

# Sampling-Based Path Planning in Highly Dynamic and Crowded Pedestrian Flow

Kuanqi Cai<sup>1</sup>, Graduate Student Member, IEEE, Weinan Chen, Daniel Dugas, Roland Siegwart<sup>2</sup>, Fellow, IEEE, and Jen Jen Chung<sup>3</sup>, Member, IEEE

**Abstract**—Autonomous pedestrian-aware navigation in shared human-robot environments is a challenging problem. Here we consider a common situation in which a large crowd of pedestrians moves together in a limited space. Traditional planners struggle to find collision-free paths in such situations since the free space is limited and always changing. To solve this problem, we proposed a flow map-based RRT\* method (FM-RRT\*) containing a velocity layer and a minimally-intrusive layer. The proposed method models the velocity of the pedestrian flow and the area where the robot is less invasive to pedestrians. Furthermore, we propose an adaptive bias sampling, which drives the robot considering relative velocity, or minimal intrusion, according to the pedestrian flow. The evaluation is conducted in the Crowdbot Challenge simulator. The results show that our method can find a feasible path considering collision risk while simultaneously avoiding intrusive human movement.

**Index Terms**—Mobile robot, path planning, human awareness, human-robot interaction.

## I. INTRODUCTION

**H**UMAN-AWARE path planning for robots among human crowds is both pervasive and challenging [1]. One specific situation frequently noticed is the masses of crowds moving together at a high density in limited space, such as at a public transportation hub or in shopping mall hallways.

Instead of static navigable free spaces, the dense and complex pedestrian flow results in a limited and constantly changing amount of open space. The pedestrian flow is made up of high-density homogeneous masses of the crowd moving together. Therefore, existing planners, such as [2], [3], and [4], struggle to find collision-free paths in such situations because of the highly dynamic and limited open space. This situation

Manuscript received 13 December 2022; revised 13 April 2023 and 28 May 2023; accepted 30 June 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62103179 and in part by the European Commission (EU H2020) project “Crowdbot” under Grant 779942. The Associate Editor for this article was J. Yan. (Corresponding author: Weinan Chen.)

Kuanqi Cai is with the Autonomous Systems Laboratory, ETH Zürich, 8092 Zürich, Switzerland, and also with the Munich Institute of Robotics and Machine Intelligence, Technische Universität München (TUM), 80992 München, Germany (e-mail: kayle.ckq@gmail.com).

Weinan Chen is with the Biomimetic and Intelligent Robotics Laboratory, Guangdong University of Technology, Guangzhou 510006, China (e-mail: chenwn@gdut.edu.cn).

Daniel Dugas and Roland Siegwart are with the Autonomous Systems Laboratory, ETH Zurich, 8092 Zurich, Switzerland (e-mail: dugasd@ethz.ch; rsiegwart@ethz.ch).

Jen Jen Chung is with the Autonomous Systems Laboratory, ETH Zürich, 8092 Zürich, Switzerland, and also with the School of EACS, The University of Queensland, St. Lucia, QLD 4072, Australia (e-mail: jenjen.chung@uq.edu.au).

Digital Object Identifier 10.1109/TITS.2023.3292927

1558-0016 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.  
See <https://www.ieee.org/publications/rights/index.html> for more information.

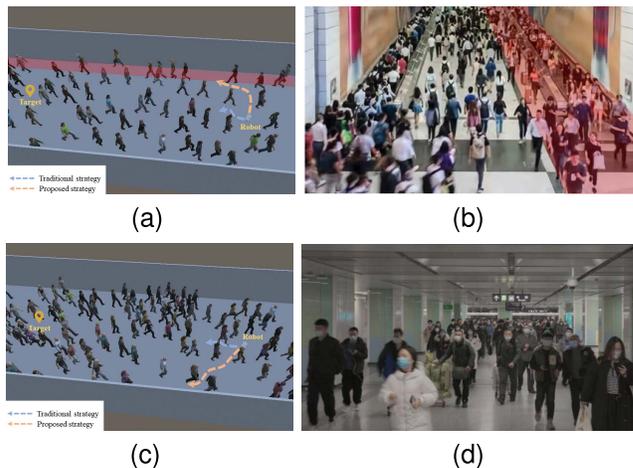


Fig. 1. Illustration of the scenarios. (a): simulation environment of pedestrian flow with different directions. (c): simulation environment of pedestrian flow in the direction opposite to the robot. (b), (d): examples of the corresponding real-life scenarios. In (a) and (b), the direction of the flow of people in the red area is opposite to that in the non-red area. The proposed navigation strategy generates optimal plans based on pedestrian flow, and minimal intrusion on pedestrians is achieved.

can be divided into two typical scenarios, which are shown in Fig.1. The first type is that there is at least a flow of pedestrians in the same direction as the robot, and the second type is that the direction of pedestrian flow is opposite to the robot. We define the positive direction if the pedestrian flow has the same direction as the robot, and the negative direction otherwise, that is, pedestrian flow against the robot’s motion. To complete navigation tasks in a safe and humanly acceptable way in each scenario, it is important to endow robots with fundamental navigation capabilities that meet the requirements of avoiding collisions [5] and minimally intruding [6]. Therefore, we propose a flow map that contains a velocity layer and a minimally-intrusive layer according to the above two scenarios. Observing the fact that the collision risk and intrusion are determined by the relative velocity between robots and surrounding pedestrians, the velocity layer is introduced [7]. The velocity layer aims to provide the robot with information about the area where the pedestrian flow is moving in the same direction. As for the negative direction flows of the above scenarios, the minimally-intrusive layer is proposed to present boundaries of walls and other obstacles. Such areas have a repulsive force on pedestrians [8] and the robot only needs to be cautious of pedestrians on one side. Taking this context into account, we propose the

