

# The Hands-Free Push-Cart: Autonomous Following in Front by Predicting User Trajectory Around Obstacles

Payam Nikdel\*,†, Rakesh Shrestha\*,† and Richard Vaughan†

**Abstract**—This paper demonstrates an autonomous mobile robot that follows a walking user while staying ahead of them. Despite several useful applications for autonomous push-carts, this problem has received much less attention than the easier problem of following from behind. In contrast to previous work, we use multi-modal person detection and a human-motion model that considers obstacles to predict the future path of the user. We implement the system with a modular architecture of obstacle mapper, human tracker, human motion model, robot motion planner and robot motion controller. We report on the performance of the robot in real-world experiments. We believe that approaches to this largely overlooked problem could be useful in real industrial, domestic and entertainment applications in the near future.

## I. INTRODUCTION

Rapid developments in robotics will bring robots to our everyday lives. There are various applications where robots could usefully follow a human user around to assist them, for example a golf caddy or self-driving luggage. Notably, Boston Dynamics' LS3 legged robots (unpublished, derived from BigDog [1]) had a well-developed person-following capability to act as load carrying mules.

Ho et al [2] categorize person following into three categories: 1) following behind the leader, 2) side-by-side with the leader, and 3) ahead of the leader. Following behind the leader is much simpler as it can be implemented with a simple proportional controller that tries to keep the person in the middle of the detection space and at a certain distance. The other two tasks are significantly more challenging as they require a predictive model of the user's motion [3]. For instance, when entering an intersection, a robot following ahead should be able to predict which direction the user might take. This problem has not been addressed well.

Following in front has a number of useful applications. Consider the push-carts in daily use in logistics warehouses, hotels, libraries, and supermarkets; they are best placed right in front of the user for quick access. Jung et al [4] did an experiment in which participants were told to walk in a straight line while a robot followed them from behind. They observe that participants look back to check the robot out of curiosity or fear of getting hit by the robot, imposing a cognitive load. The second advantage of following-ahead is that the user can see the cart for security: e.g. when walking through an airport with valuable luggage. In entertainment, an automatic UAV camera platform that follows ahead could

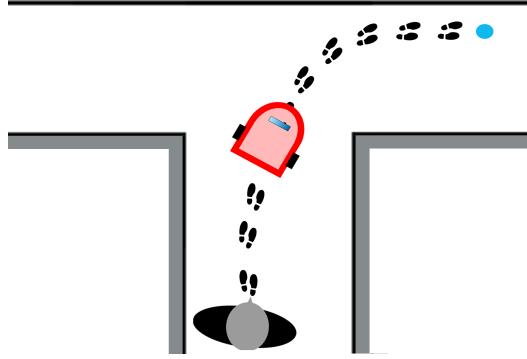


Fig. 1: Scenario of a mobile robot following ahead of a person. The robot must anticipate the person's future trajectory to stay in the correct position.

capture a view of the user's face in recreational activities like skiing and mountain biking. Autonomous push-carts are frequently proposed and commercial devices exist<sup>1</sup> but they typically remove the human in favor of map-based navigation.

In this paper, we update and extend the limited previous work on this task by considering the user's path through obstacles. Using an RGB-D camera and laser scanner data, we estimate the relative position and velocity of the uninstrumented user, using an Extended Kalman Filter (EKF), and predict their future trajectory motion using a simple motion model interacting with a local map of the environment obtained in parallel.

The contributions of this paper are: first, we propose this problem as worthy of renewed attention due to its potential utility. Second, we provide a simple novel model for human motion in a constrained environment. Third, we provide a complete and freely available modular implementation in ROS, with the major components of obstacle mapper, human tracker, human motion model, robot motion planner and robot motion controller easily replaceable. We report on the performance of the robot in real-world experiments.

## II. RELATED WORK

Person tracking has been widely studied. Munaro et al [5] propose a multiple target tracker system using Kinect RGB-D detections. They use a depth based sub-clustering method to track people even near walls. Wojke [6] successfully tracked a user/operator using Bayes-optimal state estimator using

\*Autonomy Lab, School of Computing Science, Simon Fraser University, Canada. {pnikdel, rakeshs, vaughan}@sfu.ca

†These authors contributed equally to this work.

<sup>1</sup>E.g. 'Canvas Technology' <http://canvas.technology> has an autonomous push-cart robot *without* following in front

