

# Socially-Aware Navigation Using Topological Maps and Social Norm Learning\*

Collin Johnson and Benjamin Kuipers

Computer Science and Engineering  
University of Michigan  
Ann Arbor, MI 48109, USA

## Abstract

We present socially-aware navigation for an intelligent robot wheelchair in an environment with many pedestrians. The robot learns social norms by observing the behaviors of human pedestrians, interpreting detected biases as social norms, and incorporating those norms into its motion planning. We compare our socially-aware motion planner with a baseline motion planner that produces safe, collision-free motion.

The ability of our robot to learn generalizable social norms depends on our use of a topological map abstraction, so that a practical number of observations can allow learning of a social norm applicable in a wide variety of circumstances.

We show that the robot can detect biases in observed human behavior that support learning the social norm of driving on the right. Furthermore, we show that when the robot follows these social norms, its behavior influences the behavior of pedestrians around it, increasing their adherence to the same norms. We conjecture that the legibility of the robot's normative behavior improves human pedestrians' ability to predict the robot's future behavior, making them more likely to follow the same norm.

## 1 Introduction

Social norms range from deeply held moral and ethical principles about lying, stealing, etc., to social conventions such as driving on the right side of the road (or on the left in the UK and Japan) and turn-taking protocols for drivers at rotaries or four-way stops. All along this spectrum, social norms are society's way to influence its individual members away from selfish individual utility maximization, and toward behaviors that produce greater benefits for the entire society. There are many compelling demonstrations of the tension between individual self-interest and the interest of society as a whole, including the Prisoners' Dilemma (Axelrod 1984) and the Tragedy of the Commons (Hardin 1968).

There is growing concern that robots, taking an increasing role in our society while remaining unconstrained by social norms, could produce harmful unintended consequences

---

\*This work has taken place in the Intelligent Robotics Lab in the Computer Science and Engineering Division of the University of Michigan. Research of the Intelligent Robotics lab is supported in part by grants from the National Science Foundation (IIS-1111494 and IIS-1421168).

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

even while pursuing human-given goals (Bostrom 2014). We therefore need to study, not only how social norms are expressed and applied, but how they can be learned from experience. It is important to study social norms with meaningful physical behaviors in the physical world, rather than simply by playing artificial games. Finally, for experimental study to be feasible, meaningful interactions where social norms are applicable must occur at time-scales of seconds to minutes, rather than days, weeks, or longer.

To meet these requirements, we focus on spatial navigation by a robotic wheelchair in a *campus-scale* environment – an office or industrial campus, shopping mall, medical or retirement center, or a college campus. To navigate safely and naturally in such an environment, the robot must perceive and predict the behaviors of nearby pedestrians, and plan its motion for safety as well as progress toward its goal.

Our existing motion planning algorithm (MPEPC (Park, Johnson, and Kuipers 2012; Park 2016)) plans a locally-optimal smooth trajectory within a five-second horizon, balancing the benefit of progress toward the goal against costs including the probability of collision with static or dynamic hazards. The optimal trajectory is recalculated 10-20 times per second, implementing a strategy called *model-predictive control*. Once learned, social norms can be incorporated into our motion planning framework by adding new terms to the cost-benefit trade-off.

In this paper, we present a method for learning social norms, exploiting the abstraction provided by our hybrid topological-metric map representation (Beeson, Modayil, and Kuipers 2010). For example, while traveling along a corridor, the robot should generally stay to the right and pass on the left, or when turning at an intersection, the robot should not cut the corners, which increases the chance of a collision.

We show how a topological map simplifies pedestrian intention estimation by providing a small set of possible action classes. The discrete representation of the current navigation situation provides two distinct benefits. First, social norms can be easily learned by counting the observed behavior instances in each situation. Second, the behaviors learned are conditioned only on the generic topological action taken, i.e. moving down a corridor or from one place to another. Thus, learned norms generalize to any environment with the same topological description, allowing the robot to behave appropriately in previously unvisited environments. We have im-