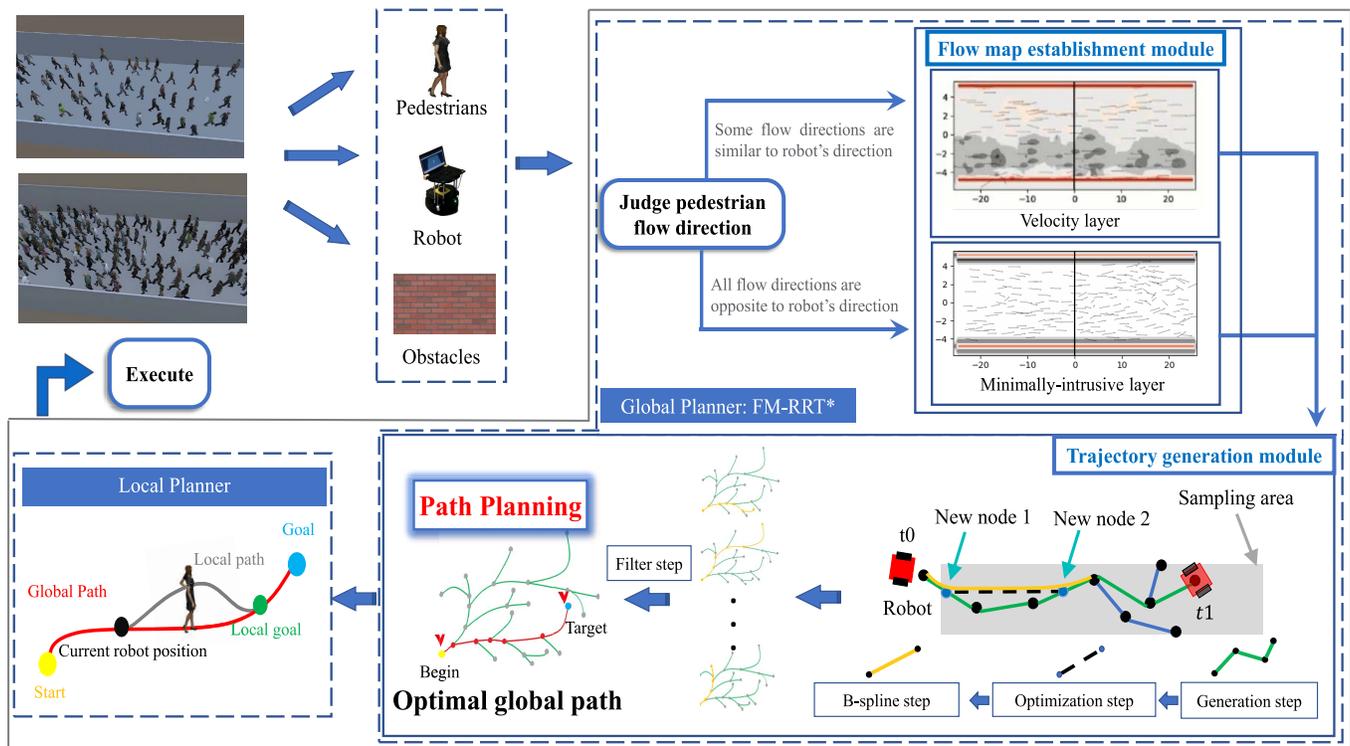


Fig. 2. The system architecture of the proposed path planner. The path planner contains a global planner (FM-RRT\* path planner) and a local planner to generate the optimal path.

flow map-based RRT\* (FM-RRT\*) to drive the robot to plan trajectories in these areas. FM-RRT\* is based on the rapidly-exploring random trees [9] framework. Compared to traditional navigation methods, the proposed method aims to generate optimal strategies depending on different types of pedestrian flow. As exhibited in Fig. 1a, when there is a pedestrian flow in the same direction as the robot, the proposed method aims to plan a path following pedestrians in the same direction as the robot. When the direction of the pedestrian flow is opposite to that of the robot, as shown in Fig. 1c, the proposed strategy makes the robot move alongside obstacles, such as walls. The proposed strategies are designed to avoid facing pedestrians head-on, which achieves minimal intrusion on pedestrians.

In summary, this paper proposes a flow map-based planner to improve robot navigation performance and enhance safety in crowded pedestrian flow. The contributions of this paper are summarized as follows:

- We propose a novel flow map-based model that comprises a velocity layer and a minimally-intrusive layer to model pedestrian flow and non-intrusive areas.
- An RRT-based path planner that incorporates the flow-map model (FM-RRT\*) is introduced, and biased sampling is incorporated to guide the robot towards areas with minimal intrusion or small velocity differences in pedestrian flow.

The remainder of this paper is organized as follows. Section II presents the related work. A series of experiments demonstrating our method's efficiency in comparison to prior

research are described in Section V. Finally, the conclusion of the research study is presented in Section VI.

## II. RELATED WORK

Generating a trajectory with low collision risk and minimal robot intrusion in a dense crowd environment is the premise for realizing the harmonious coexistence of humans and robots [10], [11]. Recently, there have been several studies focused on generating human-aware paths in dynamic environments. Everett et al. [12] proposed GA3C-CADRL, a collision avoidance algorithm that is trained in simulation using deep reinforcement learning (DRL) to reduce the collision risk between multiple agents. Reference [13] solved the distributed motion planning issue by a DRL-FMP hybrid algorithm. Even though this strategy can reduce the probability of collisions in dynamic environments, it does not consider the encroachment of robots on pedestrian movement. Reference [14] discussed smooth path planning using pre-designed steering sets to generate trajectories with smooth changes in direction, while also taking into account the movement of humans to avoid collisions. The authors extended their method to work in the presence of moving obstacles and experimentally evaluated its efficiency in an office environment using a new path evaluation method suitable for wheeled robots. Choi et al. [2] introduced the Mobile robot Collision Avoidance Learning with Path MCAL\_P framework, which used deep reinforcement learning to enable mobile robots to avoid collisions in a decentralized manner. The framework trains robots to navigate environments with dynamic characteristics and optimize their paths with the soft actor-

critic algorithm. With this approach, robots can independently avoid obstacles and reach their targets while maintaining path efficiency without requiring communication. Despite its effectiveness, training in densely crowded environments can be resource-intensive and time-consuming. Sun et al. [3] presented a method for human-aware robot navigation in environments with humans, which uses inverse reinforcement learning and a time-dependent A\* planner. The method takes into account local vision characteristics and learns from human demonstration trajectories to plan paths that comply with human behavior patterns while avoiding disruptions to their personal space and activities. However, the paper only demonstrates the feasibility of this method in an environment with a small number of pedestrians, and its performance in dynamic crowds has not been shown. Zhou et al. [4] proposed the Heterogeneous GAT-based Deep Reinforcement Learning (HGAT-DRL) algorithm to encode the human-robot environment into a graph. This algorithm includes an interactive agent-level representation for objects and incorporates kinodynamic constraints. The proposed method achieves a high success rate for navigation tasks. However, the algorithm's performance in dense crowds cannot be guaranteed. Similar approaches have also been published in [15], [16], [17], [18], and [19] to address this issue. Although the above methods achieve good performance in collision avoidance, they are usually unable to handle crowds of pedestrians.

In recent years, crowd-aware [20] approaches for satisfying navigation in crowded environments have been developed. Paez-Granados et al. [21] presented a force-limited obstacle avoidance controller incorporated into a time-invariant dynamical system (DS), allowing the robot to react instantly to pedestrian contact or abrupt arrival. A learning-based approach called LM-SARL was proposed by Chen et al. [20] to predict human motion. This method can find an optimal path in human-rich areas with the help of an attentive pooling mechanism. Yao et al. [22] introduced a map-based DRL method containing a sensor map and a pedestrian map to achieve crowd-aware navigation with a variety of humans. Another DRL-based path planner [23], which penalizes actions interfering with human movement, is developed for safety planning. Cai et al. [24] suggest a human-aware navigation strategy that can drive the robot to bypass the crowd and avoid collisions. Even though the above methods can navigate areas in crowded environments, they cannot satisfy the situation of high-density pedestrian flow. The reason is that free space is limited and constantly changing.

To safely navigate in pedestrian flow environments, many flow-aware methods have been put forward. Dugas et al. [25] employed a novel pseudo-fluid model of crowd flow and a formalized observation model to achieve flow-aware planning. However, this method does not perform well when there is no positive direction in the scenarios. Henry et al. [26] extended inverse reinforcement learning (IRL) to teach mobile robots to navigate crowded environments by learning human-like navigation behavior based on example paths. This paper proposed a method to estimate dynamic environmental features using Gaussian processes and to learn a cost function that best explains the expert behavior. The approach is demonstrated

using a realistic crowd flow simulator and is capable of guiding the robot to safely navigate when the environment is crowded. Chao et al. [27] introduced an adaptive dynamic programming (ADP) approach, which allows the robot to adjust its motion parameters to interact efficiently with pedestrians and regulate the flow of pedestrians so that it tracks a desired velocity. Fan et al. [7] similarly used deep learning methods to explore whether crowd-flow modeling can be used to help with mapping and localization and social-aware planning by treating crowd flow as a sensory measurement that encodes both the scene's traversability and the social navigation preference. Although the above learning-based methods can have good performance in pedestrian flow navigation, they have fairly high computing requirements. Training is also time-consuming and relies on large amounts of data, which may not always be readily available.

