Sidewalk Delivery Robot Navigation: A Pedestrian-Based Approach

Yuqing Du[1], Nicholas J. Hetherington[1], Chu Lip Oon[1], Wesley P. Chan[1], Camilo Perez Quintero[1], Elizabeth Croft[2], and H.F. Machiel Van der Loos[1]

***Abstract*— In this paper, we propose a novel navigation system for mobile robots in pedestrian-rich sidewalk environments. Sidewalks are unique in that the pedestrian-shared space has characteristics of both roads and indoor spaces. Like vehicles on roads, pedestrian movement often manifests as flows in opposing directions. On the other hand, pedestrians also form crowds and can exhibit much more random movements than vehicles. Classical algorithms are insufficient for safe navigation around pedestrians and remaining on the sidewalk space. Thus, our approach takes advantage of natural human motion to allow a robot to adapt to sidewalk navigation in a safe and socially-compliant manner. We developed a *group surfing* method which aims to imitate the optimal pedestrian group for bringing the robot closer to its goal. For pedestrian-sparse environments, we propose a sidewalk edge detection and following method. Underlying these two navigation methods, the collision avoidance scheme is human-aware. Components of the navigation stack are demonstrated in simulation and an integrated simulation and real-world experiment are discussed.**

Fig. 1. Our mobile robot navigating on the sidewalk in a simulated environment, using pedestrian flow.

I. Introduction

Robots are increasingly being used in environments shared with people. To facilitate human-robot coexistence in such spaces, robots must be capable of navigating in a safe and socially-compliant manner. Such applications include mobile robots operating in elderly homes [1]; in hospitals [2]; or in public crowded areas [3]. Classical dynamic navigation algorithms are often unsuitable for these more stochastic, human-populated environments. These deterministic algorithms are less viable when there is more uncertainty and when movements of humans and robots can affect one another [4]. One common outcome of such traditional navigation methods is the freezing robot problem (FRP) where dense crowds cause the robot to be unable to move due to the belief that all possible paths will lead to collisions [5], [4]. Approaches that cause FRP do not account for the likelihood that people will adjust their path to allow passage for the robot (as for other pedestrians).

We argue that as robot technology is used in more commonplace applications, a crucial environment to implement human-robot coexistent navigation is on pedestrian sidewalks. Consider the case where a mobile robot is used for delivering packages to homes. Such a robot must be capable of achieving two main goals: 1) safely sharing the sidewalk space, and 2) navigating to its goal as efficiently as possible. Sidewalks present a unique yet challenging environment in that the navigable space combines elements of both roads and free indoor spaces. Like roads, sidewalk motion is generally restricted to two linear directions and the resulting navigable space is limited. However, pedestrians generally do not walk in perfect queues. Rather, people tend to walk in groups of variable size and speeds [6] and move along with a general self-organizing crowd flow [7]. Compared to autonomous road navigation, sidewalk navigation must also account for stochastic human movement that necessitates dynamic obstacle avoidance. Furthermore, certain social rules, such as walking in lanes or affording more space in direction of walking than in the perpendicular direction [7], are rules that a robot should follow as well.

Current research in human-aware path planning takes a variety of approaches. In general, methods are either deterministic, learned, or a combination thereof. Social rules, such as pedestrian conventions and appropriate proxemics, can be induced through either cost functions in deterministic approaches, or reward functions in reinforcement-learning based approaches [8]. Examples include using inverse reinforcement learning to learn human-like navigation [9]; people tracking combined with an iterative A* planner [10]; and modeling pedestrian intention to better predict movements of neighbouring people [11]. While each of these approaches take advantage of understanding human motion, they are often more suitable for open environments. In contrast, our approach specifically handles navigation in the more restrictive sidewalk. Here, the aforementioned approaches may be less effective as they do not account for the physical sidewalk boundaries, or how robot movement will affect pedestrian flow.