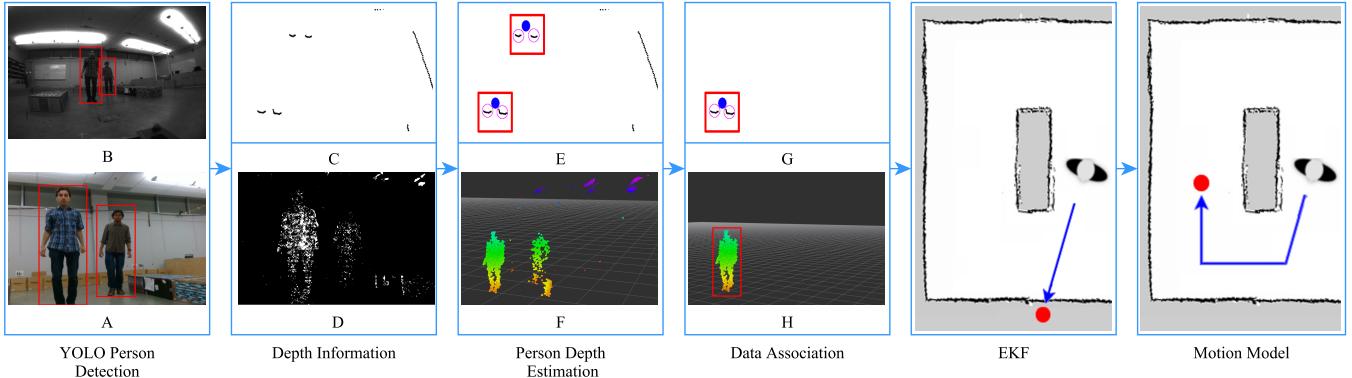


Fig. 2: Two alternate system processing pipelines to track the user. Top row: Fisheye camera and laser rangefinder; RGB-D camera. [A] Fisheye image with person ROIs, [B] RGB image with person ROIs, [C] Raw laser scan, [D] Raw depth image, [E] Leg detections on laser scan segmented based on person ROIs (blue circles represent estimated positions of people), [F] Point cloud of people obtained from depth image after segmentation based on ROIs and median filter, [G] and [H] Selected person after data association for wide FOV detection mode and narrow FOV detection mode

RGB-D detections and image-based classification. Koide and Miura [7] developed a system that uses height and gait information to achieve person tracking and identification.

Takano et al [8] propose an approach to predict human motion using symbolic inference. Ziebart et al. [9] use a Markov Decision Process (MDP) and maximum entropy learning to obtain a probabilistic model of pedestrian trajectories assuming purposeful behavior. In a similar work, Kuderer et al [10] remove the underlying MDP assumption and reason about continuous trajectory as opposed to discrete states. Our model is simpler than the aforementioned ones as we assume that a person predominantly walks in the same direction.

#### A. Following behind

Most of the previous work on person following has involved following a user from behind. In [11] Leigh et al present a human-centered tracking framework which classifies laser data as human or not human. The detected person positions are tracked using a Kalman Filter, then they apply separate PID controllers to obtain the angular and linear velocities of the robot. An interesting resource-limited example is Yao et al [12], where the Georgia Tech Miniature Autonomous Blimp detects and follows a person using a monocular camera. They use a Haar face detector and a KLT feature tracker to track the user. In a recent work, Sun et al [13] present a following behind the leader behavior using a social forces model.

#### B. Following ahead

There are very few papers that address the problem of following in front of a user. Cifuentes et al [14] propose an approach based on a human gait model that uses a wearable Inertial Measurement Unit (IMU) for estimating orientation. Ho et al [2] calculate human orientation using a Kalman filter with a nonholonomic human model for estimating human linear and angular velocities, while a special-purpose robot motion controller aims to align the human-robot poses such that the robot follows from the front. Eui-Jung et al [4]

present a holonomic motion model for tracking a human while staying ahead. In [15] Tominaga et al present another front-following system using simple visual servoing that tries to keep a person (marked with an AR tag) in the center of the robot's view. The heading of the person is not considered, and the robot can easily lose the person at sharp turns. Recent work by Moustris et al [3] describes a front-following model that uses a modified dynamic window planner without considering the current heading of the person, which we suggest is important information when predicting motion relatively far into the future. Their method is extended in [16], where they assume that a person's orientation can be estimated by how off-center (i.e., to the left or right) the user is from the middle of the robot's field of view (FOV), which is not valid at T-junction turns for example.

All of the previous works mentioned for following ahead of the leader, with the exception of [3], [16], assume that the environment is obstacle free. Obstacles make the problem much more challenging as the human may often be occluded from the robot's sensor, for example as the robot turns a corner ahead of the human. The focus of our work is to address this issue. Our approach assumes that the human will often be out of view, so it uses its recent estimate of the human's position combined with a local model of the world, and a model of the human's speed and direction of motion, to figure out where the human is likely to go *or has gone*, when they disappear from view. This mechanism also provides robust recovery when the robot guesses a turn incorrectly.

### III. SYSTEM OVERVIEW

The system is implemented in ROS [17] on a Pioneer P3-DX mobile robot. We implemented two alternate sensor modes for detecting humans: one with a narrow FOV and the other with a wide FOV. The former uses an RGB and depth image pair as input while the latter uses a monocular image from a fisheye camera combined with a 2D laser range scan. All the images are obtained from Intel® RealSense™ ZR300 development kit and laser scans are obtained from Hokuyo

URG-04LX Laser Range Finder (LRF). Both the sensors are back-facing with respect to the robot. These positions are fed to our data association module which fuses the data to give a measurement of the position of the person to be tracked relative to the robot.