Currently, rapidly exploring random tree-based methods are an efficient traditional class of methods that play a big role in robot navigation. Henderson and Ngo [28], introduced a modified version of the social force model (SFM), called the social intention model (SIM), to reshape the motion primitives (MP) of the rapidly-exploring random tree (RRT) motion planner for socially aware robot navigation. The proposed method significantly enhances safe and social interactions between robots and human agents compared to the typical RRT-embedded MP. Chen et al. [29] designed a planner for pedestrian avoidance that reduces control calibration by considering uncertainties and tire force limits. The method consists of trajectory planning and tracking phases using motion primitives and an LQR-based funnel algorithm. Simulation results confirm the effectiveness of the proposed method in avoiding pedestrian collisions. Suh et al. [30] have proposed a new cost-effective motion-planning algorithm for robots navigating through complex and realistic environments. The algorithm combines RRT\* with a stochastic optimization method called cross-entropy, which helps to efficiently find the minimum cost path. By using two separate trees to grow the search tree nonmyopically, the algorithm can quickly find low-cost paths. A novel framework based on curiosity-driven exploration in [31] helps robots navigate safely with localization precision while considering human comfort in large-scale and crowded environments where sparse landmarks and crowds make localization a significant challenge. Other publications such as [32], [33], [34], and [35], have also utilized rapidly exploring random tree-based methods to tackle similar issues. While the methods mentioned above perform well in collision avoidance and safe navigation, they often struggle with dense crowds that limit the available space for the robot.

Based on the above analysis, it is clear that path planning in dynamic and complex pedestrian flow is still a challenge. In this paper, we propose a flow map-based model that contains a velocity layer and a minimally-intrusive layer to model pedestrian flow and non-intrusive areas. In addition, we added the flow map model to the RRT framework, and we propose bias sampling to drive the robot towards regions in the pedestrian flow with a small velocity difference or minimally-intrusive areas.

### III. PROBLEM FORMULATION

We propose an integrated framework for path planning that ensures safe navigation while avoiding pedestrians in extreme pedestrian flows. The  $j$ th path from node  $x_0 \in \mathbb{R}^3$  to node  $x_I \in \mathbb{R}^3$  is denoted by  $T_j = (x_0, \dots, x_I)$ , and the robot's dynamic model is represented as follows:

$$x_i = g(x_{i-1}, u_{i-1}, Q_t), \quad (1)$$

where  $g(\cdot)$  is the motion model of the robot,  $u_{i-1} \in \mathbb{R}^2$  is the control vector of each node, and  $Q_t \in \mathbb{R}^3$  represents the motion noise. Our objective is to find the optimal trajectory  $T_{opt}$  from a set of candidate trajectories  $T = T_1, \dots, T_J$  by defining a cost function  $\mathbb{C}(\cdot)$  that calculates the cost of each candidate trajectories, and the trajectory with the minimum cost is considered the optimal path  $T_{opt}$ . The formulation is:

$$T_{opt} = \arg \min \sum_{i=0}^I \mathbb{C}(x_i), \quad (2)$$

$$\mathbb{C}(x_i) = \begin{cases} \zeta(x_i) & \text{Case1} \\ \zeta'(x_i) & \text{Case2} \end{cases} \quad (3)$$

As discussed earlier, the pedestrian flow in the environment can be divided into two scenarios, which are in Fig.1. We propose to use different optimal strategies in different scenarios. We divided the total cost function  $\mathbb{C}(\cdot)$  into  $\zeta(x_i)$  and  $\zeta'(x_i)$ , which correspond to the two typical scenarios of pedestrian flow. Specifically,  $\zeta(x_i)$  corresponds to **Case1**, where pedestrians are moving in the same direction as the robot, while  $\zeta'(x_i)$  corresponds to **Case2**, where the pedestrian flow is in the opposite direction.

### IV. METHODOLOGY

#### A. Framework Design

In this section, we propose a flow map-based path planner to plan the trajectory in the area where the pedestrians' flow direction is the same as the robot's direction or in the minimally-intrusive area near the obstacles. The system architecture of the proposed framework is illustrated in Fig.2.

The data streams (e.g., velocity, position, and direction) of pedestrians, the robot, and obstacles obtained in the environment are inputs into the planner. The proposed path planner in this paper can be divided into a local planner and a global planner. We propose FM-RRT\* as the global planner, which includes a **Flow map establishment module** and a **Trajectory generation module**. For the global planner, the direction-judging module is first implemented to judge the similarities and differences between robot direction and pedestrian flow direction. Then the velocity layer or minimally-intrusive layer of the flow map model is built according to the directional similarities and differences. In addition, the RRT\*-based **Trajectory generation module** is introduced, containing a generation step, an optimization step, a B-spline step, and a filter step. The candidates in the same direction as the robot or minimally-intrusive area are yielded by bias sampling. The optimization step aims to optimize the candidates' distance. Because the candidates generated based

on RRT\* are not suitable for robot operation, we use B-spline to further optimize them to fit robot navigation. In addition, we propose a cost function in the filter step to find the optimal global path. Finally, we implemented a local planner to avoid dynamic pedestrians when robots move along the global path. The following section elaborates on the FM-RRT\* planner in detail.

#### B. Flow Map Establishment Module

1) *Judgement of Direction*: When the pedestrians in the environment are in the negative direction, that is, the flow is against the robot's motion, the velocity layer cannot help the robot to reduce collision risk and intrusion into pedestrians. Therefore, we proposed the minimally-intrusive layer, which can safely drive the robot to the destination with less robot intrusion to pedestrians. To judge the direction of the pedestrian flow compared to the robot's motion, we define  $\overrightarrow{r_{t-1}r_t}$  and  $\overrightarrow{c_{t-1}^i c_t^i}$  as the two-dimensional vector of the robot and the  $i$ th crowd, respectively.  $r_t$  ( $c_t^i$ ) is the current position of the robot ( $i$ th crowd), and  $r_{t-1}$  ( $c_{t-1}^i$ ) is the last position of the robot ( $i$ th crowd). The judgment equations are as follows:

$$\exists \overrightarrow{c_{t-1}^i c_t^i} \in \mathbf{C}, \overrightarrow{c_{t-1}^i c_t^i} \cdot \overrightarrow{r_{t-1}r_t} \geq 0, \quad (4)$$

$$\forall \overrightarrow{c_{t-1}^i c_t^i} \in \mathbf{C}, \overrightarrow{c_{t-1}^i c_t^i} \cdot \overrightarrow{r_{t-1}r_t} < 0, \quad (5)$$

where  $\mathbf{C}$  is the set of crowd vectors. When (4) is satisfied, we believe that there is pedestrian flow in the positive direction. Therefore, a velocity layer is established to represent the motion of the pedestrian flow. When (5) is satisfied, we consider that all pedestrians in the environment move in the negative direction. Thus, we apply the minimally-intrusive layer in this case.