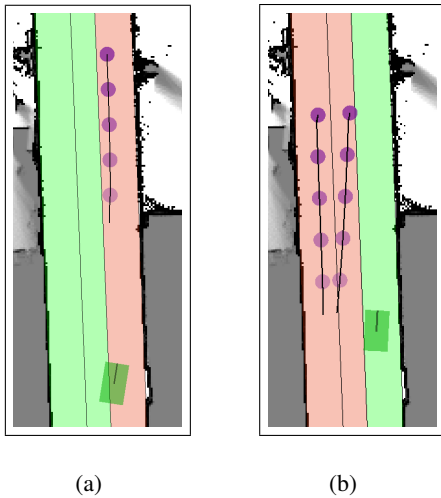


Figure 1: The robot encounters an oncoming pedestrian moving toward it on the right side of a corridor (a). By following the social norm of staying right, the robot implicitly signals to the pedestrian to move to the robot’s left to pass (b). While this interaction occurs, a second pedestrian appears moving along the left wall.

plemented social norm learning on our intelligent robotic wheelchair and have integrated preferences based on social norms into our baseline motion planner, MPEPC (Park, Johnson, and Kuipers 2012; Park 2016), creating a Socially-Aware version (SA-MPEPC), which allows our wheelchair to move more naturally through everyday environments.

After discussing related work in Section 2, we describe the representation of topological intentions in Section 3. Section 4 describes how social norms are learned, and Section 5 describes how they are used for motion planning in SA-MPEPC. Section 6 presents our evaluation and its results.

## 2 Related Work

Improvements in mapping, localization, and control are enabling robots to leave confines of the lab and operate in real-world, dynamic environments. This shift in operating environments has led to increasing interest in *socially-aware navigation*, where a robot considers additional factors beyond the traditional measures of safety and efficiency in deciding how to move through the world. As noted in a review of the field (Kruse et al. 2013), most of these additional performance measures are drawn from social science research into proxemics (measures of interpersonal distance) (Hall 1966) and social norms (accepted cultural behaviors for a variety of situations) (Lapinski and Rimal 2005). A recent study in (Zanlungo, Ikeda, and Kanda 2012) provides experimental evidence for the commonly acknowledged social norm of moving along the right (or left) of a corridor, even when not interacting with other pedestrians.

The importance of social awareness for improving human comfort in the proximity of robots has been established by a number of studies (Shiomi et al. 2014; Lasota

and Shah 2015; Joosse et al. 2013; Pacchierotti, Christensen, and Jensfelt 2006). However, because social norms are so loosely-specified and culture-dependent, the most successful approach for making a robot understand and follow social norms is to learn them from observing people’s behavior in a variety of social situations. Most relevant to our work are approaches that learn spatial navigation behaviors in the environment.

(Chung and Huang 2010; 2012) present the Spatial Behavior Cognition Model (SBCM) that divides the navigation problem into General (GSE) and Specific Spatial Effects (SSE). GSEs define a similar set of metrics to MPEPC, whereby costs are associated with static and dynamic obstacles, as well as actions. SSEs are environment-specific features that influence pedestrian behaviors, learned using a histogram of pedestrian motion in the environment. Unlike SSEs however, the responses learned by our approach generalize to any environment with a structure that can be represented as a topological map.

(Kim and Pineau 2016) use Inverse Reinforcement Learning to learn a cost function associated with navigation actions to navigate a smart wheelchair around dynamic obstacles. However, their approach is limited by choice of sensors and thus unable to generalize to fully autonomous motion. Additionally, they only learn passing behaviors and default to the shortest path to the goal otherwise. An alternate approach in (Dondrup et al. 2015) creates a qualitative description of a passing interaction and learns state transition probabilities for how the passing action should be performed. Their approach focus on passing behaviors between a single person and robot, which limits the applicability of their method to more complex scenarios. More complex passing behaviors are learned in (Chen et al. 2017) using deep reinforcement learning. They achieve impressive performance in learning norms for passing pedestrians and navigating in a busy real-world environment, but they also focus on solely on passing behaviors and otherwise optimize for time efficiency.

## 3 Situations for Topological Navigation

Our map representation is based on the Hybrid Spatial Semantic Hierarchy (HSSH) (Beeson, Modayil, and Kuipers 2010), where a topological map consists of a set of discrete, non-overlapping areas (places and paths). *Gateways* define the boundary between adjacent areas and are simple line segments whose endpoints are static obstacles in the environment.

Navigation through such a topological map can be described using two actions: traveling along a path or transitioning from one area to another. When traveling along a path, an agent moves towards the place at one of the ends, which we refer to as  $p^+$  or  $p^-$ . When transitioning between areas, an agent crosses a gateway  $g$  separating them.