[1]Collaborative Advanced Robotics and Intelligent Systems (CARIS) Laboratory, University of British Columbia
[2]Monash University

The key research question this paper addresses is how mobile robots can utilize nearby pedestrian behaviours and flows to navigate towards a global goal. Our approach is to develop a system that imitates pedestrian group behaviour in real time. When our navigation stack detects people moving towards the robot's goal, a 'group surfing' behaviour is used. This allows the robot to imitate and participate in pedestrian social behaviours. In an unpopulated and simple sidewalk environment, the default behaviour is to follow a trajectory offset from the sidewalk curb towards the goal. Simultaneously, the underlying collision avoidance algorithm is capable of socially-aware behaviour to improve pedestrian comfort and safety in the shared space.

The organization of this paper is as follows. In section II we propose the overall navigation system. In section III we present the development of the three main navigation components that make the system. Section IV presents simulated experiments for evaluating our approach. Lastly, the paper is concluded in section V and potential further research is noted.

II. System

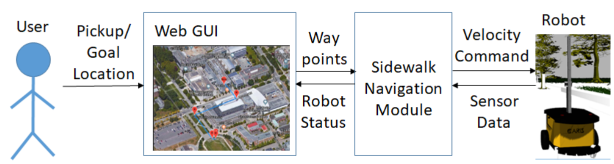


Fig. 2. System diagram

Our overall system is shown in Figure 2. We consider package delivery as our example task. The user interacts with our system using a web graphical user interface (GUI). The user first specifies a deliver destination for the package. The web GUI then generates a set of intermediate waypoints and sends them to the sidewalk navigation module. The sidewalk navigation module receives sensor data from the robot and plans the desired action for the robot by sending velocity commands to execute. The robot's status (i.e., its location) is sent back to the web GUI for user feedback.

A flow diagram of our system is shown in Figure 3. After the user selects a global goal G (deliver location), a path is computed by Google Maps' API with waypoints W_n located at each intersection between the robot's initial location and the goal. The sidewalk navigation module is in charge of moving the robot towards the waypoint. It uses two navigation modes, depending on whether there is a nearby pedestrian flow. Given a waypoint, if pedestrians moving towards the waypoint are detected, the navigation module employs a 'group surfing' method which relies on nearby pedestrian flow on the sidewalk for autonomous navigation. However, in case where no humans are present, the robot will utilize a contextual 'curb following' method. Each of these methods generate subgoals for the robot to navigate to, and in either case, the subgoals generated are sent to a collision avoidance algorithm which produces a velocity command.

At each cycle, the system will also continuously check for the robot's arrival at the given waypoint. Only if the robot has not reached the waypoint yet, then the waypoint will be sent to either group surfing or curb following. However, if the robot has reached within a predefined tolerance of the waypoint, the module checks if there are more waypoints. If more waypoints exist, the current waypoint is updated with the next waypoint. Otherwise, the module indicates that the robot has reached its goal and navigation is complete.