2) *JEstablishment of Velocity Layers*: The velocity layer aims to use pedestrian velocity to infer the impact of the flow on an unoccupied area on the map, which can guide the robot to navigate towards areas with a similar velocity. The velocity layer is built based on a two-dimensional Gaussian model, which is shown in Fig. 3. The probability density equation is shown as:

$$\mathcal{N}(q|\mu^i, \Sigma^i) = \frac{1}{2\pi|\Sigma^i|^{1/2}} \exp\{-\frac{1}{2}(q - \mu^i)^T \Sigma^{i-1} (q - \mu^i)\}, \quad (6)$$

where  $\mu^i \in \mathbb{R}^2$  is the mean, and  $\Sigma^i = [\sigma^{i0} 0 \sigma^{i'}]$  is the covariance matrix.  $q \in \mathbb{R}^2$  is a position in the map. The value of  $\mu^i$  and  $\Sigma^i$  are defined as:

$$\mu^i = h_x^i, \quad \sigma^i = v_x^i \Delta t + \rho, \quad \sigma^{i'} = v_y^i \Delta t + \rho, \quad (7)$$

where  $(v_x^i, v_y^i)$  are the  $(x, y)$  velocity components of the  $i$ th pedestrian, and  $h_x^i$  represents the pedestrian's current position.  $\Delta t$  is the time interval and  $\rho$  represents the noise introduced by velocity estimation. In the implementation, we assume that the velocity is accurately estimated and  $\rho = 0$ . The velocity of the surrounding area of pedestrians is calculated by:

$$\mathcal{V}^i(q) = \overrightarrow{v}^i \cdot \mathcal{N}(q|\mu^i, \Sigma^i), \quad (8)$$

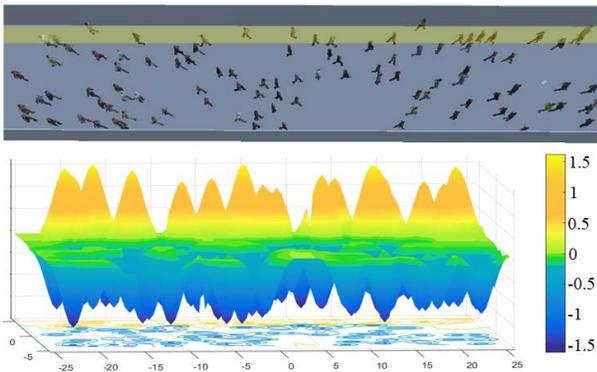


Fig. 3. Schematic diagram of the velocity layer. Colors represent different velocities. The corridor is divided into a yellow area and a blue & green area depending on the direction of pedestrian movement.

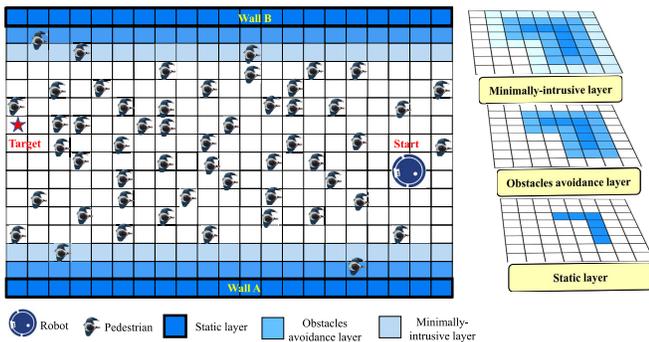


Fig. 4. Schematic diagram of the minimally-intrusive layer. All the pedestrians move in the negative direction. The environment is built by the occupancy grid map [36]. The static layer represents the static obstacles in the environment. The collision layer represents the area where the robot has a high probability of colliding with the obstacles. The minimally-intrusive layer is defined as the area where the robot encounters fewer pedestrians in the environment.

where  $\vec{v}^i$  is the two-dimensional velocity vector of the  $i$ th pedestrian. Because the density of pedestrians in the environment is high, one unoccupied area may be assigned different velocities. Therefore, we used the following function to update the velocity overlap.

$$\mathcal{V}(q) = \max(\|\mathcal{V}^i(q)\|) \cdot \vec{n}^i, \quad i = 1, 2, \dots, I, \quad (9)$$

where  $\vec{n}^i$  is the unit vector of the velocity. It is worth noting that in situations where only a few pedestrians go retrograde in the negative flow, it is not advisable for the robot to follow them. This is because the robot will lose track of them in the dense negative flow.

3) *J Establishment of the Minimally-Intrusive Layers:* The purpose of the minimally-intrusive layer is to guide the robot to navigate along the obstacles to be less intrusive to pedestrians. We cluster different obstacles according to their continuity and create different layers according to different clusters. The schematic diagram is shown in Fig. 4. The obstacle layer is used to mark a dangerous area outside the obstacle that needs to be avoided during path planning. The minimally-intrusive layer is proposed to generate the trajectory with less invasiveness to pedestrians in the environment. The collision layer  $\mathbf{L}_c \subseteq \mathbb{R}^2$  and the minimally-intrusive layer

---

### Algorithm 1 Trajectory Generation of FM-RRT\*

---

**Input:** Map  $\mathbb{M}$ , FlowMap  $\Psi$ , Start  $x_{init}$ , Goal  $x_{goal}$

**Output:** A Trajectory  $T'$  from  $x_{init}$  to  $x_{goal}$

$T.init()$ ;

**for**  $i = 1$  to  $n$  **do do**

$x_s \leftarrow SampleBiasing(\mathbb{M}, \Psi)$ ;

$x_{near} \leftarrow Nearest(x_s, T)$ ;

$x_{new} \leftarrow Steer(x_s, x_{near}, StepSize)$ ;

$x_{newcost} \leftarrow Cost(x_{nearcost} + D(x_{near}, x_{new}))$ ;

**if**  $ObstacleFree(x_{new})$  **then**

$x_{neighbor} \leftarrow FindNearNeighbor(T, x_{new})$ ;

$x_{min} \leftarrow Parent(x_{neighbor}, x_{near}, x_{new})$ ;

$Tree.addNodeEdge(x_{min}, x_{new})$ ;

$Tree \leftarrow Rewire()$ ;

**if**  $x_{new} = x_{goal}$  **then**

Return trajectory  $T$ ;

$T^* = Optimization(T, ObstacleList)$ ;

$T' = B\_Spline(T^*)$ ;

---

$\mathbf{L}_m \subseteq \mathbb{R}^2$  are:

$$\mathbf{L}_c = \{p \mid \|p_o, p\|_2 \leq r\}, \quad \mathbf{L}_c = \cup_{i=1}^I \mathbf{L}_c^i, \quad (10)$$

$$\mathbf{L}_m = \{p \mid r < \|p_o, p\|_2 \leq 2r\}, \quad \mathbf{L}_m = \cup_{i=1}^I \mathbf{L}_m^i, \quad (11)$$

where  $p_o \in \mathbb{R}^2$  is the position of the obstacles, and  $r$  is the radius of the robot's circumcircle.  $\mathbf{L}_c^i, \mathbf{L}_m^i \subseteq \mathbb{R}^2$  are the subset of  $\mathbf{L}_c$  and  $\mathbf{L}_m$ .

### C. Trajectory Generation Module

1) *J Workflow of Trajectory Generation Module:* The trajectory generation module of FM-RRT\* is based on the rapidly-exploring random tree. Alg. 1 shows the procedure of the trajectory generation module. Firstly,  $SampleBiasing(\cdot)$  is utilized to generate the random point in the flow map. After that,  $Nearest(\cdot)$  function searches for the closest point  $x_{near}$  to  $x_s$ .  $Steer(\cdot)$  finds  $x_{new}$  between  $x_{near}$  and  $x_s$  and  $Cost(\cdot)$  calculates the cost of the new node based on the distance factor. Before adding  $x_{new}$  to the tree, the obstacle-free requirement needs to be satisfied, as shown in lines 7 to 11.  $FindNearNeighbor(\cdot)$  and  $Parent(\cdot)$  are used to select the neighbor point and parent node. After connecting two points,  $Rewire(\cdot)$  needs to be used for rewiring the path that is selecting a new parent node and child node. Following the above procedure, we get a preliminary global path. In addition, we use  $Optimization(\cdot)$  to optimize the trajectory by eliminating unnecessary length and we use  $B\_Spline(\cdot)$  to smooth the trajectory to conform to robot operation. Finally, the optimal global path is found by using the cost function to navigate the robot.