When executing these actions, an agent may encounter crowded intersections or empty hallways. It might need to give way to an oncoming pedestrian or follow a pedestrian through a corridor. We formalize these different scenarios using the concept of a *situation*.

A *situation*,  $S_t = \langle M, X_t, O_t, \alpha_t \rangle$ , describes the robot’s navigation in terms of the environment’s topological and metric map  $M$ , the robot’s current pose and topological location  $X_t$ , the observed pedestrians  $O_t$  near the robot, and the current topological action being executed  $\alpha_t \in \{\Psi, \Delta\}$ , where  $\Psi$  is a path travel action and  $\Delta$  is a transition action.

We describe each pedestrian’s state  $o \in O_t$  based on the set of possible actions for its topological location. For a motion along a path ( $\alpha = \Psi$ ), a pedestrian is either moving towards the place at one of the ends, which we refer to as  $p^+$  or  $p^-$ , depending on whether the pedestrian is moving in the same (+) or opposite (−) direction as the agent. When transitioning to a new area ( $\alpha = \Delta$ ), the pedestrian is either crossing in the same ( $g^+$ ) or opposite ( $g^-$ ) direction as the agent. The topological action is estimated by adapting BH-MIP (Ferrer and Sanfeliu 2014) to estimating goals based on gateways rather than single points in the environment.

## 4 Learning Navigation Social Norms

A social norm describes the expected behavior of an agent in the environment when faced with a social situation. However, social norms are loosely defined, so a variety of people following the same social norm, like moving along the right side of a corridor, are likely to exhibit a range of behaviors. To account for the variation in behaviors, we represent a social norm as a probability distribution over possible behaviors, which can be learned by observing how pedestrians behave when responding to various situations.

Formally, we learn probability distributions across possible robot poses  $\bar{x}_t$ , given a navigation situation  $S_t$ :

$$p(\bar{x}_t|S_t) = p(\bar{x}_t|M, X_t, O_t, \alpha_t) \quad (1)$$

$$\approx p(\bar{x}_t|O_t, \alpha_t) \quad (2)$$

We can approximate the complete distribution in (1), which depends on the robot’s location  $X_t$  in a particular map  $M$ , using a more general situation description that depends only on agent states  $O_t$  and the current action  $\alpha_t$  (2). By ignoring conditioning on the robot’s specific state within a map, we learn a distribution that describes social norms for topological actions in any environment, even previously unexplored environments. However, if the robot continually operates within a single environment, (2) can be used as a prior for learning the more complex distribution in (1), which allows the robot’s behavior within known environments to be refined over time to account for local variations in how people interact.

Our approach learns two related social norms. The first norm is a preference for the robot’s lateral position while traveling longitudinally along a path. The other norm describes where the robot should cross a gateway when transitioning from one area to another. Therefore,  $\bar{x}_t$  in (1) describes either the robot’s position along the line orthogonal a path segment’s axis or its position along a gateway boundary.

To ensure the learned model generalizes to new environments, we normalize  $\bar{x}_t$  to the range  $[0, 1]$ . In this range, 0 is the location of the left wall relative to the nominal direction of motion. Therefore, the left side of a corridor or gateway

relative to the center has distance in the range  $[0, 0.5)$  and the right side range is  $(0.5, 1]$ .

We then divide  $\bar{x}_t$  into  $N$  bins of equal size. By discretizing the normalized range for  $\bar{x}_t$ , (1) becomes a simple discrete distribution, whose entire state space can be easily enumerated. We estimate the parameters of this distribution by observing other agents’ responses to situations in the environment, incrementing the bin that corresponds to their position, and then normalizing over the total number of observations.

### 4.1 Learning Norms for Path Segments

When navigating a path segment,  $p(\bar{x}_t|O_t, \alpha_t = \Psi)$  is a distribution across the robot’s lateral position along the path segment, where  $\Psi$  is the action that takes the robot from one end of the path segment to the other. We estimate  $p(\bar{x}_t|O_t, \alpha_t = \Psi)$  by considering a simpler case first, where we ignore other agents in the environment, thus estimating  $p(\bar{x}_t|\alpha_t = \Psi)$ .