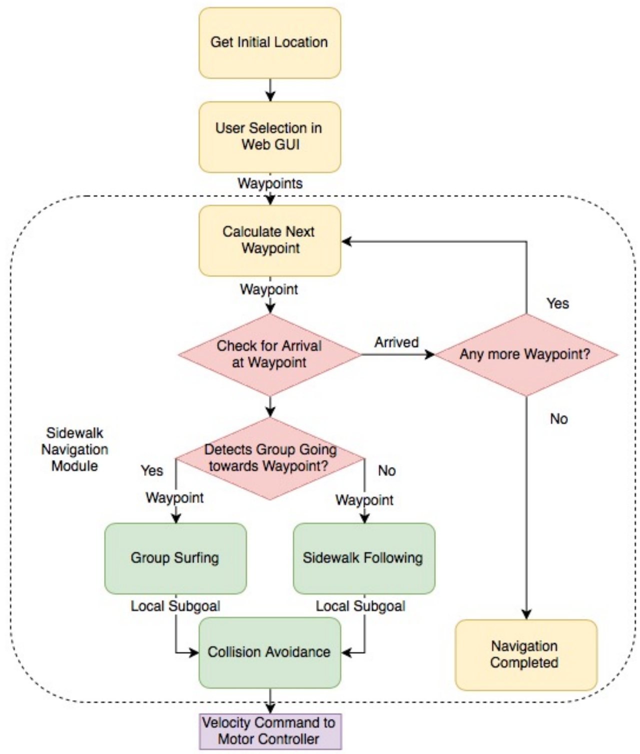


Fig. 3. Flow Diagram

III. Methods

A. Group Surfing

When given a waypoint to navigate to, 'group surfing' is the preferred method of navigation. The group surfing algorithm takes advantage of natural pedestrian behaviours through imitation. Such behaviours include: walking in lanes, avoiding collisions with other pedestrians or obstacles [7], waiting at intersections to cross, and not walking into traffic. As a result, many of the challenges in collision avoidance and remaining on the sidewalk are mitigated. To imitate these movements, the algorithm constantly computes the preferred pedestrian group location to set as a navigation subgoal.

Humans are detected through an open source leg detector [12]. The detected people are then fed through a people tracking pipeline from the SPENCER project to identify tracked groups of individual(s) [13]. Groups are classified using social relations, which the SPENCER project defines as "spatial relations between people via coherent motion indicators".

1) Filter Candidate Groups: At the lowest level, the robot should only imitate groups moving towards the waypoint. Consider the position and velocities of all parties in the global coordinate frame. Let the position of the waypoint (W) be $\boldsymbol{x}_W$ and the position of the robot be $\boldsymbol{x}_R$. The position of the waypoint relative to the robot is $\boldsymbol{x}_I = \boldsymbol{x}_W - \boldsymbol{x}_r$.

The tracked groups are sets G_i, where i can take positive integer values in the set $\{1, \ldots, n_G\}$ and n_G is the total number of groups. Each set G_i cannot be empty and contains pedestrians p_j where j can take positive integer values in the set $\{1, \ldots, n_{G_i}\}$ and n_{G_i} is the total number of people in group G_i. From the velocities $\boldsymbol{v}_{p_j}$ and positions $\boldsymbol{x}_{p_j}$ of the pedestrians in a given G_i, we compute the average group velocity $\boldsymbol{v}_{G_i}$ and determine the $\boldsymbol{x}_{p_j}$ closest to $\boldsymbol{x}_R$, giving the person $p_{closest}$ for each group.

Given this information, we then filter out groups moving away from the waypoint. For each G_i, compute $\boldsymbol{v}_{G_i} \cdot \boldsymbol{x}_I$. If this value is non-positive, discard G_i as a subgoal candidate.

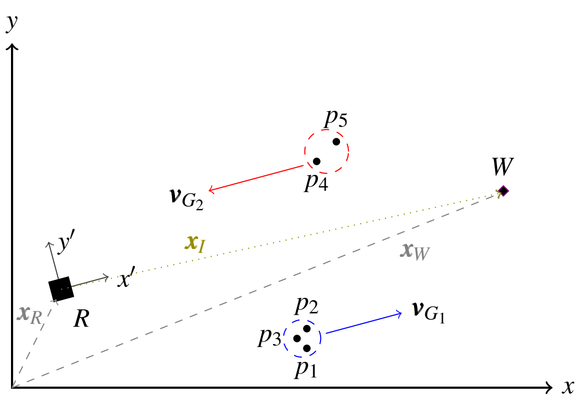


Fig. 4. In the global coordinate frame, we compute whether groups G_i are suitable for following. Given each pedestrian group's average velocity $\boldsymbol{v}_{G_i}$ and the position of the waypoint relative to the robot $\boldsymbol{x}_I$, computing if $\boldsymbol{v}_{G_i} \cdot \boldsymbol{x}_I > 0$ determines whether G_i will bring the robot closer to the waypoint. Here, G_2 is not a viable candidate as it is heading further from the waypoint.The x' and y' axes are in the robot's frame of reference and show the orientation of the robot.

