

Optimal Trajectories for Autonomous Human-Following Carts with Gesture-Based Contactless Positioning Suggestions

Merrill Edmonds, Tarik Yigit, Victoria Hong, Faiza Sikandar and Jingang Yi

Abstract—Human-following autonomous robots can help human-oriented tasks in many fields. In this paper, we focus on replacing traditional shopping carts with human-following robots as a way to aid public health measures under events such as a global pandemic, and to reduce shopping effort for consumers. Our framework mirrors efforts made in other domains to introduce human-following robots into human-oriented tasks, and consists of two major building blocks: human pose estimation from 3D data, and collision-free navigation with gesture-based positioning suggestions. RGB-D data is used to estimate human poses and extract pointing gestures. The pose and pointing direction are sent to the controller as suggestions for where the cart should be positioned. The cart then either selects to move as close to the positioning suggestion as possible, or follows the human if they move too far away. The multi-cart system uses an MPC-based optimization method to generate multiple collision-free paths that satisfy as many positioning suggestions as possible. We validate our pathing method with both large-scale multi-cart simulations in ROS/Gazebo, and we further validate the human-following block with in-lab tests using a group of omni-directional robots.

I. INTRODUCTION

Human-following autonomous robots are useful tools for increasing human productivity and lowering physical barriers for certain human-centric tasks. Unlike industrial robots, human-following robots are designed to work directly with humans and must contend with additional human-centric performance metrics such as comfort, adaptability and safety [1]. The reward for contending with these issues is, naturally, the enrichment of the human experience in many domains from manufacturing to personal in-home robotics, as well as many consumer-facing use cases in commercial settings. For example, the recent global pandemic has underlined the need for more robust human-following robotics in supermarkets and shopping malls, where considerable resources are spent on sanitizing high-touch surfaces such as shopping cart handles. Replacing traditional shopping carts with human-following shopping carts would both aid public health measures and reduce shopping effort for consumers.

While there has been some research towards producing an autonomous shopping cart [2] and some attempts were made to commercialize smart carts, the majority of relevant research focuses on other domains where human-following robots are

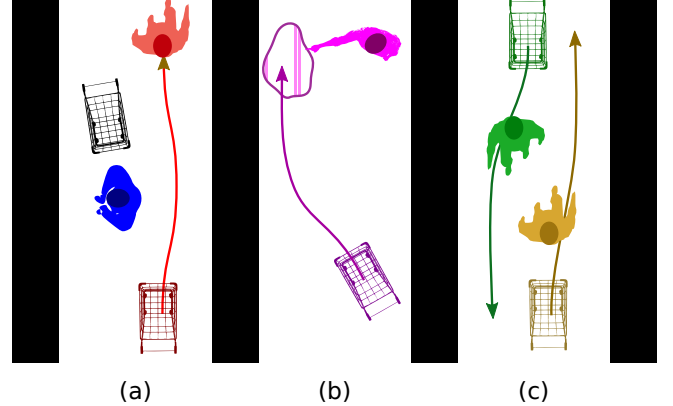


Fig. 1. (a) The human-following shopping cart navigates the aisles using on-board cameras and RGB-D sensors, which provide mapping information for collision avoidance. (b) The user can use gesture-based suggestions, e.g. pointing, to assign target positions for the shopping cart. (c) Shopping carts communicate locally and plan moves cooperatively as a multi-robot team.

used [3]. These works commonly focus on computer vision and artificial intelligence methods to track humans via body, face and hand keypoints, and combine the tracking results with more traditional planning methods to generate human-following trajectories.

The use of RGB and RGB-D cameras are quite common for human tracking and activity recognition [4], [5]. In [6], [7], depth data from Kinect sensors are used to identify the walking style of the user, but the method can fail if someone else has a similar walking style [8] [9]. Color features are used in [10] instead, but the hue-saturation-value histogram is prone to fail due to occlusions or severe illumination changes. The leg-based person detector in [11] similarly fails to detect two valid legs for a person at large distances, often finding just a single leg which can still be used to track a previously-identified person. Alternatively, binocular cameras have also been used for both indoor and outdoor environments. Even with depth information and trackers with rich image features there were still issues with the positioning precision. [12].

Map-based techniques have been used in order to follow the target [13]. Many methods have worked in robotic friendly environments, however, when it comes to public areas with many people, there are difficulties in following the target [14]. The work in [6] was able to successfully avoid both static and dynamic human beings. This method studies not only human-robot interaction, but also robust human-human interaction. However, there are limitations as additional complex interactions occur causing delays [15]. Since losing

*This work was partially supported by US National Science Foundation under IIS-1426828.

M. Edmonds, T. Yigit, F. Sikandar and J. Yi are with the Department of Mechanical and Aerospace Engineering, Rutgers University, Piscataway, NJ 08854 USA (email: merrill.edmonds@rutgers.edu; tarik.yigit@rutgers.edu; faiza.sikandar@rutgers.edu; jgyi@rutgers.edu)

V. Hong is with the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, NJ 08854 USA (email: victoria.hong@rutgers.edu)

the target is a possibility, studies on path estimation and target reidentification methods like variational Bayesian linear regression and Siamese network have been explored [16]. Predictions are based on the user's current walking speed and the preset path [17]. Given the information needed, a standard Adaptive Monte Carlo Localization (AMCL) self-localization system gives pose estimates [18] [19]. Path planning also has a possibility of failure due to frequent sudden changes in human trajectories or more complex dynamics [18]. Improvement of prediction and collision avoidance are among current research, but have not been tested in a very dynamic environment [20] [21] [7].

One of the many limitations of path planning is to maintain a connected network throughout the exploration process. [22] Physical limitations of following the target include examples such as not being able to detect advertisement boards placed above 50 centimeters and attached to the shelves and do u-turns in store aisles [2]. Therefore, gesture recognition based studies were explored [23].

In this paper, we propose a human-following algorithm that is supplemented by gesture-based contactless positioning suggestions. The design solves many of the issues present in previous methods by combining recent advances in computer-vision-based human tracking with multirobot planning and control. We also outline a novel approach to determining human-aware waiting locations for autonomous shopping carts both with and without gesture-based suggestions. We validate our cooperative multi-cart planner by simulating a supermarket setting. We further demonstrate our human-following algorithm with gesture-based contactless positioning suggestions using an omni-directional robot [24]. Our main contribution is therefore a framework and related algorithms for multi-cart human tracking and motion planning, concurrently released as a ROS [25] package on Github¹.

The rest of the paper is structured as follows. Section II outlines the mathematical formulation for the human-following shopping cart problem with positioning suggestions and details our solution method. Section III details the experimental setup for multi-cart simulations and in-lab gesture tracking experiments. Section IV discusses the results of our experiments. Finally, Section V presents conclusions and future work.