To estimate  $p(\bar{x}_t|\alpha_t = \Psi)$ , we observe other agents in the environment. For agents traveling through the environment, we can directly observe  $\bar{x}_t$  and count the number of instances of their position being in a bin  $n \in \bar{x}_t$ . Dividing by the total number of observation yields the probability distribution  $p(\bar{x}_t|\alpha_t = \Psi)$ .

In the more general case, we must consider how agents interact with one another. While the state for an agent’s action  $\alpha_t$  is simple, interactions amongst agents can be complex. In our representation, each agent can be in one of  $N$  lateral positions with a state of + or −, depending on their motion relative to the robot. For an environment with  $K$  objects, there are  $(2N)^K$  possible states.

Rather than attempting to directly learn a distribution across this potentially huge state space, we create a simplified representation of the situation and learn a new distribution:

$$p(\bar{x}_t|O_t, \alpha_t = \Psi) \approx p(\bar{x}_t|L_t, \alpha_t = \Psi) \quad (3)$$

The simplified situation  $L_t$  divides the path segment laterally into  $L$  bins. Each bin has one of three states:  $\{+, -, \emptyset\}$ , which indicates if that bin is empty ( $\emptyset$ ) or is occupied by an agent moving in the direction ( $\{+, -\}$ ).

To compute  $L_t$ , each bin  $l \in L$  is matched with the nearest agent occupying the lateral position of the bin in the direction the robot is heading. The state is assigned to ( $\{+, -\}$ ) based on the estimated goal, or ( $\emptyset$ ) if no object occupies the bin.

When learning  $p(\bar{x}_t|L_t, \alpha_t)$ , we can use the same basic counting approach as when learning  $p(\bar{x}_t|\alpha_t)$ . We create a description of the situation  $L_t$  for each pedestrian. In this description, the robot is included as an agent that can occupy one of the bins in  $L_t$ . However, we must compute a different description of the situation  $L_t$  for each pedestrian because they each have a different perspective. For example, in Fig. 1 with  $L = 2$ , the situation for the top right pedestrian is  $L_t = \{-, +\}$ , whereas the situation for the top left pedestrian is  $L_t = \{-, \emptyset\}$ .

## 4.2 Learning Norms for Transitions

Motion through from one area to another occurs by crossing the gateway between the areas. The relevant position  $\bar{x}_t$  for this action is where to cross the gateway boundary. Like the norm for path segments, we can learn this distribution by observing an agent’s position whenever it crosses a gateway boundary.

As with path travel, we first learn the simplified distribution  $p(\bar{x}_t|\alpha_t = \Delta)$  by ignoring other agents. We use a simple discrete distribution to represent  $p(\bar{x}_t|\alpha_t = \Delta)$ , which we estimate by measuring where each agent crosses a gateway in the environment and incrementing that the count of that bin, then normalizing over all measurements.

The distribution  $p(\bar{x}_t|O_t, \alpha_t)$  represents more complex interactions amongst agents as they move from one area to another via a gateway. For this distribution, we consider only the subset of agents  $P_t \subseteq O_t$  whose estimated action is to move across a gateway in the opposite of the robot:

$$p(\bar{x}_t|O_t, \alpha_t = \Delta) \approx p(\bar{x}_t|P_t, \alpha_t = \Delta) \quad (4)$$

For example, if the robot is entering a place through some gateway  $g$ , we consider all agents who are leaving the place through that same gateway. We condition the social distribution on  $|P_t|$ . Thus, we estimate the distributions for each value of  $|P_t|$  experienced by the robot during training using the same counting approach as with  $p(\bar{x}_t|\alpha_t)$ .

## 5 Socially-Aware MPEPC

MPEPC generates plans at a fixed time horizon  $T_H$  with an update rate of 5Hz. Each planning cycle computes the best robot trajectory for the next  $T_H$  seconds which requires estimating the future trajectory of all dynamic objects around the robot for the next  $T_H$  seconds as well. The best trajectory is computed by optimizing the progress towards the goal, as defined by a navigation function, weighted by the probability of safe motion.

We integrate the above goal prediction and social norm behaviors with MPEPC to create a new socially-aware MPEPC (SA-MPEPC) by defining a cost map for the learned norms from Section 4 to integrate them into the navigation function used for defining progress through the environment to bias the robot’s decision making towards obeying social norms.

