sparrow and K. M. Newell, “Metabolic energy expenditure and the regulation of movement economy,” *Psychonomic Bulletin and Review*, vol. 5, no. 2, pp. 173–196, 1998.

[13] S. Bitgood and S. Dukes, “Not another step! Economy of movement and pedestrian choice point behavior in shopping malls,” *Environment and Behavior*, vol. 38, no. 3, pp. 394–405, May 2006.

[14] W. H. Whyte, *City: Rediscovering the Center*. New York: Doubleday, 1988.

[15] E. T. Hall, *The Hidden Dimension*. New York: Doubleday, 1966.

[16] N. L. Ashton and M. E. Shaw, “Empirical investigations of a reconceptualized personal space,” *Bulletin of the Psychonomic Society*, vol. 15, no. 5, pp. 309–312, 1980.

[17] J. R. Aiello, “Human spatial behavior,” in *Handbook of Environmental Psychology*, D. Stokols and I. Altman, Eds. New York: John Wiley & Sons, 1987, vol. 1, pp. 389–504.

[18] J. C. Baxter, “Interpersonal spacing in natural settings,” *Sociometry*, vol. 33, no. 4, pp. 444–456, Dec. 1970.

[19] M. Gérin-Lajoie, C. L. Richards, and B. J. McFadyen, “The negotiation of stationary and moving obstructions during walking: Anticipatory locomotor adaptations and preservation of personal space,” *Motor Control*, vol. 9, no. 3, pp. 242–269, July 2005.

[20] M. L. Walters, et al., “The influence of subjects’ personality traits on personal spatial zones in a human-robot interaction experiment,” in *Proceedings of CogSci-2005 Workshop: Toward Social Mechanisms of Android Science*, Stresa, Italy, July 2005, pp. 29–37.

[21] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths in graphs,” *IEEE Transactions on Systems Science and Cybernetics*, vol. SSC-4, no. 2, pp. 100–107, July 1968.

[22] A. Stentz, “The D* algorithm for real-time planning of optimal traverses,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-94-37, Sept. 1994.

[23] R. Kirby, R. Simmons, and J. Forlizzi, “Variable sized grid cells for rapid replanning in dynamic environments,” 2009, to appear in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).

[24] R. Gockley, J. Forlizzi, and R. Simmons, “Natural person-following behavior for social robots,” in *Proceedings of Human–Robot Interaction*, Arlington, VA, Mar. 2007, pp. 17–24.

[25] R. C. Coulter, “Implementation of the pure pursuit path tracking algorithm,” Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-RI-TR-92-01, Jan. 1992.