II. HUMAN-FOLLOWING SHOPPING CART WITH POSITIONING SUGGESTIONS

Let \mathbf{R} be the set N of cooperative human-following shopping carts. We denote the pose of robot $r_i \in \mathbf{R}$, $i = 1, \dots, N$ at time t as $\mathbf{x}_i(t) = [x_i(t), y_i(t), \theta_i(t)]^T$, $\mathbf{x}_i(t) \in \mathcal{X}_i$, where \mathcal{X}_i is the feasible pose set for r_i , with dynamics given as $[\dot{\mathbf{x}}_i^T(t), \ddot{\mathbf{x}}_i^T(t)]^T = \mathbf{f}(\mathbf{x}_i(t), \dot{\mathbf{x}}_i^T(t), \mathbf{u}_i(t))$, where $\mathbf{u}_i(t)$ are the control inputs for r_i at time t , and the pose goal for robot r_i as $\mathbf{x}_i^{(g)} = [x_i^{(g)}, y_i^{(g)}, \theta_i^{(g)}]^T$, $\mathbf{x}^{(g)} \in \mathcal{G}_i$, where $\mathcal{G}_i \subseteq \mathcal{X}_i$ is a polytope that captures goal tolerances. The pose and gait of the human that robot r_i needs to follow at time t is denoted as $\mathbf{h}_i(t)$, and is defined as the concatenation of the pose vector

and the gait feature vector $\mathbf{h}_i(t) = [\boldsymbol{\eta}_{h,i}(t)^T, \boldsymbol{\phi}_{h,i}(t)^T]^T$ if the robot is following a human, and is undefined otherwise.

The optimal cart trajectories $\mathbf{x}_i(t)^*$ are calculated at each time step by solving a constrained multirobot trajectory optimization problem,

$$\mathbf{x}_i(t)^* = \underset{i=1, \dots, N}{\operatorname{argmin}} \sum_{i=1}^N \left[\int_t^{t+T} \psi(\mathbf{x}_i(\tau), \mathbf{u}_i(\tau)) d\tau \right] \quad (1)$$

$$\text{s.t. } \psi(\mathbf{x}_i(t), \mathbf{u}_i(t)) = p_i, \quad (2)$$

$$\dot{\mathbf{x}}_i(t) = \mathbf{f}(\mathbf{x}_i(t), \mathbf{u}_i(t)), \quad (3)$$

$$\mathbf{x}_i(t) \in \mathcal{X}_i, \quad (4)$$

$$\mathbf{x}_i(t+T) \in \mathcal{G}_i, \quad (5)$$

$$\mathbf{u}_i(t) \in \mathcal{U}_i, \quad (6)$$

where T is free and p_i is the priority of the i th robot such that a larger p_i indicates higher priority. The optimal trajectories guide the carts toward $\mathbf{x}_i^{(g)}$, selected by optimizing the desired pose $\mathbf{x}_i^{(d)}$ described in (7) using the optimization in (9), so that each robot either follows $\mathbf{h}_i(t)$ by moving to a position that trails the human by a distance of δ_f along its recent trajectory, or moves to a waiting position that is within a distance of δ_w from a pose suggested by the human via gestures, or approaches the human at a distance of δ_a to allow them to drop off any items they are holding.

The desired pose $\mathbf{x}_i^{(d)}$ is determined at each time step by a finite state machine with *Following*, *Waiting* and *Approaching* states (with transition conditions shown in Fig. 2) as

$$\mathbf{x}_i^{(d)} = \begin{cases} \mathbf{x}_i^{(w)}, & \text{robot_state=Waiting} \\ \mathbf{x}_i^{(f)}, & \text{robot_state=Following} \\ \mathbf{x}_i^{(a)}, & \text{robot_state=Approaching} \end{cases} \quad (7)$$

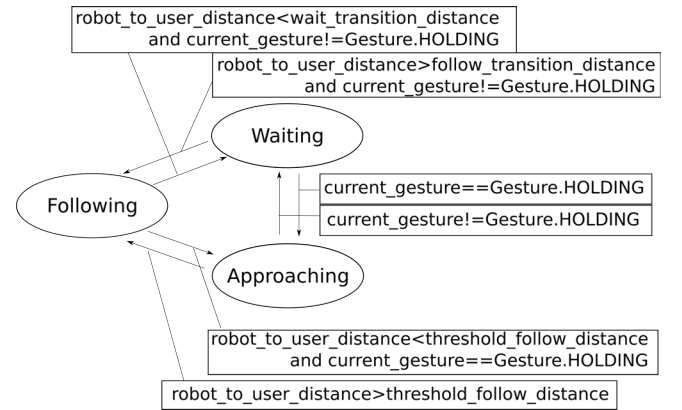


Fig. 2. The robot's operational mode is determined by a deterministic state machine with three states: *Following*, *Waiting*, and *Approaching*. Transition from one state to another causes the robot to re-evaluate its goal pose and provides intuitive human-robot interaction by switching between following behavior where the robot tries to keep a certain distance from the user and at-the-ready behavior, where the robot parks itself nearby until it needs to get closer to the user so they can drop off their items.

Pose suggestions $\mathbf{x}_i^{(s)}$ are provided by the user through pointing gestures, calculated by extracting human pose and

¹https://www.github.com/RutgersRAMLab/ramlab_mrta

gait information from 3d scans of the robot's environment as $\mathbf{h}_i(t)$, and projecting a ray from the user's extended arm pose $\eta_{h,i}(t)$ onto the environment as shown in Fig. 3 if the gait feature vector $\phi_{h,i}(t)$ matches a pointing gesture. If the user has an extended arm and an intersection point with a 3D scan of the environment is found, the positioning suggestion is then selected as the nearest feasible pose in \mathcal{X}_i , otherwise, the goal pose is selected as either the current robot position or the trailing pose described above, as set by the designer.

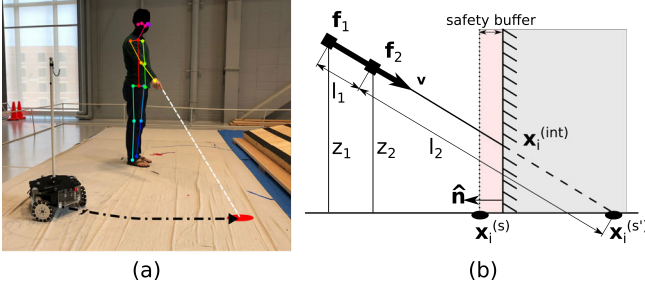


Fig. 3. (a) The robot uses point cloud data streamed from on-board RGB-D sensors to extract skeletons and estimate user pose and gait. (b) This pose is then used to determine robot pose suggestions by casting a ray from the extended arm onto the 3D point cloud map. Pose suggestions are then selected as the closest feasible pose in \mathcal{X}_i . Optimal trajectories to take the robot to a pose within a polytope \mathcal{G}_i around the pose suggestion are then calculated using (9), which optimizes coordinated paths for all robots.