An EKF is used to estimate the speed and direction of the person. This information is used to make a naive prediction of the person's future pose without taking the environment into account. This prediction is then updated using our motion model based on the person's heading and a map of the environment.

An occupancy grid map is built incrementally during runtime as the robot navigates. The map is used in robot path planning as well as human motion prediction. The robot performs Simultaneous Localization and Mapping (SLAM) using the well-known GMap ROS implementation of the grid mapping technique [18], using odometry and range data from a front-facing Hokuyo LRF. We use the ROS navigation module<sup>2</sup> to navigate the robot to goal positions. The trajectory planning module is based on [19].

The processing pipeline is summarized in Fig. 2 and the main components are described below. The system was developed using the Stage robot simulator [20] before real-world testing. In the simulation, we abstracted the details of the person detection using Stage's color-blob finder.

#### IV. PERSON DETECTION

We use YOLOv2 [21] on the narrow RGB or fisheye grayscale image from the RealSense device to get a Region of Interest (ROI) corresponding to the person. We use the ROIs to segment the range data from the calibrated depth image or laser scan that corresponds to people, as follows.

##### A. RGB-D camera, narrow field of view mode

Using the depth image, we compute the set of 3D points that correspond to the detected person's ROI. We reject background points inside the ROI using depth thresholding. The position of the person (in the image plane) is estimated as the center of the bounding box of the resulting point cloud while the depth is estimated as the median depth of the cloud.

##### B. Fisheye camera + LRF, Wide field of view mode

Person ROIs in fisheye images are used to filter the corresponding scans from the LRF. Human leg-detector [22] is used to find the distance and bearing to candidate humans; if a candidate is detected at the same bearing as the camera ROI, we have detected a candidate interaction partner.

Because the FOV of the LRF ( $240^\circ$ ) is higher than that of the fisheye camera ( $166^\circ$ ), the person can still be tracked using raw laser data, as explained in Section V. Leg detection on raw range scans without tracking leads to numerous false positives, but our data association maintains reliable tracking of the person we have previously seen with the camera.

The leg-based person detector fails to detect two valid legs for a person at large distances, often finding just a single leg which can be used to track a previously-identified person.

<sup>2</sup><http://ros.org/wiki/navigation>

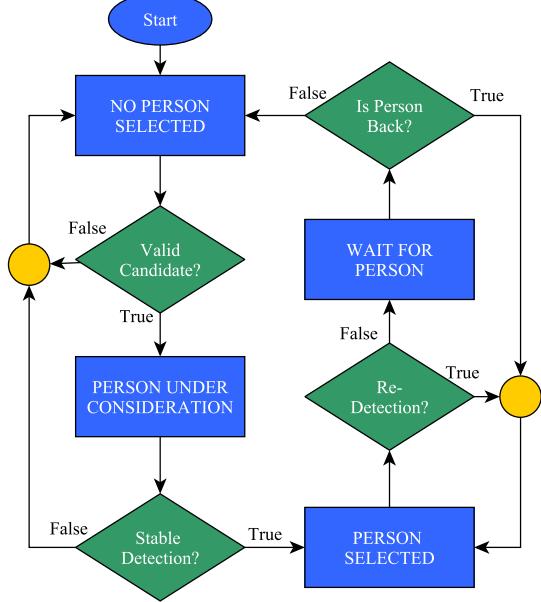


Fig. 3: Data Association Flowchart

We use the leg-only detection only when we have already selected the person to track (Section V).

#### V. DATA ASSOCIATION

To ensure that we correctly track the person interacting with the robot, we designed a tracking state machine using Nearest Neighbor (NN) data association. This reliably tracks the user when there are multiple humans in the sensor FOV or the user goes out of the FOV, or when the detectors temporarily fail to detect the valid candidate still in the FOV.

We have four tracking states:

- *NO\_PERSON\_SELECTED*
- *PERSON\_UNDER\_CONSIDERATION*
- *PERSON\_SELECTED*
- *WAITING\_FOR\_PERSON*

The starting state is *NO\_PERSON\_SELECTED*. When human(s) are detected it chooses the person closest to the robot among the candidates that are closer than a fixed initiation distance. We set this distance to 2.5m for our experiments. The state is then updated to *PERSON\_UNDER\_CONSIDERATION*. In the following  $N$  detections, if a person is present in close proximity to the position of the person in previous iteration the state is changed to *PERSON\_SELECTED*, otherwise the state goes back to *NO\_PERSON\_SELECTED*. If the selected person is no longer detected, the state changes to *WAITING\_FOR\_PERSON* for a fixed time period before going back to *NO\_PERSON\_SELECTED*. Fig. 3 illustrates the state machine for data association.

The person position is fed to the EKF only if the tracking state is *PERSON\_SELECTED*. Note that our implementation does not require the person to always be within a Human Interaction Zone [3], [14] to be tracked, but requires the person to be close enough to the robot to initiate the interaction.

