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. The 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 used as positioning suggestions. 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 a model predictive control (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. We further demonstrate the human-following block with in-lab tests using a group of omni-directional robots.**

I. Introduction

Human-following autonomous robots increase human productivity and lower physical barriers for many human-centric tasks. Unlike industrial robots, human-following robots are designed to work directly with humans and consider additional human-centric performance metrics such as comfort, adaptability and safety [1], which enrich the human experience in many domains from manufacturing to personal in-home robotics. The recent global pandemic has additionally underlined the need for 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], [3] and some attempts to commercialize smart carts, the majority of relevant research focuses on other domains where human-following robots are used [4]. These works commonly focus on computer vision and artificial intelligence methods to track humans and then combine the tracking results with traditional planning and control methods to provide human-following functionality.

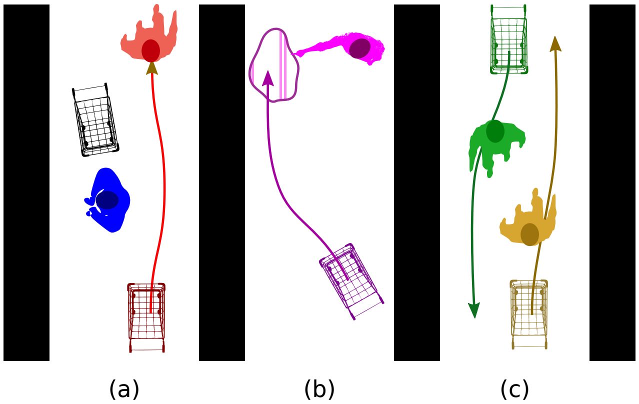


Fig. 1. (a) The human-following shopping cart navigates the aisles using on-board cameras and RGB-D sensors, which provide mapping information for collision avoidance. (b) The user can use gesture-based suggestions, e.g. pointing, to assign target positions for the shopping cart. (c) Shopping carts communicate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

Approaches to determining human-following robot poses and trajectories are varied. In [5], robots follow humans but refrain from entering their personal spaces via social field avoidance, and similar concepts are explored in [6]. In [7], [8], planned trajectories for human-following robots utilize predictions of human trajectories. The model predictive controller proposed in [9] also uses predictions of human trajectories, and a formation-based tracking method is presented in [10] for multiple robots. A recent review of human-aware navigation is provided in [11]. Visual servoing approaches include deep reinforcement learning [12], marker recognition [13], and wearable devices [14].

Human-robot interactions for human-following robots are commonly based on user gestures determined using cameras or IMUs. Human skeleton and pose data are used to determine pointing gestures [15] or actions/gaits [16], [17]. Skeletal estimates are similarly fused with face recognition for human-following robots in [18]. Gesture-based commands are used to determine tracking behavior in [19].

In this paper, we propose a human-following algorithm that is supplemented by gesture-based contactless positioning suggestions. We also outline a novel approach to determine human-aware waiting locations for autonomous shopping carts both with and without gesture-based suggestions. Our method is tailored for robot-assisted shopping, and considers potential shopper and robot trajectories when selecting comfortable waiting poses and paths. We validate the cooperative multi-cart planner by simulating a supermarket setting, and demonstrate the gesture-based suggestion method with single

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)

robot experiments. The main contributions are therefore a framework and related algorithms for environment-aware multi-cart human following, as well as our code released on GitHub[1].

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

II. Problem Statement

Let $\mathbf{R}$ be the set of N cooperative human-following carts. The pose of robot $r_i \in \mathbf{R}$, $i = 1, ..., N$ at time t is $\boldsymbol{x}_i(t) = [x_i(t), y_i(t), \theta_i(t)]^T$, $\boldsymbol{x}_i(t) \in \mathcal{X}_i$, where $\mathcal{X}_i$ is the feasible pose set for r_i, with dynamics given as $[\dot{\boldsymbol{x}}_i^T(t), \ddot{\boldsymbol{x}}_i^T(t)]^T = \boldsymbol{f}(\boldsymbol{x}_i(t), \dot{\boldsymbol{x}}_i^T(t), \boldsymbol{u}_i(t))$, where $\boldsymbol{u}_i(t)$ are the control inputs for r_i at time t, and the pose goal for robot r_i is $\boldsymbol{x}_i^{(g)} = [x_i^{(g)}, y_i^{(g)}, \theta_i^{(g)}]^T$, $\boldsymbol{x}^{(g)} \in \mathcal{G}_i$, where $\mathcal{G}_i \subseteq \mathcal{X}_i$ is a polytope of goal states. We define $\boldsymbol{h}_i(t) = [\boldsymbol{\eta}_i(t)^T, \boldsymbol{\phi}_i(t)^T]^T$ as the concatenation of the pose vector $\boldsymbol{\eta}_i(t)$ [20] and the gait feature vector $\boldsymbol{\phi}_i(t)$ [21] for the human that r_i is targeting.