2) *J Bias Sampling:* The proposed FM-RRT\* framework is based on a novel biased sampling technique (see line 3 in Alg. 1). By using the biased sampling technique, the robot can sample in the same layer direction as the pedestrians or in a minimally-intrusive layer in the flow map. The sampling

strategy in the flow map is:

$$\begin{cases} X_s = \Psi_{opt} \cap \mathbf{O}_{free}, & \tau > \tau_{thr}, \\ x_s = x_{goal}, & \tau \leq \tau_{thr}, \end{cases} \quad (12)$$

where  $X_s \subseteq \mathbb{R}^2$  is the sampling area,  $\mathbf{O}_{free} \subseteq \mathbb{R}^2$  represents the obstacle-free area, and  $\Psi_{opt}$  is intended to represent the optimal one of all layers  $\Psi \subseteq \mathbb{R}^2$ .  $x_s$  is the sample point and  $x_{goal}$  is the goal point. We sample the goal with probability  $\tau_{thr}$ . Because there may be more than one velocity layer, or minimally-intrusive layer existing in the environments,  $\Psi_{opt}$  is calculated by equation (13) and (14). The cost function  $\eta$  of different pedestrian flow or minimally-intrusive layers is designed to find  $\Psi_{opt}$ :

$$\eta(\Psi_j) = D(r_c, N(\Psi_j)) + \left| \{h'_i | h'_i \in \Psi_j\} \right|, \quad (13)$$

$$\Psi_{opt} = \arg \min_{\Psi_j \subseteq \Psi} \eta(\Psi_j). \quad (14)$$

where  $r_c \in \mathbb{R}^2$  ( $h'_i \in \mathbb{R}^2$ ) is the position of the robot (human).  $N(\cdot)$  finds the point on the  $\Psi$  that is closest to the robot. The cost function  $\eta$  aims to find the layer that is closest to the robot with the fewest pedestrians. Each component of equation 13 is normalized.

As an instance of the optimal layer selection, in Fig. 4, the minimally-intrusive layer of **Wall A** is closer to the robot and has fewer pedestrians than the **Wall B**. Therefore, the minimally-intrusive layer of **Wall A** is the optimal one.

3) *J Cost Function Design*: To find the optimal path  $T_{opt} = \{x_0, \dots, x_I\}$ , two cost functions  $\zeta(x_i)$  and  $\zeta'(x_i)$  in (3) are designed in (15) and (19), respectively. When there is a flow of people in the environment moving in the positive direction, equation (15) is used to find the optimal sampling point whose velocity does not deviate too much from the robot. This cost is able to avoid the discomfort caused by the robot's too-fast or too-slow motion with respect to the pedestrians in the velocity layer.

$$\zeta(x_i) = \xi_{rs} g_{rs} + \mathcal{D}, \quad (15)$$

where  $\mathcal{D}$  is the distance assessment model which is similar to the original RRT method.  $\xi_{rs}$  is the velocity difference, which is designed to avoid discomfort caused by the robot's too-fast or too-slow velocity.  $g_{rs}$  is the impact factor of relative velocities, which are defined as follows:

$$\xi_{rs} = \left\| \vec{v}_r - \mathcal{V}(x_i) \right\|, \quad (16)$$

$$g_{rs} = w_r - (1 - w_r) \cos \theta_{rs}, \quad (17)$$

$$\cos \theta_{rs}(t) = \frac{\vec{v}_s(t) \cdot \vec{v}_r(t)}{\left\| \vec{v}_s(t) \right\| \left\| \vec{v}_r(t) \right\|}. \quad (18)$$

The parameter  $w_r$  reflects the anisotropic nature of relative velocities with the range (0,1).  $v_r$  is the two-dimensional velocity vector of the robot.  $\theta_{rs}(t)$  is the angle between velocity vector  $\vec{v}_{sr}(t)$  and  $\vec{v}_r(t)$ . Each component of equation (15) is normalized. The equation (15) is used to filter the sampling points by minimizing the deviation.

Inspired by the social force model [8], the cost function  $\zeta'(\cdot)$  to find the optimal sampling points is defined as follows.

TABLE I  
PARAMETER SETTING

Parameters	$w_r$	$A_i$	$B_i$	$\kappa$	$K$	$r$
Value	0.9	1.3	12	0.6	0.25	0.5

The variables of this function are normalized.

$$\zeta'(x_i) = \left\| \vec{f}_{rw}^{-1} \right\| + \mathcal{D}, \quad (19)$$

$$\begin{aligned} \vec{f}_{rw} = & (A_i \cdot e^{\frac{r-d_{iw}}{B_i}} + \kappa \cdot g(r-d_{iw})) \cdot \vec{n}_{iw} \\ & - K \cdot g(r-d_{iw}) \cdot (\vec{v}_i \cdot \vec{t}_{iw}) \cdot \vec{t}_{iw}, \end{aligned} \quad (20)$$

where  $d_{iw}$  is the Euclidean distance from the robot to the collision layer.  $A_i$  is the strength of the repulsive forces, and  $B_i$  corresponds to the range of the social forces. Friction parameter  $\kappa$  and  $K$  are constants [8]. We denote  $\vec{n}_{iw}$  as the tangential direction perpendicular to  $\vec{t}_{iw}$ . The function  $g(x)$  is defined by

$$g(x) = \begin{cases} x, & x > 0, \\ 0, & otherwise. \end{cases} \quad (21)$$

The optimal trajectory has the minimum sum of the cost ( $\zeta'(\cdot)$ , or  $\zeta(\cdot)$ ) of each node  $x_i$  on the trajectory.

## V. SIMULATIONS AND RESULTS

### A. Experiment Setup

In this work, we carry out experimental studies<sup>1</sup> based on the CrowdBot Challenge [37]. The CrowdBot Challenge is a simulation benchmark for testing the performance of different navigation planning algorithms in a variety of challenging scenarios. These scenarios are generally classified as low-density and high-density crowds, crowds moving with or against the robot, or both. In this paper, eleven scenarios are designed in the simulator with different human numbers, human movement strategies, pedestrian flow ranges, and flow ratios. The parameter configuration of these scenarios is shown in Table II. Throughout our experiments, we utilized the CrowdBot Challenge simulator to obtain the position of the robot and the motion of pedestrians.

For the simulated crowd, we employed one of three simulation algorithms (ORCA [15] with 0.5s planning horizon, ORCA with 1.5s planning horizon, and Social Forces [16]) as a motion strategy. The flow ratio is defined as the ratio of the area occupied by the pedestrian flow in the positive direction. A value of 0 for the flow ratio means that all pedestrian flow directions in the environment are negative directions. Flow position is defined as the y coordinate of the positive direction pedestrian.

The parameters of Section IV in the simulation are presented in the table below. In our implementation, the design parameters were manually defined. Specifically, we selected these parameters based on a series of offline tests and chose the optimal results to be used in the experiments.