## VI. PERSON STATE ESTIMATION

### A. Person State representation

Following the nonholonomic model of human motion as proposed by [23], we represent the state of the person at time  $t$  with position  $(x_t, y_t)$ , walking direction  $\theta_t$  and velocity along the walking direction  $v_t$ . The states are relative to the map frame of the SLAM system. Additionally, we maintain the position of the person in the previous time step  $t - 1$ . The state transition from time  $t$  to  $t + 1$  is then given by:

$$\begin{aligned} x_{t+1} &= x_t + v \cos(\theta) dt \\ y_{t+1} &= y_t + v \sin(\theta) dt \\ v_{t+1} &= (1 - \alpha_v)v_t + \alpha_v \| (x_t, y_t) - (x_{t-1}, y_{t-1}) \| / dt \\ \theta_{t+1} &= (1 - \alpha_\theta)\theta_t + \alpha_\theta \text{atan2}(y_t - y_{t-1}, x_t - x_{t-1}) \end{aligned} \quad (1)$$

where  $dt$  is the interval between two consecutive times.

We model velocity and direction as smoothly changing quantities by implementing updates in the form of an Infinite Impulse Response (IIR) filter<sup>3</sup>. The coefficients  $\alpha_v$  and  $\alpha_\theta$  of IIR filter lie in the range  $[0, 1]$  and represent how fast the values are expected to change (higher values corresponding to higher change). We found by trial and error the values around 0.5 give us the best results.

For the measurement model, we assume that the position  $(x_t, y_t)$  of the person in the global frame is observable. Our actual measurement consists of position with respect to the robot, but given that we have an estimate of the robot position and orientation from SLAM, we can find the position of the person in global frame. We propagate the uncertainty in robot pose to our measurement to make state estimation more robust, which will be discussed in Section VI-B.

[2] use a similar model in which the state of the person is computed from the robot's local frame and transform the reference frame using odometry readings at every time step. This is likely to introduce drift in estimates over time with respect to a global frame. Although the drift in the coordinate frame itself does not adversely affect the person following behavior, using just the odometry is likely to give noisy estimates even over shorter time intervals. Our approach has the advantage of potentially better state estimates at the cost of additional sensing and localization/mapping.

We use an EKF for state estimation. Because our measurements are based on the fairly accurate absolute robot pose from SLAM and relative estimates of human position, we discount the linearization error inherent to the EKF. Improved variants of the nonlinear Kalman Filter like the Unscented Kalman Filter [24] could be substituted to improve the state estimation.

### B. Uncertainty Propagation

Assuming a zero-mean Gaussian distribution of noise on input  $\mathbf{X}$  with covariance  $\Sigma_X$ , the noise propagated to the function  $f(\mathbf{X})$  can be approximated to first order by

$$\Sigma_f \approx \mathbf{J}_f \Sigma_X \mathbf{J}_f^T \quad (2)$$

<sup>3</sup>For averaging angles, we use the orientation of weighted sum of unit vectors along the angles to avoid angle wraparound discrepancy

where  $\mathbf{J}_f$  is the Jacobian of  $f(\mathbf{X})$ .

Let  $r$  be the distance estimate and  $\phi$  the bearing angle estimate of the person relative to the robot, which are the actual measurements we receive from our sensors. Let  $(x_R, y_R)$  and  $\theta_R$  be the position and orientation of the robot respectively, obtained from SLAM. The measurement of our human position  $(x, y)$  is obtained as:

$$\begin{aligned} x &= -r \cos(\theta_R) \cos(\phi) - r \sin(\theta_R) \sin(\phi) + x_R \\ y &= -r \cos(\theta_R) \sin(\phi) + r \sin(\theta_R) \cos(\phi) + y_R \end{aligned} \quad (3)$$

Using the uncertainty propagation formulation of Eq. (2), we can have estimates of noise in our virtual measurement  $(x, y)$  due to independent components  $r$ ,  $\phi$ ,  $(x_R, y_R)$  and  $\theta_R$ . This implementation allows us to utilize the difference in the noise in measurement from the two modes of person detection we currently have and also uncertainty in the robot pose while still having a simple kinematic model.

With the estimate of walking direction, we set our naive goal to be a fixed follow-ahead distance in front of the person in that direction. To ensure that the direction estimate is reliable, we take the goal position as valid only when the estimated velocity is above a threshold and estimated variance of orientation is below a threshold.

## VII. PERSON MOTION MODEL

In open environments without obstacles, the estimated walking direction of the person from the EKF can be used trivially to set a navigation goal for the robot. However, in bounded environments with walls and other structures, we need a better predictor of the person's path. We propose a simple geometric model of person motion that takes into account both the current heading of the person and the environment, assuming piecewise linearity of obstacles.