The optimal cart trajectories $\boldsymbol{x}_i(t)^*$ are solutions to a constrained multirobot trajectory optimization problem,

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

$$\text{s.t.}\ \psi(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t)) = p_i \|\boldsymbol{x}_i(t) - \boldsymbol{x}_i^{(g)}\|, \quad \text{(1b)}$$

$$\dot{\boldsymbol{x}}_i(t) = \boldsymbol{f}(\boldsymbol{x}_i(t), \boldsymbol{u}_i(t)), \quad \text{(1c)}$$

$$\boldsymbol{x}_i(t) \in \mathcal{X}_i,\ \boldsymbol{x}_i(t+T) \in \mathcal{G}_i,\ \boldsymbol{u}_i(t) \in \mathcal{U}_i, \quad \text{(1d)}$$

where T is free and p_i is the priority of the ith robot. The human-following shopping cart problem with positioning suggestions is then defined using the optimization problem as finding an optimal $\boldsymbol{x}_i(t)$ for each r_i to minimize (1a) for the entire robot group $\mathbf{R}$, given predictions for $\boldsymbol{h}_i(t)$. In particular, we wish to find optimal pose goals $\boldsymbol{x}_i^{(g)}$ to calculate $\boldsymbol{x}_i^*(t)$.

III. Solution Method

The combined planning and control problem in Section II is solved in two steps by first planning high-level optimal target positions and trajectories, and then tracking these trajectories using model predictive control (MPC). The target positions $\boldsymbol{x}_i^{(g)}$ are selected by optimizing desired poses $\boldsymbol{x}_i^{(d)}$ described in (2) using the optimization in (4), so that each robot either trails $\boldsymbol{h}_i(t)$ by a distance of δ_f, or moves to a waiting position that is within a distance of δ_w from a human-suggested pose, or approaches the human at a distance of δ_a so they can drop off any items.

The desired pose $\boldsymbol{x}_i^{(d)}$ is determined at each time step by a finite state machine with *Following*, *Waiting* and *Approaching* states, which selects the corresponding robot pose as

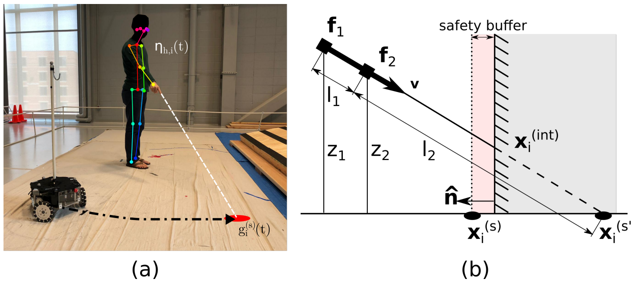
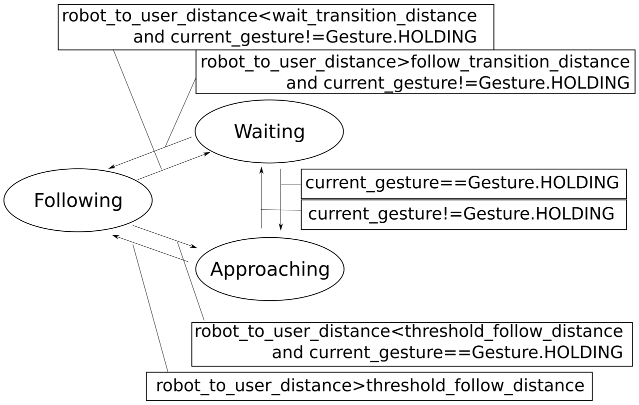