In the existing MPEPC implementation (Park 2016), the robot’s navigation function is computed using the wavefront algorithm defined by Konolige (Konolige 2000). The wavefront grows from the goal position outward using 8-way connectivity. Each cell in free space contains the distance to the goal, thereby defining a gradient that can be followed to reach the goal. Walls are marked as infinite distance.

The navigation function used by MPEPC is computed using a combination of the distance to the goal and a cost function. In many applications, including MPEPC, cost is a function of the distance to obstacles. By ensuring the navigation function is infinite in collision states for the robot, the gradient of the navigation function will always lead the robot on a collision-free path to the goal.

We integrate the learned social norms into MPEPC by introducing an additive *social cost* into the cost function for

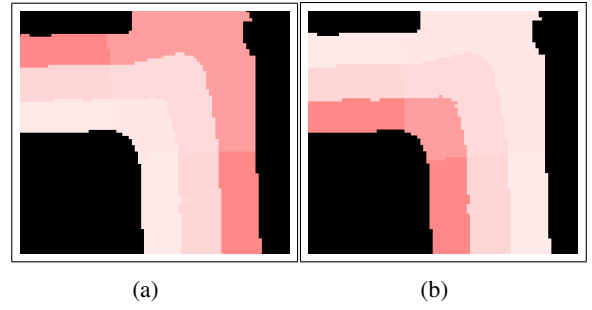


Figure 2: Examples of cost maps used to define the social norm navigation function. The cost map in (a) is for the robot making a right turn from the horizontal corridor to the vertical corridor. Note that the cost is different when navigating through the intersection because the cost is based on the transition norm rather than the path norm. In (b), the cost map is for executing a left turn at the same interaction. The high-cost and low-cost regions are reversed since the learned norm is to stay to the right.

computing the navigation function for each free cell in the occupancy grid:

$$C_{cell} = C_{obst} + C_{social} \quad (5)$$

The cost associated with being near an obstacle  $C_{obst}$  is an exponential function of the distance to the obstacle:

$$C_{obst} = \begin{cases} 0 & \text{if } d_{obst} \geq D_{max} \\ \alpha \left( \frac{D_{max} - d_{obst}}{D_{max}} \right)^\beta & \text{if } d_{obst} < D_{max} \end{cases} \quad (6)$$

Both social norms in Section 4 are probability distributions over the robot’s lateral position when moving along a path segment or across a gateway, where the robot should prefer moving through high-probability regions. The cost of being in a particular position is proportional to the learned probability of not being in a particular position:

$$C_{social} = \begin{cases} k_3 / (1 - p(d_{norm}|P_t, \alpha_t)) & \text{if } \alpha_t = G \\ k_3 / (1 - p(d_{norm}|L_t, \alpha_t)) & \text{if } \alpha_t = \Psi \end{cases} \quad (7)$$

where  $\sigma$  is an adjustable weight.

In our implementation, we use the Voronoi skeleton to compute  $d_{norm}$ . There are typically multiple branches of the Voronoi skeleton. We select the branches of interest by finding the shortest path along the skeleton between the entry and exit gateways. Only the skeleton cells along this shortest path are used for computing  $d_{norm}$ .

The normalized distance relative to the left wall for a cell in the map is  $d_{norm} = d_{obst} / (2d_{skel})$  when the skeleton is to the right of the cell. When the skeleton is to the left of the cell,  $d_{norm} = 1 - (d_{obst} / 2d_{skel})$ . Here  $d_{obst}$  is the distance to the nearest obstacle and  $d_{skel}$  as the distance of the nearest Voronoi skeleton cell to an obstacle.

Examples of the cost maps generated by (5) are shown in Fig. 2. Using such a cost map, a 2D navigation function can be computed using the wavefront algorithm. When computing the wavefront, we initialize the goal from which



Figure 3: The test environment used for evaluation, showing trajectories generated by MPEPC (blue) and SA-MPEPC (orange). All trajectories follow a figure-of-eight pattern, CW around the left loop and CCW around the right loop. (This figure is best viewed magnified and in color.)

the wavefront emanates to be the entire gateway boundary, rather than a specific point. Doing so ensures that the gradient of the navigation function will lead the robot across the gateway boundary, regardless of where it reaches it, as opposed to forcing it across at a single point, which isn't necessary for topological navigation.

## 6 Evaluation and Results

The goal of our socially-aware navigation algorithm is to enable the robot to learn how to behave in different social navigation situations, thereby improving its interactions with pedestrians to allow more socially acceptable and safer motion through the environment.