Consider that a person is at point  $P$  and the naive predicted position at  $G$ , with  $O_1$  and  $O_2$  being endpoints of an obstacle (Fig. 4). Our model predicts that the person is going to change direction at point  $P'$ , which is a fixed distance from the obstacle, and walks along one of two directions,  $\mathbf{D}_1$  and  $\mathbf{D}_2$ , parallel to the obstacle. The direction is chosen based on the current heading of the person: the person will walk in the direction that minimizes the change in heading ( $\mathbf{D}_2$  in this case), i.e. the smaller of the angles between vectors  $\overrightarrow{PG}$  and  $\mathbf{D}_1$  and  $\mathbf{D}_2$ . This process is repeated until the distance covered is equal to the follow-ahead distance and the last point serves as the final goal position for the robot.

If the vector  $\overrightarrow{P'G'}$  is exactly perpendicular to the obstacle line (or approximately perpendicular in the presence of sensor noise), the situation is undecidable. This can be remedied for the updates to goal positions other than the original goal  $G$  by choosing the direction that has a smaller angle with  $\overrightarrow{PG}$  as it represents the direction that is closer to the person's heading. Also, we avoid choosing the direction that immediately leads to obstacles which is particularly helpful for sharp corners.

Our model is applied using the constantly-updating occupancy grid map from the SLAM system. We perform morphological image operations – dilation followed by erosion

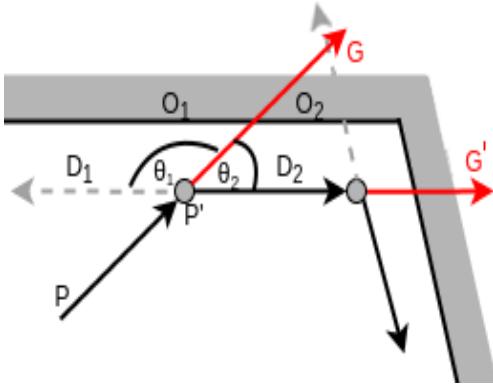


Fig. 4: User motion model illustrated.  $P$  is the person position, with the arrow showing the heading.  $G$  is the naively predicted future position extrapolated from current velocity. The map-aware motion model predicts that the person turns at  $P'$  to be parallel to the wall (line segment  $O_1, O_2$ ) along  $D_2$ , and turns again in the corner.

– to fill holes in the obstacles. Then we use ray-casting to detect obstacles between  $P$  and  $G$ . If obstacles are detected, we do a Breadth First Search (BFS) for a limited distance starting from the known obstacle point. We find the best line that fit these points using a Hough transform.

If the person is no longer in the robot’s FOV, which is the typical case in sharp turns, the robot simply goes to the last goal; before the robot gets there the person usually reappears in the sensor FOV.

### VIII. EXPERIMENT

We conduct our experiments in two settings. In the first setting, the environment is an open rectangular area with flat walls. The robot has to follow in front of a person who walks around the room along the walls. The user makes two laps, turning eight corners. We refer to this as the ‘Easy’ setting. In our second setting, the same perimeter contains two rectangular obstacles such that there is space for walking between the obstacles and between obstacles and walls: topologically a figure eight. The user passes an intersection four times and turns at corners 6 times, making both right and left turns. We refer to this as the ‘Difficult’ environment. Our environments are depicted in Figure 5 along with the trajectories. The human subject for our experiments is one of the authors. The subject tries to repeat the same trajectory in all the experiments for a particular setting, guided by marks on the floor that are invisible to the robot. The trajectories of robot and human were recorded using an external Vicon motion capture system. The size of the test arena was limited by the usable Vicon area.

Our performance error metric is the percentage of total trial time the user had to wait for the robot to come in front and resume correct following-ahead behavior. A robot that predicts the user’s motion perfectly (impossible in the ‘Difficult’ setting) and always stays ahead would score 0%. A robot that stayed still or followed behind would score 100%. These evaluations are done at the corners and turnings. We run 13 trials for each of the settings. The results (minimum,

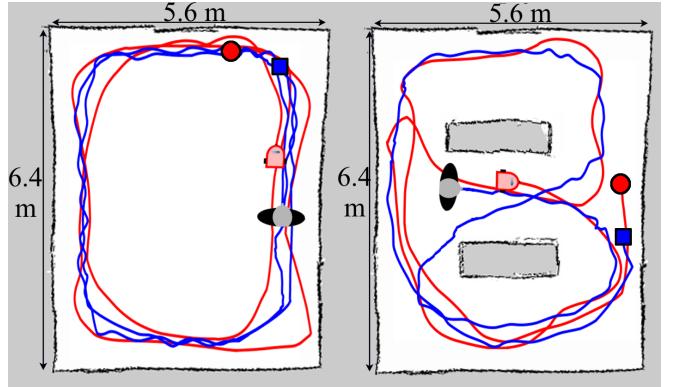


Fig. 5: Experiment settings, Left: ‘Easy’, Right: ‘Difficult’. The red line is the trajectory of the robot and blue that of the user. Person and robot markers represent their respective start positions. The blue square is the final position of the person and the red circle is the final position of robot

median and maximum waiting time) are summarized in Table I. The evaluations were done using the wide FOV mode of detection (Section IV-B).