Fig. 2. (a) Point cloud data is used to estimate user pose and gait. (b) Human pose estimates determine robot pose suggestions by casting a ray from the extended arm onto the environment and selecting the closest feasible pose in $\mathcal{X}_i$. Optimal trajectories are calculated via (4) to a pose within a polytope $\mathcal{G}_i$ around the pose suggestion.



robot_to_user_distance<wait_transition_distance and current_gesture!=Gesture.HOLDING
robot_to_user_distance>follow_transition_distance and current_gesture!=Gesture.HOLDING
Waiting
Following
current_gesture==Gesture.HOLDING
current_gesture!=Gesture.HOLDING
Approaching
robot_to_user_distance<threshold_follow_distance and current_gesture==Gesture.HOLDING
robot_to_user_distance>threshold_follow_distance

Fig. 3. The robot's operational mode and its goal pose are determined by a state machine with three states: *Following (F)*, where the robot keeps within δ_f of the human, *Waiting (W)*, where the robot parks itself nearby, and *Approaching (A)*, where the robot approaches to within δ_a.

$$\boldsymbol{x}_i^{(d)} = \begin{cases} \boldsymbol{x}_i^{(w)}, & \text{robot\_state=}Waiting \\ \boldsymbol{x}_i^{(f)}, & \text{robot\_state=}Following \\ \boldsymbol{x}_i^{(a)}, & \text{robot\_state=}Approaching \end{cases} . \quad (2)$$

State transitions between *Following (F)* and *Waiting (W)* are triggered at a robot-human distance of $R_{f\to w}$ for *F*→*W*, and a distance of $R_{w\to f}$ for *W*→*F*. Fig. 3 shows the transition conditions for each state, and Fig. 4 shows a schematic for the *F*→*W* transition. Note that the actual trajectory changes once the *F*→*W* threshold is crossed.

The user provides pose suggestions $\boldsymbol{x}_i^{(s)}$ through pointing gestures calculated form $\boldsymbol{h}_i(t)$. Rays are projected from the user's extended arm pose $\boldsymbol{\eta}_i(t)$ onto the environment as shown in Fig. 2 if the gait feature vector $\phi_i(t)$ matches a pointing gesture. The desired pose is selected as a feasible pose in $\mathcal{X}_i$ closest to the suggested pose, otherwise, the desired pose is selected as $\boldsymbol{x}_i(t)$ or the trailing pose described above, as set by the designer.

The intersection point $\boldsymbol{x}_i^{(int)}$ is follows the segment between the base $\boldsymbol{f}_1$ and tip $\boldsymbol{f}_2$ of the index finger, with direction vector $\boldsymbol{v} = \boldsymbol{f}_2 - \boldsymbol{f}_1$. The no-hit pose $\boldsymbol{x}_i^{(g')}$ is

$$\boldsymbol{x}_i^{(s')} = \boldsymbol{f}_2 + l_2\boldsymbol{v},\ l_2 = \frac{z_2}{z_1 - z_2}\|\boldsymbol{f}_1 - \boldsymbol{f}_2\|, \quad (3)$$

where $\|\cdot\|$ is the Euclidean norm, and the goal pose is selected with safety buffer δ_b as $\boldsymbol{x}_i^{(s)} = \boldsymbol{x}_i^{(s')}$ if $\boldsymbol{x}_i^{(int)} =$

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

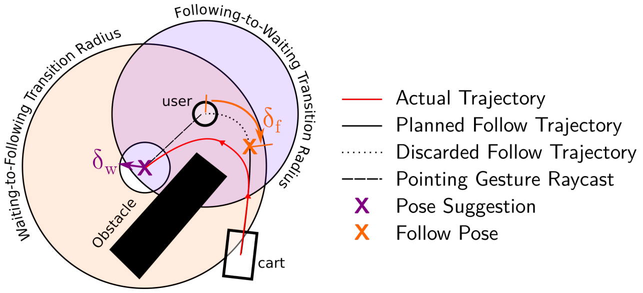

Following-to-Waiting Transition Radius
Waiting-to-Following Transition Radius
user
δf
δw
Obstacle
cart
Actual Trajectory
Planned Follow Trajectory
Discarded Follow Trajectory
Pointing Gesture Raycast
X Pose Suggestion
X Follow Pose

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.