2) Smart Group Selection: Once we have filtered out unsuitable groups, the algorithm selects the optimal group to follow. We define the optimal group as follows. Given a desired speed, often the robot's maximum safe speed v_{max}, a group G_n is defined as more optimal than a group G_m if $v_{max} - |\boldsymbol{v}_{G_n}|$ is a smaller positive value than $v_{max} - |\boldsymbol{v}_{G_m}|$. The robot will not follow a group where $|\boldsymbol{v}_{G_i}| > v_{max}$ as it will not be able to keep up with G_i. After determining the optimal group, the algorithm outputs the position of the $p_{closest}$ of that group as the group surfing subgoal. We intentionally select the closest person as a subgoal as attempting to reach the average group position could lead to colliding with pedestrians located between the average group position and the robot's current position. This subgoal selection is based on the assumption that $p_{closest}$ will move to a new location before the robot arrives. However, if this is not the case, the collision avoidance method will prevent the robot from colliding with the closest pedestrian. The optimal group at any given time is constantly updated, effectively causing the robot to 'surf' between different pedestrian groups to maximize efficiency.

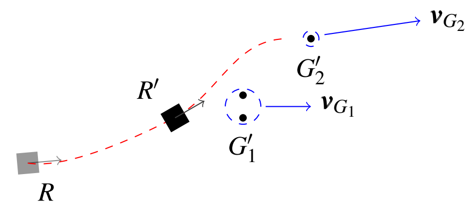


Fig. 5. Illustrating group surfing behaviour. R is the robot's initial position and the dotted red line illustrates the path taken. Initially the algorithm will lead the robot to imitate the closer G_1 until it reaches R', when G'_2 is within the field of view. Since $\boldsymbol{v}_{G_2} > \boldsymbol{v}_{G_1}$, the algorithm will switch to imitating G'_2.

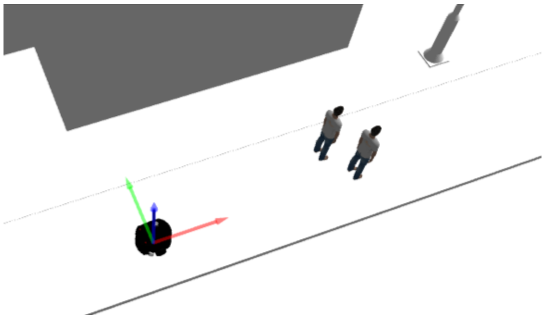
In summary, given a typical sidewalk where there may be groups of pedestrians moving either towards or away from a waypoint, the algorithm will constantly output the position of the optimal group to imitate (if it exists) as a local subgoal.

B. Pedestrian Independent Sidewalk Navigation

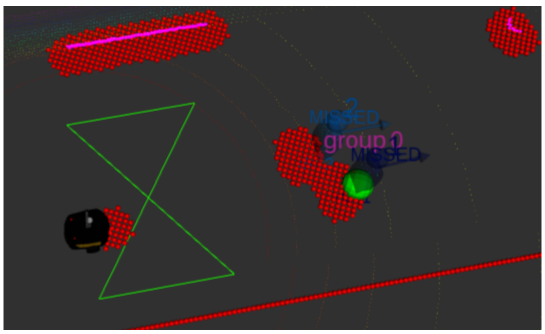
Our system defaults to group surfing for navigation. When pedestrians are not in the field of view of the robot, however, the sidewalk detection navigation is activated. We make use of contextual knowledge; sidewalks are normally surrounded by streets and buildings or empty space. To find the boundary between the sidewalk and the street, our robot first acquires a surrounding point cloud using a 3D laser sensor and filters out points above the plane defined by the robot wheel contacts ending with a point cloud set S. Then the Random Sample Consensus (RANSAC) method [14] is used to fit a plane on the remaining points by:

1) Selecting three non-collinear points in the filtered set S and compute a candidate model.
2) Computing distances from all other points in S to the plane.
3) Calculating a score value by counting the number of points inside of a 5 cm threshold. Steps one to three are performed for k iterations, the model with more inliers is selected.
4) The inlier points in the winner set are used to calculate the model coefficients in a least-squares formulation.
5) After projecting all the inlier points into the calculate plane a concave hull point cloud C is estimated with $\alpha = 5.0$ [15], where α describes the smoothness level of computed hull.