The intersection point $\mathbf{x}_i^{(int)} = [x_i^{(int)}, y_i^{(int)}, z_i^{(int)}]^T$ is determined by a raycast that follows the segment between the base of the index finger $\mathbf{f}_1 = [x_1, y_1, z_1]^T$ and the tip of the index finger $\mathbf{f}_2 = [x_2, y_2, z_2]^T$, with direction vector $\mathbf{v} = \mathbf{f}_2 - \mathbf{f}_1$. The no-hit pose $\mathbf{x}_i^{(g')}$ is calculated as

$$\mathbf{x}_i^{(s')} = \mathbf{f}_2 + l_2 \mathbf{v}, \quad l_2 = \frac{z_2}{z_1 - z_2} \|\mathbf{f}_1 - \mathbf{f}_2\|, \quad (8)$$

where $\|\cdot\|$ is the Euclidean norm, and the goal pose is selected as $\mathbf{x}_i^{(s)} = \mathbf{x}_i^{(s')}$ if $\mathbf{x}_i^{(int)} = \mathbf{x}_i^{(s')}$, and $\mathbf{x}_i^{(s)} = [x_i^{(int)} + \delta_{buf} n_x, y_i^{(int)} + \delta_{buf} n_y, 0]^T$ otherwise, where $\hat{\mathbf{n}} = [n_x, n_y]^T$ is the map normal at $\mathbf{x}_i^{(int)}$ and δ_{buf} is the safety buffer.

The following pose $\mathbf{x}_i^{(f)}$ is determined as the first point along the user's trajectory that is at least δ_f away from the user. For tractability, user trajectories are stored as a queue so the last pose is at least δ_f along the trajectory, and the penultimate pose is less than δ_f along the trajectory. $\mathbf{x}_i^{(f)}$ is selected as the last pose in the user trajectory queue.

Conversely, the approach pose $\mathbf{x}_i^{(a)}$ is determined as the pose that is δ_a away from the user along a trajectory plan from the robot's current position to the user's current position. The robot maintains its pose once within a distance of δ_a , even if the user continues approaching it.

State transitions between *Following* and *Waiting* are dependent on the distance between the cart and the user, and are triggered at a distance of $R_{f \rightarrow w}$ for *Following* \rightarrow *Waiting*, and a distance of $R_{w \rightarrow f}$ for *Waiting* \rightarrow *Following*. Fig. 4 shows a schematic for this transition. Note that the actual trajectory changes once the *Following* \rightarrow *Waiting* threshold is crossed.

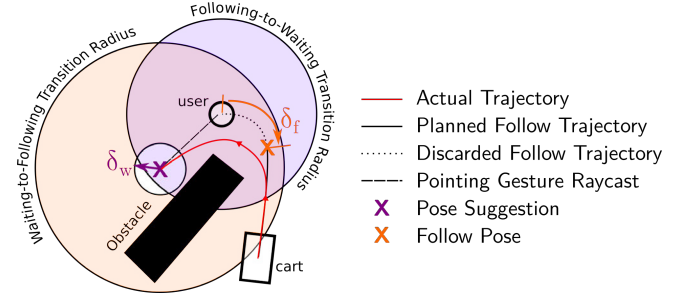


Fig. 4. The cart transitions from its *Following* state to its *Waiting* state when the cart reaches the following-to-waiting transition radius $R_{f \rightarrow w}$, and resumes following by transitioning from its *Waiting* state to its *Following* state when the user reaches the waiting-to-following transition radius $R_{w \rightarrow f}$. Switching to the cart's *Approaching* state pre-empts this transition, and instead triggers an approaching-to-following or approaching-to-waiting transition.

At any given time, the cart solves a sub-optimization to determine the best goal pose $\mathbf{x}_i^{(g)}$, which is selected based on a local optimization on a cost map determined by current and historical pose data. The robot targets a goal pose

$$\mathbf{x}_i^{(g)*} = \operatorname{argmin}_{\mathbf{x}_i^{(g)}} \sum_i \|e_s\| + \alpha_2 f_c(\mathbf{x}_i^{(g)}) + \alpha_3 f_b(\mathbf{x}_i^{(g)}), \quad (9)$$

where $e_s = \mathbf{x}_i^{(g)} - \mathbf{x}_i^{(d)}$ is the error between the goal pose $\mathbf{x}_i(t)$ and the desired pose $\mathbf{x}_i^{(d)}$, f_c is a pose cost that discourages waiting in poses that coincide with heavy foot traffic, and f_b is a blocking cost that discourages blocking the predicted trajectories of nearby humans. The relative strengths of these three cost components are determined by weighting variables α_2 and α_3 .

The pose cost f_c for a pose \mathbf{x}_i is determined by querying a heat map constructed from historical human trajectories using 2D kernel density estimation for query point $\mathbf{x}_i^{(g)}$ as

$$f_c(\mathbf{x}_i^{(g)}) = \frac{1}{|\mathcal{H}|} \sum_{\mathbf{h}(t) \in \mathcal{H}} \frac{1}{T_h} \int_{t_0}^{t_f} K_{\mathcal{H}}(\mathbf{x}_i^{(g)} - \mathbf{h}(t)) dt, \quad (10)$$

where $K_{\mathcal{H}}(\mathbf{x}) = \frac{1}{2\pi} |\mathbf{H}|^{-1/2} e^{-\frac{1}{2} \mathbf{x}^T \mathbf{H}^{-1} \mathbf{x}}$ is a user-defined bivariate anisotropic Gaussian kernel with covariance matrix \mathbf{H} , \mathcal{H} is the set of all past human trajectories, and T_h is the total trajectory time for $\mathbf{h}(t) \in \mathcal{H}$.

Since human access to shelves should be prioritized at all times, the robot makes an effort to move out of the way to avoid blocking any humans when it is in a *Waiting* state. This is done by adding a blocking potential f_b to (9) as

$$f_b(\mathbf{x}_i^{(g)}) = \sum_{\mathbf{h}_j \in \mathcal{N}(\mathbf{x}_i(t))} \phi_{\theta}(\mathbf{x}_i^{(g)}, \mathbf{h}_j) K_b \quad (11)$$

that uses a radial basis function (RBF) with user-defined parameter γ_1 , $K_b = e^{-\gamma_1 \|\mathbf{x}_i^{(g)}(t) - \mathbf{h}_j(t)\|^2}$, to determine the influence of nearby humans in the neighborhood $\mathcal{N}(\mathbf{x}_i(t))$ of \mathbf{r}_i , which is modified by a weighting $\phi_{\theta}(\mathbf{x}_i^{(g)}, \mathbf{h}_j) = e^{-\gamma_2 t^2}$ with user-defined temporal attenuation parameter γ_2 .