$\boldsymbol{x}_i^{(s')}$, and as $\boldsymbol{x}_i^{(s)} = [x_i^{(int)} + \delta_b n_x, y_i^{(int)} + \delta_b n_y, 0]^T$ otherwise, where $\hat{\boldsymbol{n}} = [n_x, n_y]^T$ is the map normal at $\boldsymbol{x}_i^{(int)}$.

The following pose $\boldsymbol{x}_i^{(f)}$ is the pose trailing the user's trajectory by δ_f. 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. Conversely, the approach pose $\boldsymbol{x}_i^{(a)}$ is the pose δ_a away from the user along a trajectory from the robot to the user.

At any given time, the cart solves a sub-optimization to determine the best goal pose $\boldsymbol{x}_i^{(g)}$,

$$\boldsymbol{x}_i^{(g)\,*} = \underset{\boldsymbol{x}_i^{(g)}}{\operatorname{argmin}} \sum_i \|\boldsymbol{e}_s\| + \alpha_2 f_c(\boldsymbol{x}_i^{(g)}) + \alpha_3 f_b(\boldsymbol{x}_i^{(g)}), \quad (4)$$

where $\boldsymbol{e}_s = \boldsymbol{x}_i^{(g)} - \boldsymbol{x}_i^{(d)}$ is the error between the goal pose $\boldsymbol{x}_i(t)$ and the desired pose $\boldsymbol{x}_i^{(d)}$, 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 $\boldsymbol{x}_i$ is determined by querying a heat map constructed from historical human trajectories using 2D kernel density estimation for query point $\boldsymbol{x}_i^{(g)}$ as

$$f_c(\boldsymbol{x}_i^{(g)}) = \frac{1}{|\mathcal{H}|} \sum_{\boldsymbol{h}(t)\in\mathcal{H}} \frac{1}{T_h} \int_{t_0}^{t_f} K_{\mathcal{H}}(\boldsymbol{x}_i^{(g)} - \boldsymbol{h}(t))\,\mathrm{d}t, \quad (5)$$

where $K_{\mathcal{H}}(\boldsymbol{x}) = \frac{1}{2\pi}|\mathbf{H}|^{-1/2} e^{-\frac{1}{2}\boldsymbol{x}^T\mathbf{H}^{-1}\boldsymbol{x}}$ 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 $\boldsymbol{h}(t) \in \mathcal{H}$.

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 (4) as

$$f_b(\boldsymbol{x}_i^{(g)}) = \sum_{\boldsymbol{h}_j\in\mathcal{N}(\boldsymbol{x}_i(t))} \phi_\theta(\boldsymbol{x}_i^{(g)}, \boldsymbol{h}_j) K_b \quad (6)$$

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

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

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

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

$$\underset{i=1,\ldots,N}{\boldsymbol{x}_i^{(l)\,*}} = \underset{\substack{\boldsymbol{x}_i(t)\\ i=1,\ldots,N\\ t_0<t<t_0+T}}{\operatorname{argmin}} \sum_{i=1}^{N} p_i \sum_{t=t_0}^{t_0+T} \|\boldsymbol{e}_i(t)\|_{\boldsymbol{Q}} \quad (7a)$$

$$\text{s.t. } \dot{\boldsymbol{z}}_i(t) = \boldsymbol{f}(\boldsymbol{z}_i(t), \boldsymbol{u}_i(t)), \quad (7b)$$

$$\boldsymbol{z}_i^{(l)}(t_0) = \boldsymbol{z}_i(t_0), \quad (7c)$$

$$\boldsymbol{x}_i^{(l)}(t) \in \mathcal{W}, t_0 < t < t_0 + T, \quad (7d)$$

$$\|\boldsymbol{x}_i^{(l)}(t) - \boldsymbol{x}_j^{(l)}\| > R_r, j \neq i, \quad (7e)$$

$$\|\boldsymbol{x}_i^{(l)}(t) - \boldsymbol{h}_j^{(l)}\| > R_h, \quad (7f)$$

$$\boldsymbol{u}_i(t) \in \mathcal{U}_i, \quad (7g)$$

where $\|\cdot\|_{\boldsymbol{Q}}$ is the vector norm weighted by positive-definite matrix $\boldsymbol{Q}$, and (7e)–(7f) are collision constraints for robots and humans with distances R_r and R_h, respectively.

(7) is recast as a constrained quadratic program (QP) and solved with an iterative linear time-varying model predictive control solver [22], where all matrices are chosen so the QP satisfies the constraint equations in (7).