The set C is arranged in a kd-tree data structure and a nearest search algorithm is used to find the k nearest points to the robot. These points are used to fit a line that represents the curb (bottom red line in Figure 6(b)). Then, a local goal point is set in the parallel line to the curb line that passes across the robot location . This point is used as the goal for the collision avoidance approach described in Section III-C.



(a) Robot following a group of people



(b) Visualization of people group, curb and wall detection and avoidance for the scenario in (a)

Fig. 6. Group following and static obstacle avoidance. a) The robot following a group of pedestrians to its goal at the next intersection. b) Visualization showing: the curb detection described in Section III-B (green polygon); static obstacle avoidance described in Section III-C (small red spheres); and the pedestrian detection and group goal (green sphere) described in Section III-A

C. Collision Avoidance

1) Human-Aware Collision Avoidance: Either while following the selected group or curb, we also avoid groups and individual pedestrians we are not following. We use a learned approach, Socially-Aware Collision Avoidance with Deep Reinforcement Learning (SA-CADRL) [16], as the collision avoidance component of our navigation stack. The collision avoidance system navigates to a local goal generated by either the group surfing algorithm or the curb following approach.

The SA-CADRL policy $\pi : \boldsymbol{s_t} \rightarrow \boldsymbol{a_t}$ maps a state $\boldsymbol{s_t}$ to an action $\boldsymbol{a_t}$. A state $\boldsymbol{s_t} = (P_{r_t}, v_{r_t}, P_{0_t}, v_{0_t}, \ldots, P_{n_t}, v_{n_t})$ is the robot's own state and its observation of surrounding pedestrians, namely their pose P and velocity v. The corresponding action $\boldsymbol{a_t} = (v_t, \omega_t)$ is a control command consisting of a linear velocity v_t, and an angular velocity ω_t, that minimizes collisions and the time to goal, and maximizes the social awareness of the robot's motion.

The reinforcement training process induces social awareness through social reward functions, which give higher values to actions that follow social rules. Maximal social awareness is considered to be achieved when an action follows two socially-aware behaviours: staying to the right; and passing on the left. This training used simulated data, but the policy was demonstrated with hardware in a populated shopping-mall-like environment. We did not train the policy ourselves, but used the results made publicly available [17] by the authors.

2) Static Obstacle Avoidance: In addition to following groups, avoiding individuals, and/or following the curb, we avoid static obstacles. These obstacles include street posts, bins, benches, walls, and curbs. We also use SA-CADRL to avoid these static obstacles, by adding "static pedestrians" to the state vector s_t. Using data from the curb detection detailed in Subsection III-B and a 2D laser scan, we populate s_t with small, stationary "pedestrians" at the locations of the static obstacles. The action generated by the policy then avoids both these static obstacles and the real pedestrians.

IV. Demonstration

To validate our navigation approach we created a replica of our mobile robot and our research lab neighborhood in a simulation environment. We calibrated the measurement in the simulation with the real world using GPS coordinates, such that landmarks in the simulated world corresponds to those in the real world. We then populated the world with simulated pedestrians. As a case study, we sent the robot to the closest coffee shop. A user selects through a browser interface a deliver destination. Depending on the pedestrian flow and the robot's surroundings, the robot navigates by surfing across groups of people or by detecting the curb and following it. A detailed demonstration of our system is included in the attached video [1].

V. Conclusions

In this paper, we presented a novel system for navigation through sidewalks. Our group surfing method is a new navigation strategy based on observations of our surroundings and some assumptions about pedestrian flow. However, not all sidewalks are constantly populated. Thus, for pedestrian independent navigation, we follow the sidewalk edge. Simultaneously, we avoid static obstacles and other pedestrians with a socially-aware collision avoidance strategy.