<sup>1</sup>Video demonstration is available at <https://youtu.be/X8rTFtFVXCc>

TABLE II  
EXPERIMENTAL PARAMETER CONFIGURATION

Scenario	SEM1	SEM2	SEC1	SEC2	SEC3	SEC4	SEC5	SEC6	SEC7	SEC8	SEC9
Human Number	100	100	50	109	138	138	138	165	200	200	350
Strategy	ORCA1.5s	ORCA1.5s	ORCA 0.5s	ORCA 1.5s	SocialForce	ORCA0.5s	ORCA1.5s	ORCA0.5s	Social force	ORCA1.5s	ORCA1.5s
Flow Position	[-3,-5]	None	[3, 5]	[4, 5]	None	None	None	[-2.5, -5]	[-2.5, -4]	None	None
Flow Ratio	20.00%	0	20%	10%	0	0	0	25%	15%	0	0

TABLE III

MODULE FEASIBILITY VERIFICATION. AFFECTED TIMES REPRESENTS THE TIMES THAT THE ROBOT AFFECTS THE NUMBER OF PEDESTRIANS. PROX REPRESENTS CROWD PROXIMITY. NBRREAC REPRESENTS CROWD EFFECTS THAT RELATE THE SPEED OF THE ROBOT'S NEIGHBORS TO THE SPEED OF THE ENTIRE CROWD. NBRVEL IS CALCULATED BY COMPARING THE ANGULAR VELOCITY OF THE ENTIRE CROWD RELATIVE TO THE ROBOT'S NEIGHBORS

Scenario	Algorithm	Path Efficiency			Crowd Awareness					
		Time (s)	Length (m)	Velocity (m/s)	Collision Times	Duration of Collisions	Affected Times	Prox	NBRreac	NBRvel
SEM1	RVO	79.90	41.68	0.522	0	0.00%	55	0.825	0.183	0.892
	RVO+Velocity layer	53.70	41.90	0.780	0	0.00%	8	0.801	0.298	0.942
	Baseline	40.70	40.05	0.984	2	0.50%	46	0.647	0.121	0.930
	Baseline+Velocity layer	44.80	44.27	0.988	1	0.10%	5	0.759	0.330	0.984
	RVO	138.00	54.68	0.469	5	0.40%	69	0.661	0.101	0.777
SEM2	RVO+Minimally-intrusive layer	59.48	43.70	0.735	0	0.00%	12	0.726	0.167	0.926
	Baseline	41.60	40.06	0.963	9	2.20%	42	0.793	0.108	0.891
	Baseline+Minimally-intrusive layer	45.44	44.99	0.990	0	0.00%	9	0.767	0.117	0.913

### B. Metrics Setup

We focus on nine metrics, which can be split into two categories: path efficiency and crowd awareness. In terms of path efficiency metrics, we calculate and compare the time consumption of different algorithms, the length of the robot's trajectory, and the average velocity of the robot. The latter six metrics focus on the effect the robot has on the crowd. We evaluate each collision by using the simulator's report of the colliding human body component as the collision metric, which is based on the contribution of the pedestrians' inertia to the kinetics of the impact [37]. We calculate the collision times during the navigation tasks. Another collision metric is the duration of collisions, which is the ratio of the accumulation of collision time to the navigation time. In addition, this paper also records the NBRreac, NBRvel [37], and affected times to indicate how the robot under different algorithms affects humans. Since humans have their own navigation strategy, affected times are used to present the frequency of the human's motion changes due to the influence of the robot. We only recorded the number of humans directly affected by the robot. NBRreac is defined by  $NBRreac = v_{Neighbors}/v_{All}$  where  $v_{neighbors}$  is the average speed of the robot neighbors (within a 1m range) and  $v_{All}$  is the average speed of the whole crowd. The higher the neighbors' velocity, the bigger the value. NBRvel is calculated by  $NBRvel = \omega_{All}/\omega_{Neighbors}$ .  $\omega_{Neighbors}$  is the angular velocity of the robot's neighbors and  $\omega_{All}$  is that of the crowd. The more the neighbors rotate, the smaller the NBRvel will be. When the robot's influence on pedestrians is small, the NBRvel (NBRreac) value is close to 1. For the sake of fully representing the influence of the robot, we also include the proximity metric  $Prox = 1 - \frac{1}{t_{final}} \sum_{t=0}^{t_{final}} \frac{d_{min}(t)}{R}$ , which is the metric presenting crowd proximity in the Crowdbot Challenge simulator [37]. In this function,  $d_{min}$  is the minimum distance from the robot to the crowd and  $R$  is the range in meters.

### C. Evaluation of Velocity Layer and Minimally-Intrusive Layer

In order to test the proposed layers, we perform experiments on the velocity layer and minimally-intrusive layer,

respectively. We use Reciprocal Velocity Obstacles (RVO) [38] as well as a baseline approach. The baseline algorithm acts as a benchmark, directing the robot to move straight towards the goal without local collision avoidance. This causes the robot to move continuously in a straight line towards the goal. We designed the scenario (SEM1) with the positive direction to test the velocity layer and the scenario (SEM2) without the positive direction to test the minimally-intrusive layer. The velocity layer and minimally-intrusive layer are added in the RRT\* path planner framework for the global path, respectively. We compared the experimental results with and without the designed layers to observe the difference. The experiment is repeated ten times for each algorithm for each scenario.

The experimental results are shown in Table III. According to the efficiency of robot navigation tasks, the RVO algorithm with the proposed layers has a faster average velocity, shorter time, and shorter trajectory length than the pure RVO algorithm. Despite needing to travel a longer path than the baseline algorithm, our method moves at a faster velocity and results in significantly fewer collisions.

In terms of crowd awareness, collision times, duration of collisions, and affected times, algorithms with the addition of the velocity layer or the minimum-intrusive layer perform better than those without these layers. This means that these two layers have a good performance in crowd safety and reduce the collision risk with the crowd. In addition, compared with the RVO method and the baseline method, the proposed layer has better performance in terms of NBRreac and NBRvel, which are closer to 1. The crowd is less responsive to the movement of the robot, indicating that the proposed layers allow the robot to have less impact on the pedestrians around it.

### D. Contrast Experiment

In order to evaluate the feasibility and superiority of the comparison algorithm, we implemented the proposed method in the Crowdbot Challenge and compared its performance to those of traditional planners like the Dynamic Window Approach (DWA) [39] and RVO. Furthermore, we ran two versions of the FM-RRT\* planner. The first version is