Once the goal pose is determined, the optimization problem in (1)–(6) can be solved in an iterative manner using a multi-robot receding-horizon planner described below.

Let $\mathbf{z}_i(t) = [\mathbf{x}_i^T(t), \dot{\mathbf{x}}_i^T(t)]^T$ be the state of the robot at time t , with dynamics $\dot{\mathbf{z}}_i(t) = \mathbf{f}(\mathbf{z}_i(t), \mathbf{u}_i(t))$, discretized around $\mathbf{z}_i(t_0)$ as $\mathbf{z}_i[k+1] = \mathbf{A}_d \mathbf{z}_i[k] + \mathbf{B}_d \mathbf{u}_i[k]$, with discretized matrices $\mathbf{A}_d = e^{\mathbf{A}T} \mathbf{A}$, $\mathbf{B}_d = (e^{\mathbf{A}T} - \mathbf{I}) \mathbf{B} \mathbf{A}^{-1}$, and $\mathbf{A} = \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{z}}|_{\mathbf{z}_i(t_0)}$, $\mathbf{B} = \frac{\partial \mathbf{f}(\cdot)}{\partial \mathbf{u}_i(t_0)}|_{\mathbf{z}_i(t_0)}$, such that the dynamics are written as $\Delta \mathbf{z}_i[k+1] = \mathbf{A}_d \Delta \mathbf{z}_i[k] + \mathbf{B}_d \Delta \mathbf{u}_i[k]$ given $\Delta \mathbf{z}_i[k] = \mathbf{z}_i[k] - \mathbf{z}_i(t_0)$ and $\Delta \mathbf{u}_i[k] = \mathbf{u}_i[k] - \mathbf{u}_i(t_0)$.

An A* search is performed on a coarsely-discretized map at each time step to determine desired trajectory $\mathbf{x}_i^{(G)}(t)$, which provides a global trajectory plan. The local trajectory plan $\mathbf{x}_i^{(l)}(t)$ for optimization window $t_0 < t < t+T$ is calculated by first placing a moving boundary \mathcal{W} with diagonal corners $[\mathbf{x}_i(t_0) - \mathbf{W}, \mathbf{x}_i(t_0) + \mathbf{W}]$, $\mathbf{W} = [w_x, w_y]^T$ around $\mathbf{x}_i(t_0)$, $i = 1, \dots, N$ such that $\|\mathbf{x}_i(t_0) - \mathbf{x}_i(t_0)\|_1 < \|\mathbf{W}\|_1$, $t_0 < t < t_0 + T$, and solving (1)–(6) for this window only, up to $\mathbf{x}_i^{(l)}(t_0 + T) = \mathbf{x}_i^{(d)}(t_0 + T)$ at $t = t_0 + T$, with tracking error $\mathbf{e}_i(t) = \mathbf{x}_i^{(l)}(t) - \mathbf{x}_i^{(G)}(t)$. We rewrite (1)–(6) as an approximately equivalent constrained optimization problem

$$\mathbf{x}_i^{(l)*} = \underset{\substack{i=1, \dots, N \\ \mathbf{x}_i(t) \\ t_0 < t < t_0 + T}}{\operatorname{argmin}} \sum_{i=1}^N p_i \sum_{t=t_0}^{t_0+T} \|\mathbf{e}_i(t)\|_{\mathbf{Q}} \quad (12)$$

$$\text{s.t. } \dot{\mathbf{z}}_i(t) = \mathbf{f}(\mathbf{z}_i(t), \mathbf{u}_i(t), \quad (13)$$

$$\mathbf{z}_i^{(l)}(t_0) = \mathbf{z}_i(t_0), \quad (14)$$

$$\mathbf{x}_i^{(l)}(t) \in \mathcal{W}, t_0 < t < t_0 + T, \quad (15)$$

$$\mathbf{x}_i^{(l)}(t) \in \mathcal{X}, t_0 < t < t_0 + T, \quad (16)$$

$$\|\mathbf{x}_i^{(l)}(t) - \mathbf{x}_j^{(l)}(t)\| > R_r, j \neq i, \quad (17)$$

$$\|\mathbf{x}_i^{(l)}(t) - \mathbf{h}_j^{(l)}(t)\| > R_h, \quad (18)$$

$$\mathbf{u}_i(t) \in \mathcal{U}_i, \quad (19)$$

where $\|\cdot\|_{\mathbf{Q}}$ is the matrix norm weighted by positive-definite matrix \mathbf{Q} , and (17)–(18) are time-varying collision constraints for robots and humans with distances R_r and R_h , respectively.

(12)–(19) are recast as a constrained quadratic program (QP) and solved with an iterative linear time-varying model predictive control solver [26], with $\Delta \tilde{\mathbf{e}} = [\Delta \tilde{\mathbf{e}}_1^T, \dots, \Delta \tilde{\mathbf{e}}_N^T]^T$, $\Delta \tilde{\mathbf{e}}_i = [\Delta \mathbf{e}_i^T[0], \dots, \Delta \mathbf{e}_i^T[k]]^T$, $\Delta \mathbf{e}_i[k] = \mathbf{e}_i[k] - \mathbf{e}_i[0]$ and $\mathbf{e}_i[k] = \mathbf{e}_i(t_0 + k\Delta T)$, $\Delta T = T/N_{\text{steps}}$, where N_{steps} is the prediction horizon. (12)–(19) are then equivalent to

$$\min_{\Delta \tilde{\mathbf{e}}} \Delta \tilde{\mathbf{e}}^T \tilde{\mathbf{Q}} \Delta \tilde{\mathbf{e}} \quad (20)$$

$$\text{s.t. } \Delta \tilde{\mathbf{e}} = \tilde{\mathbf{A}}_d \Delta \tilde{\mathbf{e}} + \tilde{\mathbf{B}}_d \tilde{\mathbf{u}}, \quad (21)$$

$$\underline{\mathbf{Z}} \leq \tilde{\mathbf{Z}} \tilde{\mathbf{e}} \leq \overline{\mathbf{Z}}, \quad (22)$$

$$\underline{\mathbf{R}}_r \leq \tilde{\mathbf{R}}_r \tilde{\mathbf{e}}_r \leq \overline{\mathbf{R}}_r, \quad (23)$$

$$\underline{\mathbf{R}}_h \leq \tilde{\mathbf{R}}_h \tilde{\mathbf{e}}_h \leq \overline{\mathbf{R}}_h, \quad (24)$$

$$\underline{\mathbf{U}} \leq \tilde{\mathbf{U}} \tilde{\mathbf{u}} \leq \overline{\mathbf{U}}, \quad (25)$$

such that $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{h}}$ are constructed similarly to $\tilde{\mathbf{e}}$ from $\mathbf{u}_i[k]$ and $\mathbf{h}_j[k]$, respectively, $\tilde{\mathbf{A}}_d$ and $\tilde{\mathbf{B}}_d$ in (21) are selected to satisfy (13), $\underline{\mathbf{Z}}$, $\tilde{\mathbf{Z}}$ and $\overline{\mathbf{Z}}$ satisfy (14)–(16), $\tilde{\mathbf{e}}_r$ and $\tilde{\mathbf{e}}_h$ are constructed from all combinations of $\mathbf{x}_i - \mathbf{x}_j$ and $\mathbf{x}_i - \mathbf{h}_j$, respectively, with $\underline{\mathbf{R}}_r$, $\tilde{\mathbf{R}}_r$ and $\overline{\mathbf{R}}_r$ chosen to satisfy (17) and