To test the effectiveness of our algorithm, we have implemented the socially-aware MPEPC algorithm on our robotic wheelchair, Vulcan (Williams et al. 2017). Vulcan is equipped with two Hokuyo URG-30LX lasers, an IMU, and wheel encoders. The software runs on a standard laptop with an Intel i7-4800MQ processor and 8GB of memory.

We learned the distributions in (3) and (4) using training data collected by Vulcan during autonomous explorations of campus buildings at the University of Michigan. While exploring, Vulcan was controlled using the MPEPC algorithm described in (Park 2016).

To evaluate the effectiveness of our new approach, we compare the performance of SA-MPEPC against the previous MPEPC algorithm. For this evaluation, we performed approximately 90 minutes of autonomous circuits with each algorithm around the figure-8 loop shown in Fig. 3. These circuits were performed at varying times of day, so the robot would encounter a more varied set of social situations and pedestrians. During autonomous navigation, the robot's pose and velocity were estimated at 50Hz, and the position and velocity of pedestrians were estimated at 20Hz using only the onboard laser sensors.

All robot-human interactions were with people carrying out their daily activities unaware of the experiment being

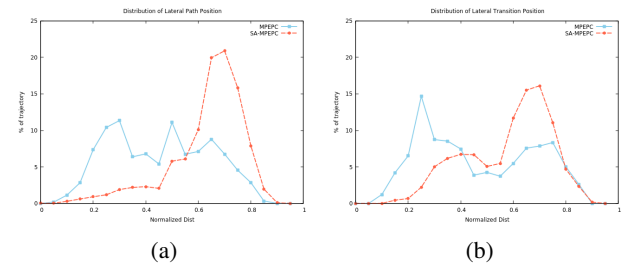


Figure 4: The distributions over the normalized lateral position of the robot show a clear shift to the right for both (a) travel along paths and (b) transitions across gateways.

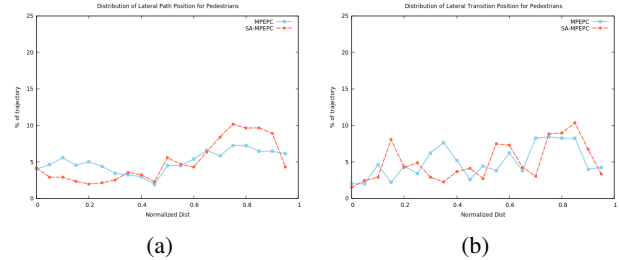


Figure 5: The observed pedestrian positions also see a shift to the right, though the effect is less dramatic.

performed. To ensure the safety of others, Vulcan was operated during all experiments by one of the authors who would manually intervene if unsafe conditions were detected.

### 6.1 Lateral Position During Navigation

In our approach, preferences for behavior influenced by social norms are learned as probability distributions over the agent's lateral position. Lateral position represented using the normalized distance from the left wall, with a distance of 0 being the agent touching the left wall and a distance of 1 touching the right wall. Since the test was performed in the US, the robot successfully learned and following social norms will be demonstrated by a significant increase (closer to 1) in the normalized lateral position of the robot.

The results of our experiment demonstrate a clear rightward shift in the robot's position while traveling along a path and across transitions, as can be seen in Fig. 4. We compared Gaussian distributions computed from the experimental data (Table 1) and found a significant difference between MPEPC and SA-MPEPC ( $p < .001$ ,  $t = 349$ ) supporting the hypothesis that a robot controlled by SA-MPEPC travels along paths closer to the right wall. Similarly, comparing Gaussian distributions for the lateral position when transitioning between areas, we found a significant difference between SA-MPEPC and MPEPC ( $p < .001$ ,  $t = 41.9$ ).

In analyzing our data, we also explored how the robot's adherence to social norms affects the behavior of other agents in the environment.

We hypothesize that the robot's more normative behavior improves the adherence to norms of agents the robot interacts with. Table 1 shows a significant rightward shift in

Table 1: Distribution of normalized lateral positions for the robot and observed pedestrians, for travel along paths and transitions across gateways.