Setting	Minimum	Median	Maximum
Easy	2.4	5.0	12.0
Difficult	14.1	21.1	32.3

TABLE I: User waiting time each setting, as a percentage of total trial time, over 13 repeated trials. This is an error measure of the total time a user had to stop walking and wait for the robot to resume correct following-ahead behavior after making a prediction mistake.

### IX. DISCUSSION

The ‘Difficult’ setting has a significant number of sharp turns located close together. Either the person or the robot (or both) are turning almost all the time. When the user starts to turn down the middle passage the robot is usually already past the turning, having incorrectly predicted the user’s trajectory. Once the human is seen to start turning, the robot quickly updates its user model and travels back to be in front of the person. Since the robot is limited to safe speeds close to people, the human has to wait for it to catch up. Fig. 8 shows this behavior. This is the cause of the long waiting times for this setting. The robot is usually able to recover after overshooting and moves ahead of the user again. There were 2 cases in the ‘Easy’ setting and 4 cases in the ‘Difficult’ setting (not included in our analysis) where the robot made an erroneous prediction (causing it to turn in the wrong direction), and the user then falls out of the robot’s FOV. As a result, the robot strayed away from the user’s intended path. Additional error checks on the predicted goal can help remedy this problem which will be explored in future works.

Fig. 6 and 7 visualizes the robot’s performance in one trial for each of the settings. The top plot shows samples of the absolute positions of the person and robot at evenly-spaced

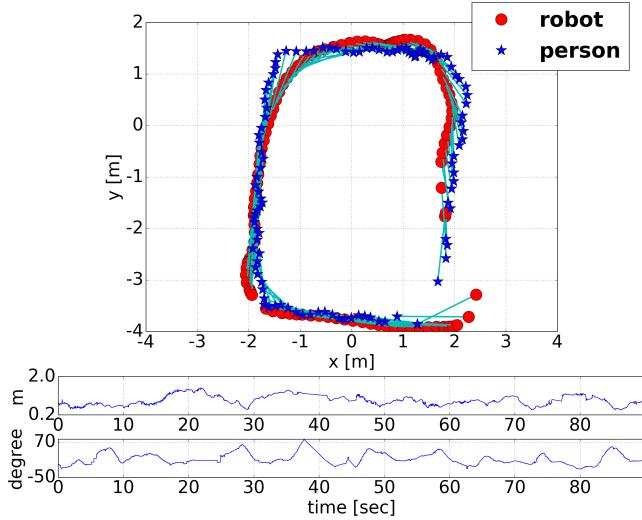


Fig. 6: Evaluation of ‘Easy’ setting. Top: cyan lines show the corresponding simultaneous positions of robot (red dots) and person (blue stars) over time. Middle: Distance between the robot and person over time. Bottom: Bearing angle of the person with respect to the robot over time.

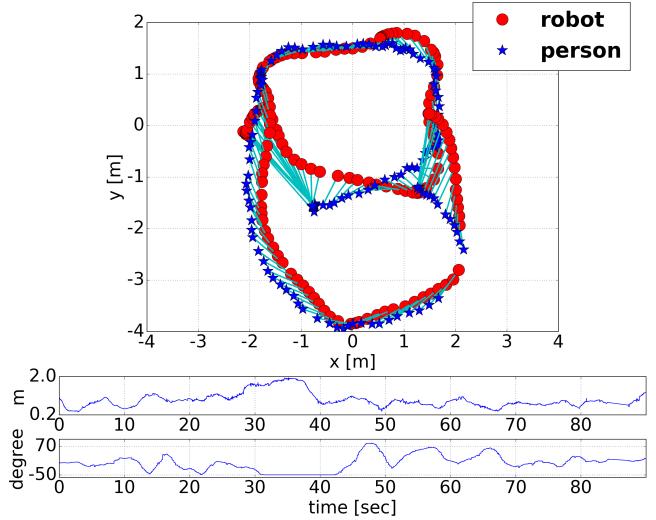


Fig. 7: Evaluation of ‘Difficult’ setting. Top: cyan lines show the corresponding simultaneous positions of robot (red dots) and person (blue stars) over time. Middle: Distance between the robot and person over time. Bottom: Bearing angle of the person with respect to the robot over time (measurement between 30 to 40 seconds is missing data due to the user being out of the robot’s FOV).