$\underline{\mathbf{R}}_h$, $\tilde{\mathbf{R}}_h$ and $\overline{\mathbf{R}}_h$ chosen to satisfy (18), and $\underline{\mathbf{U}}$, $\tilde{\mathbf{U}}$ and $\overline{\mathbf{U}}$ satisfy (19) and weighting matrix $\tilde{\mathbf{Q}}$ chosen as

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} \mathbf{P}_1 & & \\ & \ddots & \\ & & \mathbf{Q} \mathbf{P}_N \end{bmatrix}, \quad \mathbf{P}_i = p_i \begin{bmatrix} 1 & & \\ & 1 & \\ & & k_1 \end{bmatrix}. \quad (26)$$

Where applicable, constraints are relaxed or tightened to make designing the QP constraints feasible or speed up solutions. The overall data flow for both systems is shown in Fig. 5. An algorithmic description of the trajectory optimization method is given in Algorithm 1.

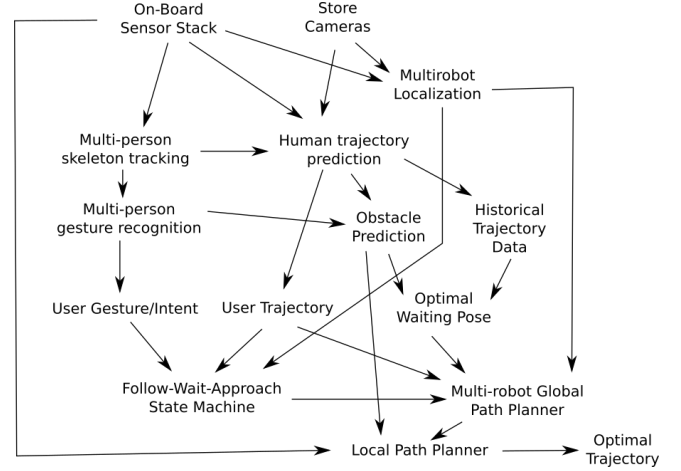


Fig. 5. Data flows in a similar manner for both the simulation and the in-lab experimental platforms. Sensor data is fed into the on-board computer to perform skeleton tracking, trajectory prediction, localization and local obstacle mapping. Skeleton data is used to determine user gestures, which are then used to determine contactless positioning suggestions. Gesture, trajectory and localization data is then used to determine each robot's state as shown in Fig. 2. The goal pose is then used to plan a global and local optimal trajectory.

Algorithm 1 Multirobot Receding-Horizon Planner

```

1: procedure GENERATE OPTIMAL TRAJECTORY
2:    $\mathbf{x}_i^{(d)} \leftarrow \text{DESIREDPOSE}(\text{robot.State})$   $\triangleright$  from (7)
3:    $\mathbf{x}_i^{(g)} \leftarrow \text{OPTIMIZEPOSE}(\mathbf{x}_i^{(d)})$   $\triangleright$  from (9)
4:   while True do
5:      $\mathbf{x}_i^{(l)}(t) \leftarrow \text{LOCALWINDOW}(\mathbf{x}_i^{(G)}(t) = A^*(\mathbf{x}_i^{(g)}))$ 
6:      $\mathbf{x}_i^{(l)*}(t) \leftarrow \text{QP}(\mathbf{e}_i(t) = \mathbf{x}_i^{(l)} - \mathbf{x}_i^{(G)})$   $\triangleright$  from (20)
7:      $\mathbf{u}_i(t) \leftarrow \text{LOWLEVELCTRL}(\mathbf{x}_i^{(l)*}(t), \mathbf{u}_i^*(t))$ 

```

III. EXPERIMENTAL SETUP

The details of both the simulation studies and the in-lab experiments are provided below. The carts in both experiments follow a common configuration with regards to their on-board sensors and localization methods, but they differ in terms of low-level control due to the differences in their dynamics equations. The simulation studies are performed using a differential drive Turtlebot3 platform, whereas the in-lab experiments use a three-wheeled omni-directional robot.

A. Multi-Cart Simulations

A multi-cart supermarket environment is simulated in ROS/Gazebo using modified Turtlebot3 platforms. Fig. 6 shows one of the supermarket layouts used to perform simulation experiments. All robots are equipped with laser scanners, RGB cameras and Kinect-like depth sensors, and communicate over ROS. The simulated robots share floor plans and environmental scans as 2D maps, which simulate cooperative map sharing over a wifi network installed in a real supermarket.

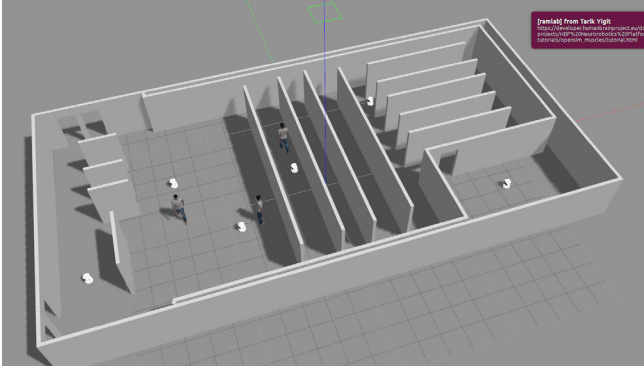


Fig. 6. A supermarket is simulated in Gazebo, and includes up to 40 autonomous shopping carts and simulated humans. Several humans can be actively controlled by the user to test pointing and walking behavior, and the rest are simulated using crowd dynamics.