Fig. 5 shows the overall data flow for both systems. An algorithmic description of the trajectory optimization method is given in Algorithm 1.

Algorithm 1 Multirobot Receding-Horizon Planner

```
procedure GENERATE OPTIMAL TRAJECTORY
    $x_i^{(d)} \leftarrow$ DESIREDPOSE($robot.State$)    ▷ from (2)
    $x_i^{(g)} \leftarrow$ OPTIMIZEPOSE($x_i^{(d)}$)    ▷ from (4)
    while $True$ do
        $x_i^{(l)}(t) \leftarrow$ LOCALWINDOW($x_i^{(G)}(t) = A^*(x_i^{(g)})$)
        $x_i^{(l)\,*}(t) \leftarrow$ QP($e_i(t) = x_i^{(l)} - x_i^{(G)}$)
        $u_i(t) \leftarrow$ LOWLEVELCTRL($x_i^{(l)\,*}(t), u_i^*(t)$)
```

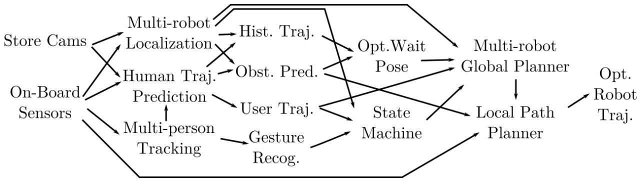


Fig. 5. Data flows in a similar manner for both the simulation and the experimental platforms. Sensor data is used to perform skeleton tracking, trajectory prediction, localization and local obstacle mapping. Skeleton data is used to determine user gestures, which are then used to determine positioning suggestions. Gesture, trajectory and localization data is used to determine each robot's state as shown in Fig. 3. The goal pose is used to plan a global and local optimal trajectory.

IV. Experimental Setup

In this section, we describe the setups for simulation studies performed using a differential drive Turtlebot3 platform, and the experiments conducted using a three-wheeled omni-directional robot. Section IV-A and Section IV-B detail the simulation and experimental setups, respectively.

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 simulated robots are equipped with laser scanners, RGB cameras and Kinect-like depth sensors, and share floor plans and environmental scans.

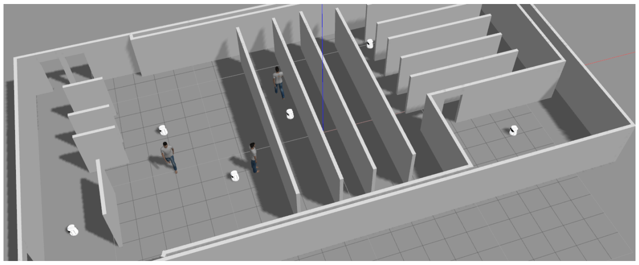


Fig. 6. A supermarket is simulated in Gazebo, and includes up to 40 autonomous shopping carts and simulated humans. Several humans can be actively controlled by the user to test pointing and walking behavior, and the rest are simulated using crowd dynamics.

Each robot's local pose estimated via adaptive Monte Carlo localization. Obstacle avoidance and goal selection is done locally (i.e., separately) in a moving window with on-board sensors and shared maps. We implement (6) as a costmap for a node that calculates (4). Similar to in-store conditions, the robots are given *a priori* information (floor plans and historical trajectories), but perform obstacle avoidance and human tracking online with on-board data. The simulations therefore test state and goal switching, obstacle avoidance and shelf blocking, and multi-robot planning.

B. Human-Following Experiments

We perform experiments with the three-wheel omni-directional robot shown in Fig. 7. Robot models and low-level controllers are adapted from [23] with kinematic model

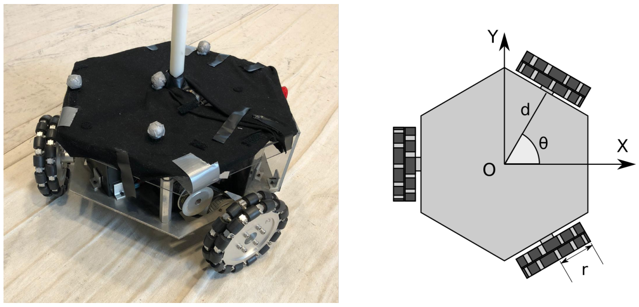


Fig. 7. A three-wheel omni-directional robot is instructed to follow a human target in a lab setting. The robot is fitted with color stereo and RGB-D cameras. Vicon motion tracking cameras are used to validate the experimental result.