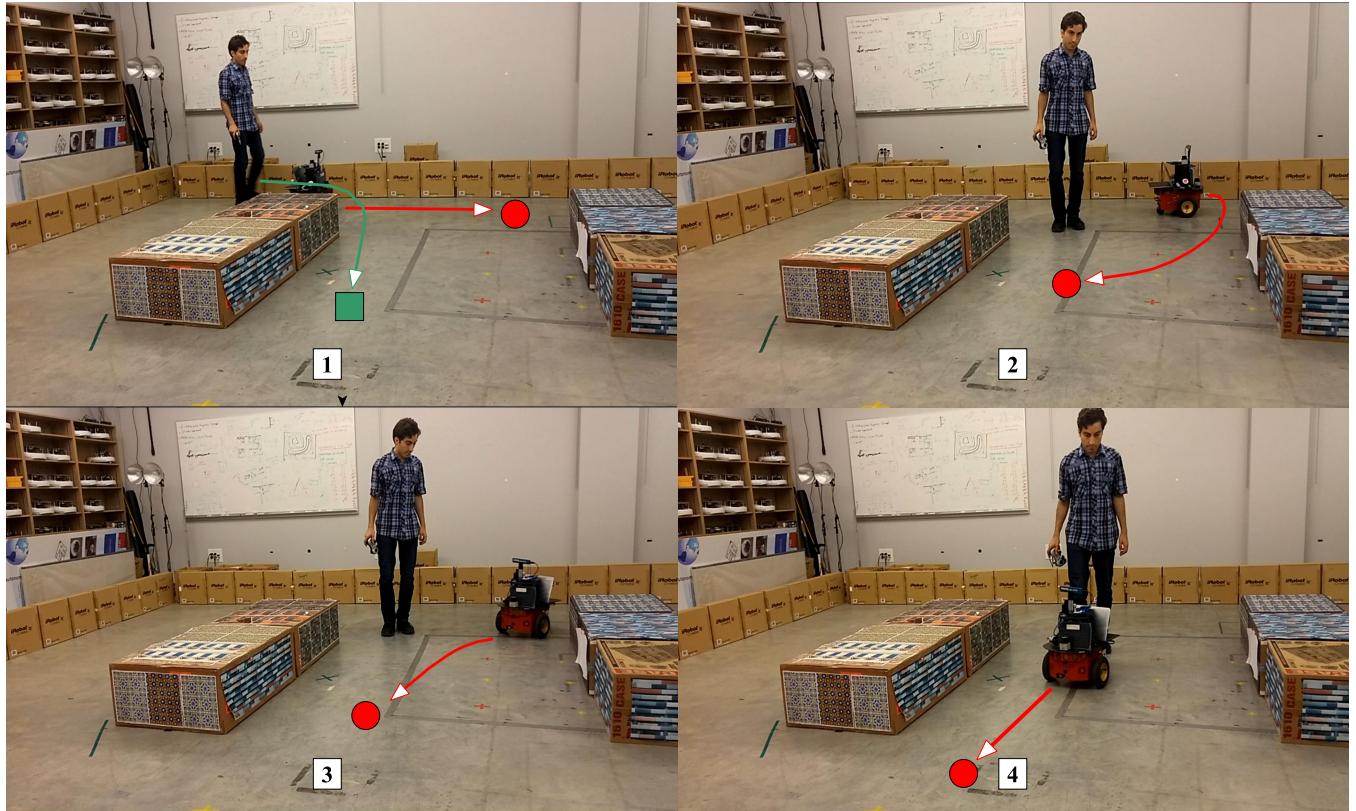


Fig. 8: User waiting due to incorrect motion prediction and recovery (red circle: goal point estimated by motion model. Green square: actual goal position of the user. [1] Robot is following-ahead just before the human starts to turn to his right. [2] Robot perceives new receives updated goal position along the new walking direction of the person [3] Subject pauses briefly, waiting for the robot to get in front [4] Front-following resumes.

time intervals, joined by a line segment (only one loop is shown for clarity). The middle plot shows the distance in meters between the robot and person while the bottom plot shows the bearing angle of the person with respect to the robot. The trajectory and distances were obtained from a Vicon motion capture system while the bearing information was obtained from the estimated relative positions.

The system has consistent performance except in cases where the off-the-shelf path planner fails to navigate to a valid goal position. We conducted additional experiments on a real building hallway with two dogleg intersections using both wide and narrow FOV modes of detection. No ground truth data from a motion capture system were available, but the trials can be seen in supplementary videos<sup>4</sup>.

## X. CONCLUSION

We propose a “following in front of the leader” robot behavior. Our implementation improves on previous work by featuring a motion model which predicts the trajectory of the person by reasoning about walking direction in the context of the immediate surroundings. We use state-of-the-art CNN-based object detection, and all our code is freely available online as ROS modules. We proposed a simple error metric for this behavior and evaluated our system in easy and hard settings. The results are qualitatively good, especially for the easy setting. A user study would be required to make a formal claim, but informally we believe our following behavior feels natural and easy in our experiments.

In future work, we aim to integrate gesture recognition and other cues to resolve ambiguity in user intention, which can happen when the walking direction is normal to the obstacle line. There is also much room for improvement in our motion model – for instance, we could remove the piecewise linearity of obstacle constraint. The user’s heading could possibly be anticipated from the gaze direction like in the work of [25].

## XI. ACKNOWLEDGEMENTS

Supported by NSERC Canadian Field Robotics Network.