TABLE IV  
COMPARATIVE EXPERIMENTAL METRICS

Scenario	Algorithm	Time (s)	Path Efficiency		Collision Times	Duration of Collisions	Crowd Awareness			
			Length (m)	Velocity (m/s)			Affected Times	Prox	NBRreac	NBRvel
SEC1	Baseline	41.80	40.08	0.959	10	2.40%	24	0.486	0.073	0.912
	DWA	82.00	77.83	0.949	1	0.10%	37	0.682	0.078	0.863
	RVO	97.70	47.68	0.488	1	0.10%	51	0.683	0.099	0.815
	FM-RRT*+Baseline	47.10+1.43	46.63	0.990	0	0.00%	7	0.695	0.296	1.058
	FM-RRT*+RVO	48.38+1.37	42.74	0.883	0	0.00%	6	0.546	0.664	1.031
SEC2	Baseline	41.40	40.05	0.967	7	1.70%	18	0.729	0.080	0.849
	DWA	56.50	53.14	0.941	4	0.70%	27	0.719	0.080	0.859
	RVO	58.60	42.56	0.726	2	0.30%	31	0.731	0.278	0.969
	FM-RRT*+Baseline	45.58+1.82	45.04	0.988	1	0.10%	10	0.763	0.280	1.040
	FM-RRT*+RVO	54.95+1.63	42.34	0.771	0	0.00%	12	0.691	0.555	0.934
SEC3	Baseline	41.40	40.09	0.968	7	1.70%	34	0.802	0.007	0.943
	DWA	69.40	64.92	0.935	3	0.40%	30	0.714	0.070	0.839
	RVO	105.00	49.02	0.467	4	0.40%	57	0.798	0.102	0.797
	FM-RRT*+Baseline	44.50+1.92	43.95	0.987	0	0.00%	9	0.874	0.152	0.956
	FM-RRT*+RVO	51.88+1.96	42.5	0.819	0	0.00%	7	0.724	0.112	0.890
SEC4	Baseline	41.20	40.02	0.971	6	1.50%	37	0.832	0.087	0.936
	DWA	54.10	50.83	0.940	3	0.60%	26	0.796	0.111	0.817
	RVO	151.70	63.11	0.416	2	0.13%	106	0.749	0.172	0.857
	FM-RRT*+Baseline	43.64+2.12	43.16	0.989	0	0.00%	8	0.834	0.196	1.008
	FM-RRT*+RVO	49.92+2.01	42.52	0.852	0	0.00%	9	0.636	0.202	0.937
SEC5	Baseline	41.00	40.05	0.977	4	1.00%	39	0.838	0.101	0.955
	DWA	65.20	56.52	0.867	4	0.60%	24	0.615	0.103	0.764
	RVO	109.60	45.56	0.416	1	0.10%	93	0.785	0.179	0.903
	FM-RRT*+Baseline	48.20+1.97	47.75	0.991	0	0.00%	13	0.739	0.156	0.968
	FM-RRT*+RVO	45.10+2.03	44.65	0.990	0	0.00%	8	0.666	0.419	0.996
SEC6	Baseline	42.50	41.03	0.965	7	1.60%	40	0.799	0.080	0.959
	DWA	59.30	56.32	0.950	3	0.50%	16	0.824	0.096	0.812
	RVO	73.20	49.26	0.673	1	0.10%	43	0.738	0.149	0.841
	FM-RRT*+Baseline	43.00+2.27	42.55	0.990	0	0.00%	12	0.826	0.265	0.968
	FM-RRT*+RVO	56.12+2.41	42.19	0.752	0	0.00%	10	0.821	0.606	0.909
SEC7	Baseline	42.80	41.09	0.960	9	2.10%	57	0.763	0.092	0.986
	DWA	58.00	53.11	0.916	3	0.50%	24	0.843	0.139	0.849
	RVO	170.00	76.78	0.452	8	0.50%	114	0.807	0.145	0.782
	FM-RRT*+Baseline	46.00+2.77	45.21	0.983	1	0.20%	17	0.846	0.441	0.995
	FM-RRT*+RVO	56.30+2.65	42.52	0.755	0	0.00%	16	0.842	0.289	0.921
SEC8	Baseline	43.10	40.05	0.929	20	4.60%	78	0.8	0.088	0.909
	DWA	—	—	—	—	—	—	—	—	—
	RVO	—	—	—	—	—	—	—	—	—
	FM-RRT*+Baseline	44.35+2.13	43.43	0.979	3	0.70%	39	0.852	0.086	0.910
	FM-RRT*+RVO	—	—	—	—	—	—	—	—	—
SEC9	Baseline	44.50	40.01	0.899	30	6.70%	99	0.845	0.095	0.890
	DWA	—	—	—	—	—	—	—	—	—
	RVO	—	—	—	—	—	—	—	—	—
	FM-RRT*+Baseline	44.58+2.16	43.24	0.970	6	1.50%	54	0.857	0.098	0.909
	FM-RRT*+RVO	—	—	—	—	—	—	—	—	—

FM-RRT\*+Baseline. This method implies that there is no local collision avoidance, while the baseline velocity tends to match that of the crowd wherever feasible. The second version, FM-RRT\*+RVO, does not directly execute the baseline velocity but rather modifies it depending on the local path planner RVO. We designed nine scenarios for the experiment, which are visualized in Fig.5.

As can be seen in Fig.5, in order to increase the challenge, we designed the region of positive pedestrian flow to be far away from the initial position of the robot, and the proportion of flow in the positive direction was smaller than that of the flow in the negative direction. In the second and fifth rows, the velocity layers of the flow map visually show the velocity distribution in the environment, which is consistent with the trend of the pedestrian flow in the scenarios. From a qualitative perspective, the trajectories of FM-RRT\*+RVO and FM-RRT\*+Baseline are smoother and reduce unnecessary detours that are generated when the robot avoids walking toward humans.

The simulation results are shown in Table IV. The proposed method still has good performance in the field of crowd awareness. To be more specific, from the perspective of collisions, FM-RRT\*+Baseline and FM-RRT\*+RVO have lower collision times and the duration of collisions regardless of whether the flow of people is moving in the positive direction or the negative direction. In terms of the influence of the robot on its neighbors, our algorithm achieves the least influence on pedestrians indicated by NBRreac and NBRvel, and the smallest affected times among all the algorithms. Because the robot rushes into the opposite pedestrian flow under the control of other algorithms (Baseline, DWA, and RVO), the trajectories of its neighbors have to update in order to avoid collision with the robot, which will cause a reduction in their velocity and even change their direction. Therefore, the NBRreac and NBRvel of other algorithms are lower and far from 1. In addition, the affected times are higher.

Our algorithm comprises the global and the local planning, and we recorded the computation times for each component in