$$\dot{\boldsymbol{q}} = \bar{\boldsymbol{B}}(\theta)\boldsymbol{u},\ \bar{\boldsymbol{B}}(\theta) = \begin{bmatrix}\boldsymbol{R}_\theta & 0\\ 0 & 1\end{bmatrix},\ \boldsymbol{R}_\theta = \begin{bmatrix} c\theta & -s\theta \\ s\theta & c\theta\end{bmatrix}, \quad (8)$$

where $\boldsymbol{q} = [x, y, \theta]$ 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 $\boldsymbol{u} = [u_1, u_2, u_3]^T$ 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, $\bar{\boldsymbol{B}}^T(\theta) = \bar{\boldsymbol{B}}^{-1}(\theta)$.

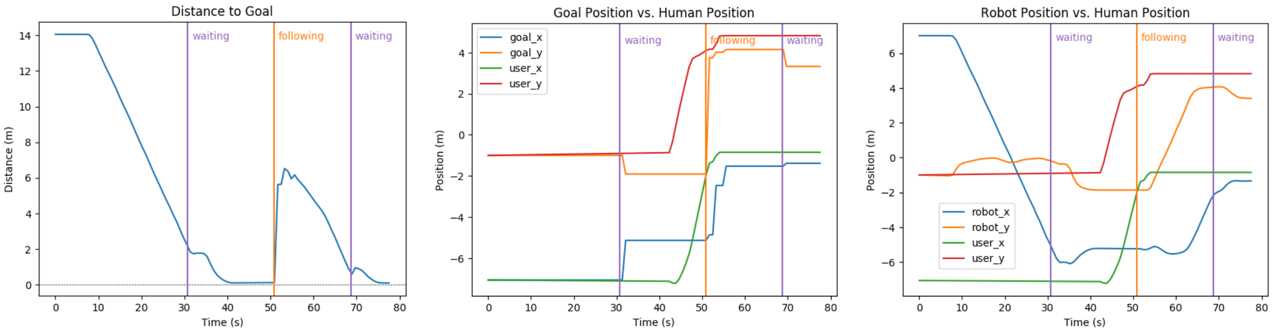
The robot is fitted with RGB-D cameras for navigation and human pose estimation, and Vicon Bonita cameras collect ground truth data using surface-adhered retro-reflective markers. Position feedback is provided by a LabView ground control station. The human subject is asked to draw random targets within a 30 m^2 test area using pointing gestures.

V. Results and Discussion

We report the results under four main categories: State and goal switching, and non-blocking pose selection are demonstrated with single-robot simulations in Sec. V-A and Sec. V-B, respectively; multi-robot planning is demonstrated with large-scale simulations in Sec. V-C; and gesture-based suggestion tracking is demonstrated with lab experiments in Sec. V-D. For consistency, all simulation results are reported using the same map with six parallel aisles. The results of all experiments are reported using ground truth Vicon data, which have sub-millimeter precision at a 100 Hz sample rate for the given test area. Each category is presented in isolation to test edge cases and robustness.

A. State and Goal Switching

To test goal switching, the user is first 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 $R_{w\to f}$, and finally waiting at a second position determined by the user. Related metrics are shown in Fig. 8. Fig. 9 shows the simulation trajectories for a cart following a user through parallel aisles.



(a) (b) (c)

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 $R_{f\to w}$. 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 $R_{f\to w}$ again. (c) Comparison of robot and human positions as influenced by distance-dependent state switching.