	SA-MPEPC	MPEPC	p-value
Robot Path	$0.66 \pm 0.02$	$0.48 \pm 0.03$	$< .001$
Robot Transition	$0.61 \pm 0.03$	$0.49 \pm 0.05$	$< .001$
Pedestrian Path	$0.62 \pm 0.08$	$0.55 \pm 0.09$	$< .001$
Pedestrian Transition	$0.58 \pm 0.08$	$0.57 \pm 0.07$	$< .3$

Table 2: Comparison of oncoming passing behaviors: minimum passing distance and frequency of passing on the left.

	SA-MPEPC	MPEPC	p-value
Count	61	42	
Distance (m)	$0.52 \pm 0.22$	$0.50 \pm 0.24$	$< .4$
% Left	88.5	59.5	$< .001$

the lateral position of observed agents traveling along paths ( $p < .001$ ,  $t = 19.35$ ). The shift in the mean of 0.07 corresponds to 15 – 20cm in the test environment, depending on the corridor. We do not, however, find a significant difference in the lateral position of transitions between the two approaches ( $p < .3$ ,  $t = 0.6745$ ).

## 6.2 Oncoming Pedestrian Avoidance Behavior

In addition to lateral positioning while navigating, we explored the behavior of SA-MPEPC when faced with another common scenario: passing an oncoming pedestrian moving the opposite direction along a corridor. During our experiment, performed 42 oncoming passes with MPEPC and 61 oncoming passes with SA-MPEPC (see Table 2).

To assess the safety of the passing behavior, we looked at the passing distance between the robot and the oncoming pedestrians. We found no significant difference ( $p < .4$ ,  $t = 0.372$ ) between the average passing distance of 0.52m for SA-MPEPC versus 0.50m for MPEPC. This behavior is expected because MPEPC does not rely on the navigation function to ensure safety because progress towards the goal is weighted by the probability of collision with pedestrians. Therefore, safe distances are determined by the uncertainty of the robot’s perception of the environment, along with encoded preferences on passing distances, which do not change between SA-MPEPC and MPEPC.

While the passing distance is similar, the qualitative behavior (pass on the left or the right) of SA-MPEPC does show improved conformance to passing norms. In the US, the expected behavior is for each agent to stay to the right, so an oncoming pedestrian will be to the left when passing. We found a significant increase ( $p < .001$ ,  $z = 3.421$ ) in the percent of time SA-MPEPC passes an oncoming pedestrian

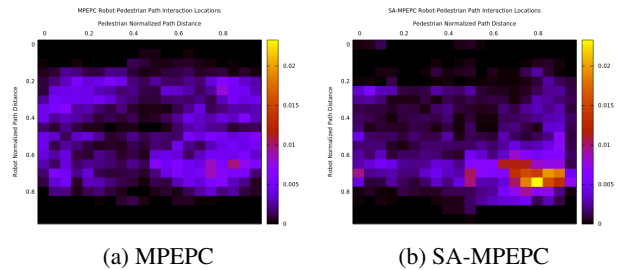


Figure 6: 2D histograms encoding the number of robot-pedestrian interactions at each combination of normalized lateral positions of the pedestrian (x) and robot (y). SA-MPEPC has a dominant peak in the lower right corner, corresponding to both the pedestrian and robot being on their own right sides of the corridor. In contrast, MPEPC has a much weaker peak in the bottom right corner and more interactions with a left-side robot and right-side pedestrian (top right) or a left-side robot and left-side pedestrian (top left).

on their left.

This improvement in passing behavior comes from the preference to stay to the right. Often, the robot is already moving to the right, so an oncoming pedestrian, also moving on their right, can simply pass the robot with no deviation from the trajectory. In more complex situations, where the robot is not moving along the right side of the corridor, perhaps due to clutter or other pedestrians, the learned norm encodes a preference to move right. As a result, the gradient in the navigation function will bias the robot to move right to avoid the collision.

## 7 Discussion

In order to function well in our society, robots must be able to learn and follow human social norms. We have shown how a robotic wheelchair, Vulcan, designed for safe motion in environments with many human pedestrians, can learn human social norms for travel in the shared environment. The hybrid topological-metrical spatial representation we use is critical for abstracting behavior in the environment, providing two major benefits: (a) tractable representation of observed behavior, and (b) generalization of learned norms to previously unseen environments.