## REFERENCES

- [1] D. Wooden, M. Malchano, K. Blankepoor, A. Howard, A. A. Rizzi, and M. Raibert, “Autonomous navigation for BigDog,” in *Robotics and Automation, 2010 IEEE Int. Conf.* IEEE, May 2010, pp. 4736–4741.
- [2] D. M. Ho, J. S. Hu, and J. J. Wang, “Behavior control of the mobile robot for accompanying in front of a human,” in *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME Int. Conf.* IEEE, July 2012, pp. 377–382.
- [3] G. P. Moustris and C. S. Tzafestas, “Assistive front-following control of an intelligent robotic rollator based on a modified dynamic window planner,” in *Biomedical Robotics and Biomechatronics (BioRob), 2016 6th IEEE Int. Conf.* IEEE, June 2016, pp. 588–593.
- [4] E. J. Jung, B. J. Yi, and S. Yuta, “Control algorithms for a mobile robot tracking a human in front,” in *Intelligent Robots and Systems, 2012 IEEE/RSJ Int. Conf.* IEEE, Oct 2012, pp. 2411–2416.
- [5] M. Munaro and E. Menegatti, “Fast RGB-D people tracking for service robots,” *Autonomous Robots*, vol. 37, no. 3, pp. 227–242, 2014.
- [6] N. Wojke, R. Memmesheimer, and D. Paulus, “Joint operator detection and tracking for person following from mobile platforms,” in *Information Fusion (Fusion), 2017 20th Int. Conf.* IEEE, 2017, pp. 1–8.
- [7] K. Koide and J. Miura, “Identification of a specific person using color, height, and gait features for a person following robot,” *Robotics and Autonomous Systems*, vol. 84, pp. 76 – 87, 2016. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889015303225>
- [8] W. Takano, H. Imagawa, and Y. Nakamura, “Prediction of human behaviors in the future through symbolic inference,” in *Robotics and Automation (ICRA), 2011 IEEE Int. Conf.* IEEE, 2011, pp. 1970–1975.
- [9] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, “Planning-based prediction for pedestrians,” in *Intelligent Robots and Systems (IROS), Int. Conf.* IEEE, 2009, pp. 3931–3936.
- [10] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, “Feature-Based Prediction of Trajectories for Socially Compliant Navigation,” in *Proceedings of Robotics: Science and Systems (RSS)*, Sydney, Australia, Jul. 2012.
- [11] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, “Person tracking and following with 2D laser scanners,” in *Robotics and Automation (ICRA), 2015 IEEE Int. Conf.* IEEE, 2015, pp. 726–733.
- [12] N. Yao, E. Anaya, Q. Tao, S. Cho, H. Zheng, and F. Zhang, “Monocular vision-based human following on miniature robotic blimp,” in *Robotics and Automation (ICRA), 2017 IEEE Int. Conf.* IEEE, 2017, pp. 3244–3249.
- [13] Y. Sun, L. Sun, and J. Liu, “Human comfort following behavior for service robots,” in *Robotics and Biomimetics (ROBIO), 2016 IEEE Int. Conf.* IEEE, 2016, pp. 649–654.
- [14] C. A. Cifuentes, A. Frizera, R. Carelli, and T. Bastos, “Human–robot interaction based on wearable IMU sensor and laser range finder,” *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1425–1439, 2014.
- [15] J. Tominaga, K. Kawauchi, and J. Rekimoto, “Around me: a system with an escort robot providing a sports player’s self-images,” in *Proceedings of the 5th Augmented Human Int. Conf.* ACM, 2014, p. 43.
- [16] G. P. Moustris and C. S. Tzafestas, “Intention-based front-following control for an intelligent robotic rollator in indoor environments,” in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, Dec 2016, pp. 1–7.
- [17] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, vol. 3, no. 3.2. Kobe, 2009, p. 5.
- [18] G. Grisetti, C. Stachniss, and W. Burgard, “Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb 2007.
- [19] B. P. Gerkey and K. Konolige, “Planning and control in unstructured terrain,” in *In Workshop on Path Planning on Costmaps, Proceedings of the IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2008.
- [20] R. Vaughan, “Massively Multi-Robot Simulations in Stage,” *Swarm Intelligence*, vol. 2, no. 2-4, pp. 189–208, December 2008.
- [21] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” *arXiv preprint arXiv:1612.08242*, 2016.
- [22] D. V. Lu and W. D. Smart, “Towards more efficient navigation for robots and humans,” in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ Int. Conf. On.* IEEE, 2013, pp. 1707–1713.
- [23] G. Arechavaleta, J.-P. Laumond, H. Hicheur, and A. Berthoz, “On the nonholonomic nature of human locomotion,” *Autonomous Robots*, vol. 25, no. 1, pp. 25–35, Aug 2008. [Online]. Available: <https://doi.org/10.1007/s10514-007-9075-2>
- [24] E. A. Wan and R. V. D. Merwe, “The unscented Kalman filter for nonlinear estimation,” in *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373)*, 2000, pp. 153–158.
- [25] L. Zhang and R. Vaughan, “Optimal robot selection by gaze direction in multi-human multi-robot interaction,” in *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ Int. Conf.* IEEE, 2016, pp. 5077–5083.

<sup>4</sup><http://autonomylab.github.io/following-ahead>