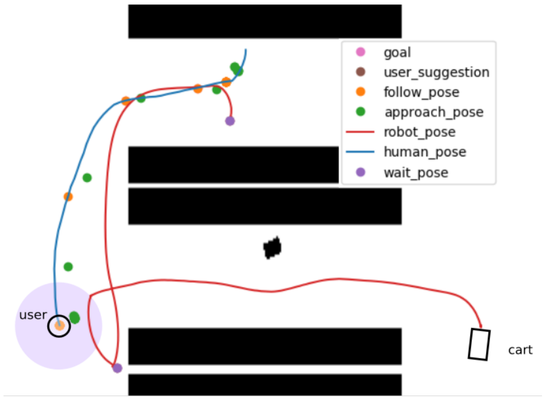


Fig. 9. Results from a user following test, where the cart was asked to follow the user at a distance of $R_{f\to w} = 1$ m. The cart initially follows the user until it reaches $R_{f\to w}$, and then moves to the suggested waiting pose (purple). The robot resumes following once the user moves further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

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 for L_2 distance cost, the pose cost based on historical shopper trajectories, and the blocking cost based on the predicted human path. The minimal cost across the combined costmap provides the optimal waiting position for the robot. The blocking cost converts predicted trajectories into a distance-based penalty that is discounted with time.

The pose cost is constant since the shopper heatmap is roughly constant over long-term observations, while 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. 10 by varying the suggested waiting positions and the human trajectories.

The behaviors described in this section are tunable by the designer using α_2 and α_3 in (4). 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 require different αs.

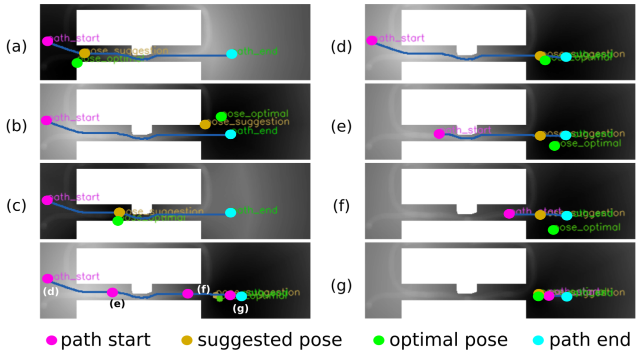
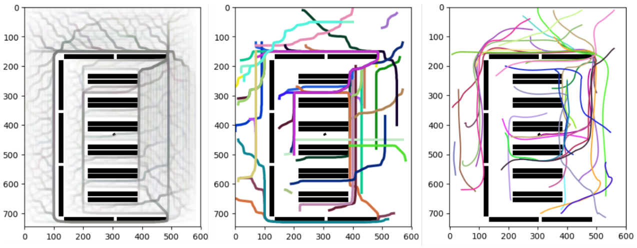


Fig. 10. Optimal non-blocking poses selected for various scenarios. (a-c) User suggestion is varied along the projected path. (d-g) User approaches and passes robot.



(a) (b) (c)

Fig. 11. (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.

C. Multi-Robot Planning

We validate our approach, with simulations of 50 robots and 50 humans. Fig. 11 shows trajectories from 1000 trials with randomized start and goal poses.

We compared the performance of our planner against a scenario where no robots are present, a scenario where robots are present but exhibit pure pursuit behavior, and a scenario where robots plan individual trajectories. The results in Table I look at three metrics: mean robot-user separation, human wait time, and time to goal. The multirobot planner outperforms other methods when looking at wait time and time to goal metrics, whereas the individual planner has lower mean separation.

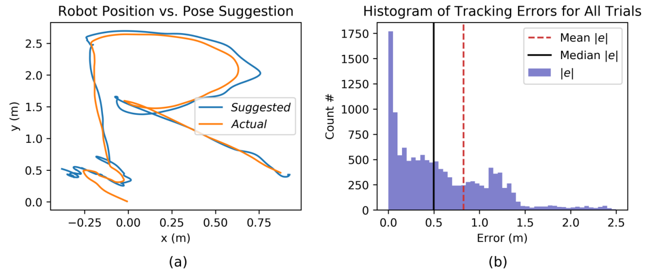
TABLE I

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 conebot at a rate of 20 Hz via wifi for online tracking. Pose suggestions were filtered with a Savitzky-Golay filter (1.5 s window, 3$^{\text{rd}}$ order) to reduce the effect of occasional jumps in finger tracking. Tracking errors are shown in Fig. 12 and were on average around 0.7 m across all trials and were calculated using the instantaneous pose suggestions. Practitioners are advised to filter finger estimates to reduce errors due to jitter.



(a)

(b)

Fig. 12. (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).