Vulcan’s baseline motion planner (MPEPC (Park 2016)) can guarantee safe motion, but it optimizes safe progress toward its target without a bias toward any lateral position. In our experiments, Vulcan observes a bias in lateral positions of moving pedestrians, and interprets this bias as a social norm that it should follow. This social norm is embodied in an extended motion planner (SA-MPEPC). Vulcan follows this social norm, both as it moves along path segments and also as it performs right or left turns at decision points in the environment (Fig. 3).

We also observed that when Vulcan behaves according to this social norm (using SA-MPEPC), human pedestrians that Vulcan encounters follow the social norm more strongly than they did in the original condition, illustrating the mutually-reinforcing nature of social norms.

## References

- Axelrod, R. 1984. *The Evolution of Cooperation*. Basic Books.
- Beeson, P.; Modayil, J.; and Kuipers, B. 2010. Factoring the mapping problem: Mobile robot map-building in the hybrid spatial semantic hierarchy. *The International Journal of Robotics Research* 29(4):428–459.
- Bostrom, N. 2014. *Superintelligence: Paths, Dangers, Strategies*. Oxford University Press.
- Chen, Y. F.; Everett, M.; Liu, M.; and How, J. P. 2017. Socially aware motion planning with deep reinforcement learning. *arXiv preprint arXiv:1703.08862*.
- Chung, S.-Y., and Huang, H.-P. 2010. A mobile robot that understands pedestrian spatial behaviors. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, 5861–5866. IEEE.
- Chung, S. Y., and Huang, H. P. 2012. Incremental learning of human social behaviors with feature-based spatial effects. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2417–2422.
- Dondrup, C.; Bellotto, N.; Hanheide, M.; Eder, K.; and Leonards, U. 2015. A computational model of human-robot spatial interactions based on a qualitative trajectory calculus. *Robotics* (4):63–102.
- Ferrer, G., and Sanfeliu, A. 2014. Bayesian human motion intentionality prediction in urban environments. *Pattern Recognition Letters* 44:134–140.
- Hall, E. T. 1966. The hidden dimension.
- Hardin, G. 1968. The tragedy of the commons. *Science* 162:1243–1248.
- Joosse, M.; Sardar, A.; Lohse, M.; and Evers, V. 2013. BEHAVE-II: The revised set of measures to assess users attitudinal and behavioral responses to a social robot. *International Journal of Social Robotics* 5(3):379–388.
- Kim, B., and Pineau, J. 2016. Socially adaptive path planning in human environments using inverse reinforcement learning. *International Journal of Social Robotics* 8(1):51–66.
- Konolige, K. 2000. A gradient method for realtime robot control. In *IROS '00*, volume 1, 639–646 vol.1.
- Kruse, T.; Pandey, A. K.; Alami, R.; and Kirsch, A. 2013. Human-aware robot navigation: A survey. *Robotics and Autonomous Systems* 61(12):1726–1743.
- Lapinski, M. K., and Rimal, R. N. 2005. An explication of social norms. *Communication theory* 15(2):127–147.
- Lasota, P. A., and Shah, J. A. 2015. Analyzing the effects of human-aware motion planning on close-proximity human-robot collaboration. *Human factors* 57(1):21–33.
- Pacchierotti, E.; Christensen, H. I.; and Jensfelt, P. 2006. Evaluation of passing distance for social robots. In *ROMAN 2006 - The 15th IEEE International Symposium on Robot and Human Interactive Communication*, 315–320.
- Park, J. J.; Johnson, C.; and Kuipers, B. 2012. Robot navigation with model predictive equilibrium point control. In *IROS*, 4945–4952.
- Park, J. J. 2016. *Graceful Navigation for Mobile Robots in Dynamic and Uncertain Environments*. Ph.D. Dissertation, University of Michigan – Ann Arbor.
- Shiomi, M.; Zanlungo, F.; Hayashi, K.; and Kanda, T. 2014. Towards a socially acceptable collision avoidance for a mobile robot navigating among pedestrians using a pedestrian model. *International Journal of Social Robotics* 6(3):443–455.
- Williams, T.; Johnson, C.; Scheutz, M.; and Kuipers, B. 2017. A tale of two architectures: A dual-citizenship integration of natural language and the cognitive map. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS '17*, 1360–1368. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems.
- Zanlungo, F.; Ikeda, T.; and Kanda, T. 2012. A microscopic “social norm” model to obtain realistic macroscopic velocity and density pedestrian distributions. *PLOS ONE* 7:1–10.