Future work will further develop each component of our navigation system. For the group surfing component, future work can focus on improving the selection process for groups to imitate. In particular, it would be beneficial to filter based on criteria in addition to group velocity, such as group trajectory or group size. For collision avoidance, a more specialized technique would allow for more efficient navigation. Future work to this end will consider a reinforcement learning approach with data from a real sidewalk instead of the simulated data for a generic pedestrian environment that was used for SA-CADRL. The SA-CADRL policy also prioritizes constant linear motion, which results in arcing motions when faced with obstructions. These motions are not suitable for the static constrictions of a sidewalk environment, and will be remedied in a new policy. For the pedestrian sparse navigation, further improvements can be made for edge-following where curbs are less distinct or may be obscured.

[1] `https://youtu.be/lyZnBzAZ5DU`

REFERENCES

[1] N. Roy, G. Gordon, and S. Thrun, "Planning under uncertainty for reliable health care robotics," in *Field and Service Robotics*. Springer, 2003, pp. 417–426.

[2] M. Takahashi, T. Suzuki, F. Cinquegrani, R. Sorbello, and E. Pagello, "A mobile robot for transport applications in hospital domain with safe human detection algorithm," in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec 2009, pp. 1543–1548.

[3] R. Triebel, K. Arras, R. Alami, L. Beyer, S. Breuers, R. Chatila, M. Chetouani, D. Cremers, V. Evers, M. Fiore, *et al.*, "Spencer: A socially aware service robot for passenger guidance and help in busy airports," in *Field and service robotics*. Springer, 2016, pp. 607–622.

[4] P. Trautman, J. Ma, R. M. Murray, and A. Krause, "Robot navigation in dense human crowds: Statistical models and experimental studies of human–robot cooperation," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 335–356, 2015.

[5] P. Trautman and A. Krause, "Unfreezing the robot: Navigation in dense, interacting crowds," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 797–803.

[6] K. K. Laxman, R. Rastogi, and S. Chandra, "Pedestrian flow characteristics in mixed traffic conditions," *Journal of Urban Planning and Development*, vol. 136, no. 1, pp. 23–33, 2010.

[7] D. Helbing, P. Molnár, I. J. Farkas, and K. Bolay, "Self-organizing pedestrian movement," *Environment and planning B: planning and design*, vol. 28, no. 3, pp. 361–383, 2001.

[8] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, December 2013. [Online]. Available: https://doi.org/10.1016/j.robot.2013.05.007

[9] P. Henry, C. Vollmer, B. Ferris, and D. Fox, "Learning to navigate through crowded environments," in *2010 IEEE International Conference on Robotics and Automation*, May 2010, pp. 981–986.

[10] J. Müller, C. Stachniss, K. O. Arras, and W. Burgard, "Socially inspired motion planning for mobile robots in populated environments," in *Proc. of International Conference on Cognitive Systems*, 2008.

[11] Y. Tamura, P. Dai Le, K. Hitomi, N. P. Chandrasiri, T. Bando, A. Yamashita, and H. Asama, "Development of pedestrian behavior model taking account of intention," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 382–387.

[12] C. Pantofaru, "Ros people stack," https://github.com/wg-perception/people.

[13] T. Linder, S. Breuers, B. Leibe, and K. O. Arras, "On multi-modal people tracking from mobile platforms in very crowded and dynamic environments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 5512–5519.

[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[15] S. Asaeedi, F. Didehvar, and A. Mohades, "A-concave hull, a generalization of convex hull," *Theoretical Computer Science*, vol. 702, pp. 48–59, 2017.

[16] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *Intelligent Robots and Systems (IROS), 2017 IEEE International Conference on*.

[17] M. Everett, "Ros package for dynamic obstacle avoidance for ground robots trained with deep rl," https://github.com/mfe7/cadrl_ros.