Motion planning is done according to (9) using each robot's local pose estimation as determined by adaptive Monte Carlo localization (AMCL). Obstacle avoidance and goal selection is done locally (i.e. separately) for each robot with only on-board sensors and shared map and location data. Obstacle avoidance for the simulated robot is mainly handled by a dynamic window planner that only considers the world around the robot within a moving window centered around the robot. We implement (11) as a costmap layer for a node that takes desired poses and outputs optimal final poses according to (9). These layers are updated with incoming sensor data, forcing recalculations as necessary. Optimal poses are then fed into the motion planner nodes to generate coordinated obstacle-free trajectories. Pose estimates and internal robot states are automatically collected in a bag file for later analysis. The simulations tests the following scenarios:

- 1) State and goal switching for a user-cart setup
- 2) Obstacle avoidance and shelf blocking
- 3) Multi-robot planning

All localization and mapping is done using laser scanner, depth camera and RGB camera plugins attached to the robot frame (on-board) through Gazebo. Obstacle avoidance, human tracking, skeletal pose estimation and trajectory planning is therefore performed with on-board knowledge only, but each robot is provided with a fully-explored global static map during initialization. This is meant to mimic actual store conditions, as floorplans are often readily available, but obstacles such as shoppers, carts or other temporary roadblocks are

unavailable *a priori* and therefore must be determined using on-board sensors.

B. Human-Following Experiments

We perform experiments with the three-wheel omnidirectional robot shown in Fig. 7. Robot models and low-level controllers are adapted from [24] with kinematic model

$$\dot{\mathbf{q}} = \bar{B}(\theta)\mathbf{u}, \quad \bar{B}(\theta) = \begin{bmatrix} \mathbf{R}_\theta & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{R}_\theta = \begin{bmatrix} c\theta & -s\theta \\ s\theta & c\theta \end{bmatrix}, \quad (27)$$

where $\mathbf{q} = [x, y, \theta]$ are the generalized robot coordinates, θ is the angle between the world-frame x -axis and the robot's leading wheel, $s\theta = \sin \theta$, $c\theta = \cos \theta$, and inputs $\mathbf{u} = [u_1, u_2, u_3]^T$ depend on wheel angular velocities ω_1 , ω_2 and ω_3 as $u_1 = \frac{r\sqrt{3}}{3}(\omega_3 - \omega_2)$, $u_2 = -\frac{2r}{3}(\omega_1 - \omega_2 - \omega_3)$, $u_3 = \frac{r}{6r}(\omega_1 + \omega_2 + \omega_3)$. Additionally, $\bar{\mathbf{B}}^T(\theta) = \bar{\mathbf{B}}^{-1}(\theta)$.

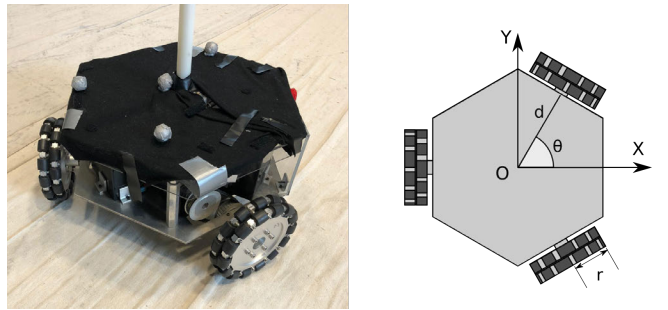


Fig. 7. A three-wheel omni-directional robot is instructed to follow a human target in a lab setting. The robot is fitted with color stereo and RGB-D cameras. Vicon motion tracking cameras are used to validate the experimental result.

The robot is fitted with forward-facing color and depth cameras for navigation and human pose and gesture recognition. Retro-reflective markers are adhered to the top surface of the robot and a set of Vicon Bonita cameras are used to collect ground truth robot odometry and human pose information.

IV. RESULTS AND DISCUSSION

We report the results for the simulation studies and the in-lab experiments together under four main categories: State and goal switching, non-blocking pose selection, multi-robot planning, and gesture-based suggestion tracking. We look at each category in isolation to test edge cases as well as general robustness, and to provide more in-depth analysis.

A. State and Goal Switching

Fig. 9 shows the simulation trajectories for a cart following a user through parallel aisles. The user is asked to remain stationary while the cart navigates towards them. The user moves to the next aisle once the robot has reached a waiting position. The robot demonstrates state switching behavior by first following the user, then waiting at a position the user points to, then following them again once they move further than $R_{w \rightarrow f}$, and finally waiting at a second position determined by the user. Related metrics are shown in Fig. 8.

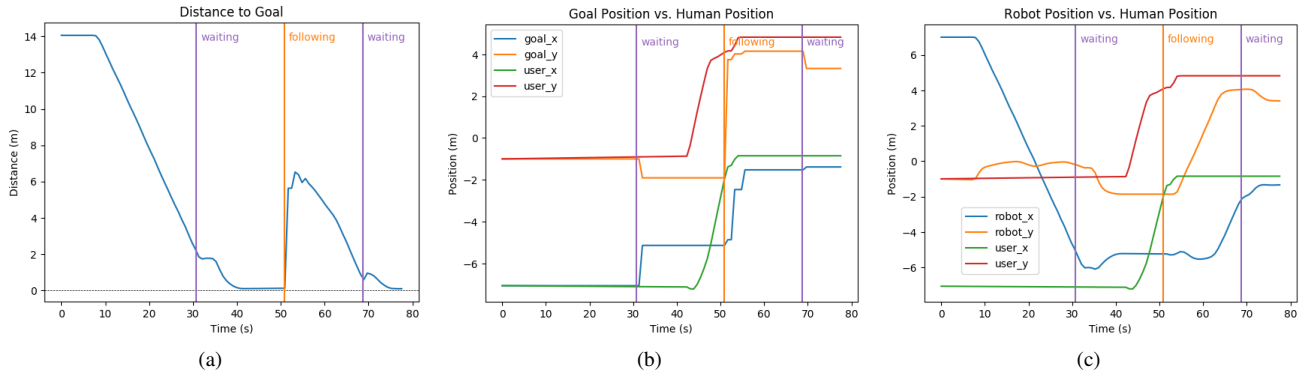


Fig. 8. (a) Plot of the distance between the robot and its selected goal. (b) Comparison of the goal position and the human position. The goal pose initially aligns with the human’s pose due to the absence of a trailing path, and is subsequently switched to an optimal waiting pose as the robot crosses $R_{f \rightarrow w}$. Following behavior resumes when the human starts moving away, with a goal pose that trails the human’s trajectory by δ_f . The robot selects a waiting pose based on user suggestions once it reaches $R_{f \rightarrow w}$ again. (c) Comparison of robot and human positions as influenced by distance-dependent state switching.



Fig. 9. Results from a user following test, where the cart was asked to follow the user at a distance of $R_{f \rightarrow w} = 1$ m. The cart initially follows the user until it reaches $R_{f \rightarrow w}$, and then moves to the suggested waiting pose (purple). The robot resumes following once the user moves further than $R_{w \rightarrow f}$.

B. Non-Blocking Pose Selection

Non-blocking pose selection allows the robot to make way for other shoppers while still waiting close to its user. Non-blocking behavior relies on three costmap layers, shown in Fig. 10, representing the distance cost, which increases with L2 distance from the pose suggestion, the pose cost, which mirrors the historical shopper heatmap, and the blocking cost, which increases towards the predicted human path. The minimal cost across the combined costmap provides the optimal waiting position for the robot. The blocking cost is the most critical layer for non-blocking pose selection. The layer ingests time-stamped pose projections based on the motion of nearby humans, and converts them into a distance-based penalty that increases the cost of any waiting pose that blocks the path of nearby shoppers. The resultant cost is discounted with increasing travel time from the shopper’s current position.