VI. Conclusion

In this paper, we presented a framework for autonomous human-following shopping carts in crowded environments with multiple users and robots. Robots switched between following the user, waiting while users browsed the shelves, and approaching users so they could place their items in the basket. We considered historical shopper trajectories to determine optimal waiting positions during browsing. 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 were extracted from RGB-D skeletal estimates. We demonstrated our framework in a supermarket setting using large-scale simulations, and in lab experiments with a three-wheel omni-directional robot. Future work includes the fabrication of multiple prototypes and on-site deployment.

VII. 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 *et al.*, "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 *et al.*, "RoboCart: toward robot-assisted navigation of grocery stores by the visually impaired," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Aug. 2005, pp. 2845–2850.

[3] A. J. R. Neves *et al.*, "A Personal Robot as an Improvement to the Customers' In-store Experience," in *Smart Cities, Green Technologies, and Intelligent Transport Systems*, Cham, 2019, pp. 296–317.

[4] M. J. Islam *et al.*, "Person-following by autonomous robots: A categorical overview," *Int. J. Robot. Res.*, vol. 38, no. 14, pp. 1581–1618, Dec. 2019.

[5] D. Herrera *et al.*, "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.

[6] Y. Sun *et al.*, "Human comfort following behavior for service robots," in *Proc. IEEE Int. Conf. Robot. Biomimet.*, Dec. 2016, pp. 649–654.

[7] B.-J. Lee *et al.*, "Robust Human Following by Deep Bayesian Trajectory Prediction for Home Service Robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 7189–7195.

[8] A. Rudenko *et al.*, "Human motion trajectory prediction: a survey," *Int. J. Robot. Res.*, vol. 39, no. 8, pp. 895–935, Jul. 2020.

[9] Z. Wang *et al.*, "The Simulation of Nonlinear Model Predictive Control for a Human-following Mobile Robot," in *Proc. IEEE Int. Conf. Robot. Biomimet.*, Dec. 2015, pp. 415–422.

[10] J. Ahn *et al.*, "Formation-Based Tracking Method for Human Following Robot," in *2018 15th International Conference on Ubiquitous Robots (UR)*, Jun. 2018, pp. 24–28.

[11] K. Charalampous *et al.*, "Recent trends in social aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 93, pp. 85–104, Jul. 2017.

[12] L. Pang *et al.*, "A human-following approach using binocular camera," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2017, pp. 1487–1492.

[13] J.-G. Lee *et al.*, "A mobile robot which can follow and lead human by detecting user location and behavior with wearable devices," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, Jan. 2016, pp. 209–210.

[14] L. Hu *et al.*, "Follow me Robot-Mind: Cloud brain based personalized robot service with migration," *Future Generation Computer Systems*, vol. 107, pp. 324–332, Jun. 2020.

[15] B. Azari *et al.*, "Commodifying Pointing in HRI: Simple and Fast Pointing Gesture Detection from RGB-D Images," in *2019 16th Conference on Computer and Robot Vision (CRV)*, May 2019, pp. 174–180.

[16] S. Ding *et al.*, "Simultaneous body part and motion identification for human-following robots," *Pattern Recognition*, vol. 50, pp. 118–130, Feb. 2016.

[17] W. Chi *et al.*, "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.

[18] J. Yuan *et al.*, "Fusing Skeleton Recognition With Face-TLD for Human Following of Mobile Service Robots," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–17, 2019.

[19] J. Chen and W.-j. Kim, "A Human-Following Mobile Robot Providing Natural and Universal Interfaces for Control With Wireless Electronic Devices," *IEEE/ASME Trans. Mechatronics*, vol. 24, no. 5, pp. 2377–2385, Oct. 2019.

[20] Z. Cao *et al.*, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *IEEE Trans. Pattern Anal. Machine Intell.*, 2019.

[21] T. Yigit *et al.*, "Wearable IMU-Based Early Limb Lameness Detection for Horses Using Multi-Layer Classifiers," in *Proc. IEEE Conf. Automat. Sci. Eng.*, 2020, pp. 956–961.

[22] M. Edmonds and J. Yi, "A Model Predictive Control Based Iterative Trajectory Optimization Method for Systems with State-Like Disturbances," in *Proc. Amer. Control Conf.*, Jul. 2019, pp. 1635–1640.

[23] K. Hunte and J. Yi, "Collaborative Object Manipulation Through Indirect Control of a Deformable Sheet by a Mobile Robotic Team," in *Proc. IEEE Conf. Automat. Sci. Eng.*, Aug. 2019, pp. 1463–1468.