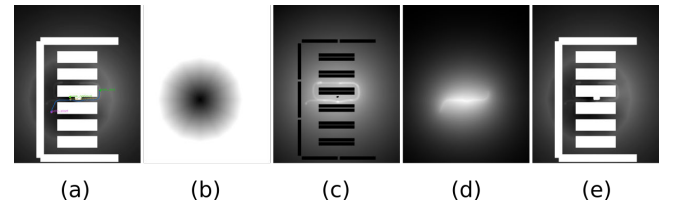


Fig. 10. Breakdown of costs for the non-blocking pose selection process. (a) Costmap with poses and paths annotated. (b) Distance cost. (c) Pose cost. (d) Blocking cost. (e) Combined cost.

While the pose cost is constant since the shopper heatmap is roughly constant over long-term observations, the distance cost and the blocking cost are highly dependent on the suggested waiting position and the motions of nearby humans. We demonstrate this variability in Fig. 11 by varying the suggested waiting position and the human trajectory to match the different stages (and immediacy) of motion.

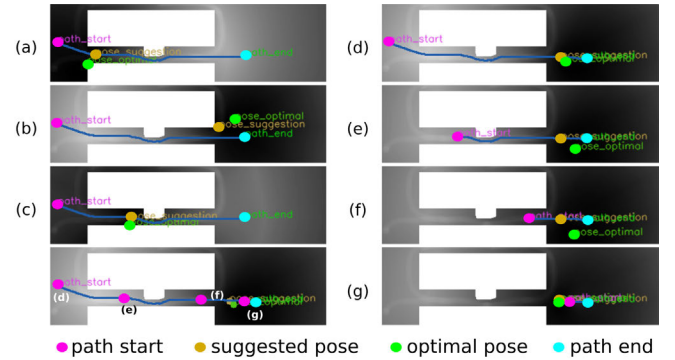


Fig. 11. Optimal non-blocking poses selected for various scenarios. (a-c) User suggestion is varied along the projected path. (d-g) User approaches and passes robot.

The behaviors described in this section are tunable by the designer using α_2 and α_3 in (9). All figures shown here are calculated using $\alpha_2 = 0.7$ and $\alpha_3 = 0.4$ with a map resolution of 5 cm/pixel, but we note that different robot configurations or supermarket layouts might require different weightings.

C. Multi-Robot Planning

In order to validate our planner approach, we ran simulation studies with 50 robots and 50 human subjects. Fig. 12 shows trajectories from 1000 trials with 50 human subjects run with and without our multirobot planner. Human start and goal positions are randomized.

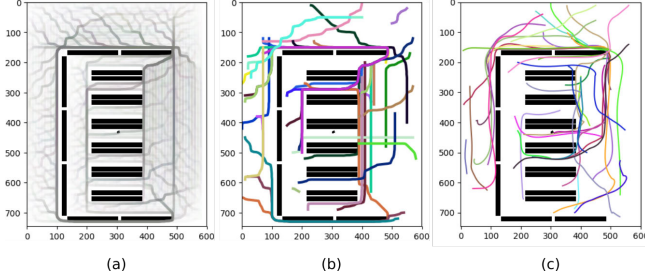


Fig. 12. (a) Combined global trajectories for 50 people, using data from all trials. (b) Trajectories from a single trial (no robots) colored by person, showing high-activity areas and bottlenecks. (c) Trajectory data from a trial with robots using the multirobot planner. Map resolution is 5cm/px.

We compared the performance of our planner against three other scenarios: A scenario where no robots are present and humans follow their global trajectories, a scenario where robots are present but exhibit pure pursuit behavior, and a scenario where robots plan individual trajectories. We compare the four scenarios in Table I according to three metrics: mean separation, which keeps track of robot-user distance; wait time, which keeps track of how long the human subjects had to wait for a path to clear; and time to goal, which measures the total trajectory time. The multirobot planner outperforms other methods when looking at wait time and time to goal metrics, whereas the individual planner performs well in the mean separation category, as it does not care if the human it is following gets stuck.

TABLE I
PLANNER COMPARISON FOR LARGE-SCALE MULTIROBOT SIMULATIONS

Metric	Mean Sep. Dist.	Wait Time	Time To Goal
No Robots (Base)	N/A	39.5 s	210.9 s
No Planner	2.41 m	117.3 s	235.0 s
Individual Planner	2.30 m	103.8 s	227.5 s
Multirobot Planner	2.34 m	100.2 s	225.9 s

D. Gesture-Based Suggestion Tracking

Pose suggestions were extracted from human finger estimates, transformed into the world frame and sent to the cone-bot at a rate of 20 Hz via wifi for online tracking. Pose suggestions were filtered with a Savitzky-Golay filter (1.5 s window, 3rd order) to reduce the effect of occasional jumps in finger tracking. Tracking errors are shown in Fig. 13 and were on average around 0.7 m across all trials and were calculated using the instantaneous pose suggestions.

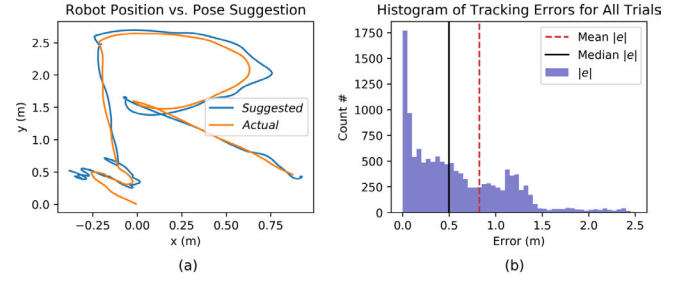


Fig. 13. (a) Experimental gesture-based suggestion tracking results for a single trial. (b) Histogram of normed tracking errors across all trials, with mean error (dashed red) median error (solid black).

Errors are larger than previous tracking results with the same robot, correlating to users tendency to lead the robot with their pose suggestions. Additionally, the jerky movement of the finger in some trials and the delay in pose estimation and transmission likely moved the filtered pose suggestion away from the user-intended trajectory. Practitioners who do not require real-time tracking are advised to collect finger estimates over a couple of seconds and transmit the averaged pose suggestion. Edge cases with slow and fast changes in the suggested pose are shown in Fig. 14.

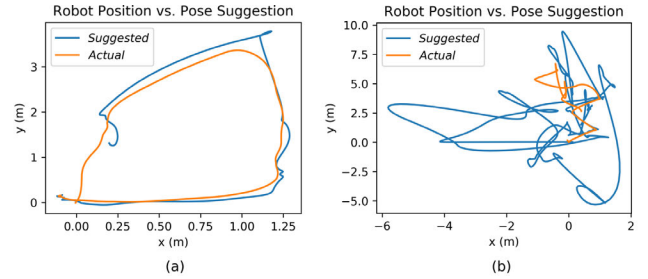


Fig. 14. Edge case where the user provides pose suggestions that (a) change relatively slowly, and (b) change very rapidly.

V. CONCLUSION

In this paper, we presented a framework for autonomous human-following shopping carts in crowded environments with multiple users and robots. Our framework switched between following the user, waiting while they browsed the shelves, and approaching them so they could place their items in the basket. We considered historical shopper trajectories to determine optimal waiting positions for the cart while the user was gathering items. User actions and gestures were determined by a skeletal estimate of the user's pose. User access to shelves was prioritized by updating the robot pose cost by a shelf-blocking term. We presented a method for users to suggest waiting poses using hand and arm gestures, which we extracted from the skeletal estimates. We demonstrated our framework on both a large-scale simulation in a supermarket setting, and in-lab experiments with a three-wheel omni-directional robot. Future work includes the fabrication of a prototype multi-robot shopping cart team and on-site deployment.

VI. ACKNOWLEDGEMENTS

Victoria Hong and Faiza Sikandar would like to thank the Rutgers University Project SUPER STEM Summer Stipend Program for their financial and academic support.

REFERENCES

- [1] W. He, Z. Li, and C. L. P. Chen, "A survey of human-centered intelligent robots: issues and challenges," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 4, pp. 602–609, 2017.
- [2] V. Kulyukin, C. Gharpure, and J. Nicholson, "RoboCart: toward robot-assisted navigation of grocery stores by the visually impaired," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug. 2005, pp. 2845–2850.
- [3] M. J. Islam, J. Hong, and J. Sattar, "Person-following by autonomous robots: A categorical overview," *The International Journal of Robotics Research*, vol. 38, no. 14, pp. 1581–1618, Dec. 2019.
- [4] P. Wang, W. Li, P. Ogunbona, J. Wan, and S. Escalera, "RGB-D-based human motion recognition with deep learning: A survey," *Computer Vision and Image Understanding*, vol. 171, pp. 118–139, Jun. 2018.
- [5] H.-B. Zhang, Y.-X. Zhang, B. Zhong, Q. Lei, L. Yang, J.-X. Du, and D.-S. Chen, "A Comprehensive Survey of Vision-Based Human Action Recognition Methods," *Sensors*, vol. 19, no. 5, p. 1005, Jan. 2019.
- [6] D. Herrera, M. Monllor, D. Santiago, F. Roberti, and R. Carelli, "Null-space based control for human following and social field avoidance," in *2017 XVII Workshop on Information Processing and Control (RPIC)*, Sep. 2017, pp. 1–6.
- [7] Y. Sun, L. Sun, and J. Liu, "Human comfort following behavior for service robots," in *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2016, pp. 649–654.
- [8] W. Chi, J. Wang, and M. Q.-H. Meng, "A Gait Recognition Method for Human Following in Service Robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 9, pp. 1429–1440, Sep. 2018.
- [9] H.-M. Gross, A. Scheidig, S. Miller, B. Schtz, C. Fricke, and S. Meyer, "Living with a Mobile Companion Robot in your Own Apartment - Final Implementation and Results of a 20-Weeks Field Study with 20 Seniors*," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 2253–2259.
- [10] R. Algabri and M.-T. Choi, "Deep-Learning-Based Indoor Human Following of Mobile Robot Using Color Feature," *Sensors*, vol. 20, no. 9, p. 2699, Jan. 2020.
- [11] P. Nikdel, R. Shrestha, and R. Vaughan, "The Hands-Free Push-Cart: Autonomous Following in Front by Predicting User Trajectory Around Obstacles," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 4548–4554.
- [12] L. Pang, L. Zhang, Y. Yu, J. Yu, Z. Cao, and C. Zhou, "A human-following approach using binocular camera," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2017, pp. 1487–1492.
- [13] K. Li, Y. Xu, J. Wang, and M. Q.-H. Meng, "SARL: Deep Reinforcement Learning based Human-Aware Navigation for Mobile Robot in Indoor Environments," in *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Dec. 2019, pp. 688–694.
- [14] C.-A. Yang and K.-T. Song, "Control Design for Robotic Human-Following and Obstacle Avoidance Using an RGB-D Camera," in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, Oct. 2019, pp. 934–939.
- [15] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-Robot Interaction: Crowd-Aware Robot Navigation With Attention-Based Deep Reinforcement Learning," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 6015–6022.
- [16] B.-J. Lee, J. Choi, C. Baek, and B.-T. Zhang, "Robust Human Following by Deep Bayesian Trajectory Prediction for Home Service Robots," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7189–7195.
- [17] S. Yan, J. Tao, J. Huang, and A. Xue, "Model Predictive Control for Human Following Rehabilitation Robot," in *2019 IEEE International Conference on Advanced Robotics and its Social Impacts (ARSO)*, Oct. 2019, pp. 369–374.
- [18] Z. Talebpour and A. Martinoli, "Risk-Based Human-Aware Multi-Robot Coordination in Dynamic Environments Shared with Humans," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 3365–3372.
- [19] —, "Multi-Robot Coordination in Dynamic Environments Shared with Humans," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 4593–4600.
- [20] Z. Talebpour, D. Viswanathan, R. Ventura, G. Englebienne, and A. Martinoli, "Incorporating perception uncertainty in human-aware navigation: A comparative study," in *2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2016, pp. 570–577.
- [21] N. Prez-Higueras, F. Caballero, and L. Merino, "Learning Human-Aware Path Planning with Fully Convolutional Networks," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 5897–5902.
- [22] A. Dutta, A. Ghosh, and O. P. Kreidl, "Multi-robot Informative Path Planning with Continuous Connectivity Constraints," *2019 International Conference on Robotics and Automation (ICRA)*, pp. 3245–3251, May 2019.
- [23] J. Chen and W.-j. Kim, "A Human-Following Mobile Robot Providing Natural and Universal Interfaces for Control With Wireless Electronic Devices," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 5, pp. 2377–2385, Oct. 2019.
- [24] K. Hunte and J. Yi, "Collaborative Object Manipulation Through Indirect Control of a Deformable Sheet by a Mobile Robotic Team," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, Aug. 2019, pp. 1463–1468.
- [25] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 3, 2009.
- [26] M. Edmonds and J. Yi, "A Model Predictive Control Based Iterative Trajectory Optimization Method for Systems with State-Like Disturbances," in *2019 American Control Conference (ACC)*, Jul. 2019, pp. 1635–1640.