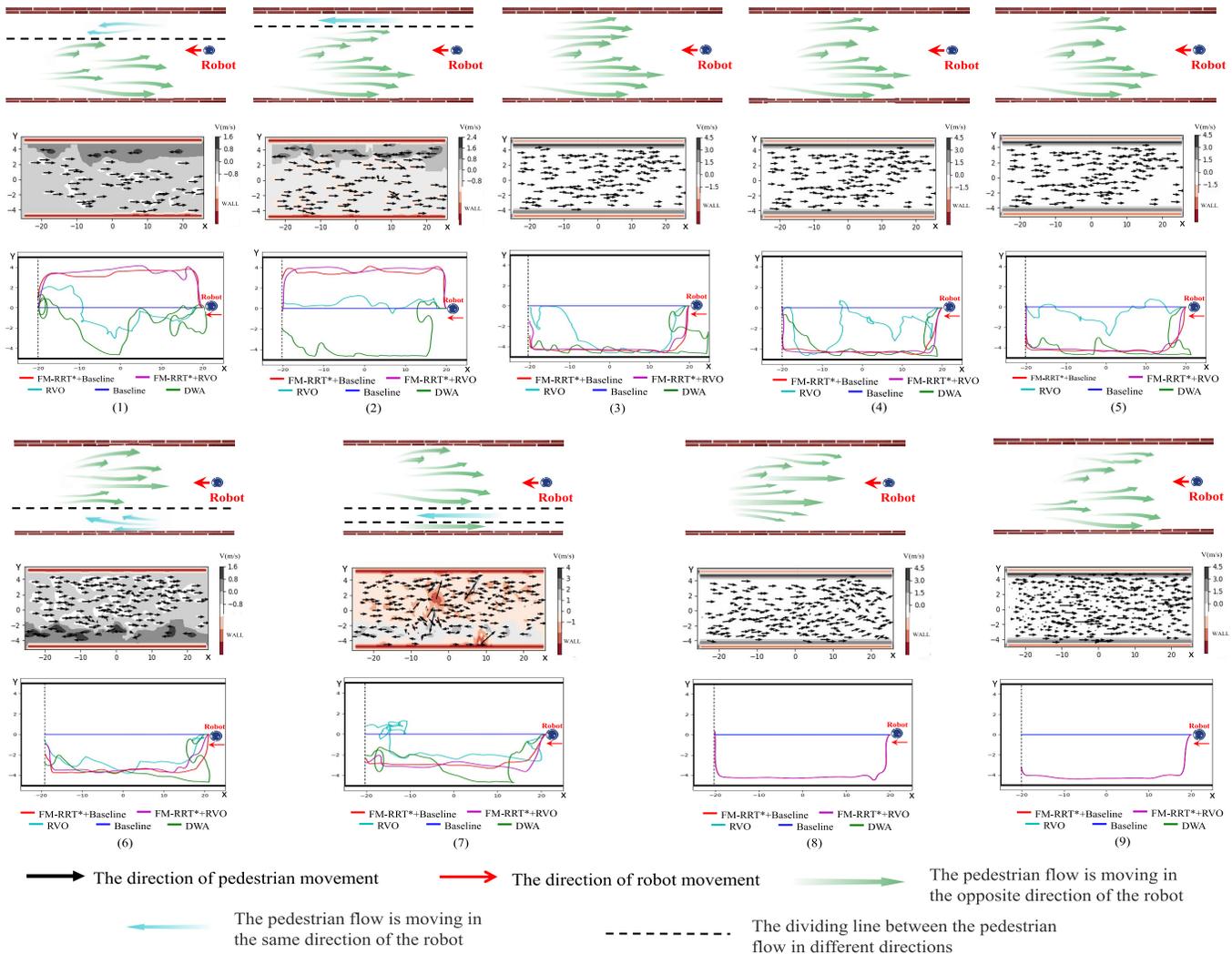


Fig. 5. Comparative experiments. (1)SEC1, (2)SEC2, (3)SEC3, (4)SEC4, (5)SEC5, (6)SEC6, (7)SEC7, (8)SEC8, (9)SEC9. The second and fifth rows are the flow maps generated by FM-RRT\* with arrows indicating the position and direction of the pedestrian. In scenario (1), (2), (6), and (7), the flow maps represent the velocity layers, while in (3), (4), (5), (8), and (9), they illustrate the minimally-intrusive layers. The third and sixth rows display the trajectories of different planners.

Table IV. The first value of the “Time (s)” metrics represents the duration of local planning and execution, which occurs simultaneously during the navigation process. The second value represents the time taken for global planning. Because the time consumption of planning is small, the velocity of the pedestrian flow does not change significantly during the time delay between planning and execution. Therefore, the initially planned global path remains applicable throughout the execution. As for efficiency, although the focus of this study is to reduce collision risk and invasive pedestrians in crowded environments, our planner can obtain comparable efficiency to other obstacle avoidance algorithms.

Furthermore, we designed highly dynamic and dense environments (SEC8 and SEC9). Because the free areas in these two highly dynamic scenarios are limited and constantly changing, DWA, RVO algorithms, and FM-RRT\*+RVO, which use RVO as a local planner, cannot complete the robot navigation task. However, the baseline algorithm and FM-RRT\*+Baseline can complete the navigation task. Although

FM-RRT\*+Baseline is not the most path-efficient algorithm due to the consideration of pedestrians, it performs better than the baseline in terms of crowd awareness and average velocity.

## VI. CONCLUSION

In this paper, we proposed a flow map-based planner (FM-RRT\*) to achieve navigation in highly dynamic and limited open spaces, such as railway stations and hallways. Our proposed approach introduces a flow map model that consists of velocity layers and a minimally-intrusive layer, which considers pedestrian velocity distribution and the encroachment on human movement. We also present a biased sampling strategy based on the flow map, which enables robot planning in the above layers. If there is a pedestrian flow moving in the same direction as desired by the robot, the robot goes with the flow as enabled by the velocity layer, resulting in a low collision risk and minimal invasiveness to pedestrians. In the absence of such flow, the robot can reach its destination safely without encroaching on pedestrians’ movement, with

the help of the minimally-intrusive layer. Our proposed FM-RRT\* method, considering two layers, can effectively reduce collision probability and intrusion into pedestrian movement around counter-flows in highly dynamic environments with limited open space.

In the future, we aim to further enhance the performance of our system in unstructured and complex scenarios. We will improve the FM model by combining human awareness. Additionally, we plan to address the impact of noise, specifically addressing measurement and motion noise resulting from robot motion systems and sensor uncertainty. To achieve this, we intend to integrate a filter-based approach into our system, which will help to reduce noise. Furthermore, we will include algorithms for accurately locating dense crowds and recognizing pedestrians. We will then integrate these algorithms and evaluate the proposed method in a real-world setting.

#### ACKNOWLEDGMENT

The authors are grateful to Dr. Luis F. C. Figueredo (Munich Institute of Robotics and Machine Intelligence (MIRMI) at the TUM) for his help with the revision.

#### REFERENCES

- [1] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robot. Auto. Syst.*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013, doi: [10.1016/j.robot.2013.05.007](https://doi.org/10.1016/j.robot.2013.05.007).
- [2] J. Choi, G. Lee, and C. Lee, "Reinforcement learning-based dynamic obstacle avoidance and integration of path planning," *Intell. Service Robot.*, vol. 14, pp. 1–15, Nov. 2021, doi: [10.1007/s11370-021-00387-2](https://doi.org/10.1007/s11370-021-00387-2).
- [3] S. Sun, X. Zhao, Q. Li, and M. Tan, "Inverse reinforcement learning-based time-dependent A\* planner for human-aware robot navigation with local vision," *Adv. Robot.*, vol. 34, no. 13, pp. 888–901, Jul. 2020, doi: [10.1080/01691864.2020.1753569](https://doi.org/10.1080/01691864.2020.1753569).
- [4] Z. Zhou et al., "Navigating robots in dynamic environment with deep reinforcement learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 25201–25211, Dec. 2022, doi: [10.1109/TITS.2022.3213604](https://doi.org/10.1109/TITS.2022.3213604).
- [5] Z. Zhao, M. Jin, E. Lu, and S. X. Yang, "Path planning of arbitrary shaped mobile robots with safety consideration," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16474–16483, Sep. 2022, doi: [10.1109/TITS.2021.3128411](https://doi.org/10.1109/TITS.2021.3128411).
- [6] K. Cai, C. Wang, J. Cheng, C. W. De Silva, and M. Q.-H. Meng, "Mobile robot path planning in dynamic environments: A survey," 2020, *arXiv:2006.14195*.
- [7] T. Fan, D. Wang, W. Liu, and J. Pan, "Crowd-driven mapping, localization and planning," in *Proc. Int. Symp. Exp. Robot.* Cham, Switzerland: Springer, 2020, pp. 354–368, doi: [10.1007/978-3-030-71151-1\\_32](https://doi.org/10.1007/978-3-030-71151-1_32).
- [8] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation situations," *Pedestrian Evacuation Dyn.*, vol. 21, no. 2, pp. 21–58, 2002.
- [9] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Dept. Comput. Sci., Iowa State Univ., Ames, IA, USA, Tech. Rep. TR 98-11, Oct. 1998.
- [10] Z. Zheng, C. Cao, and J. Pan, "A hierarchical approach for mobile robot exploration in pedestrian crowd," *IEEE Robot. Autom. Lett.*, vol. 7, no. 1, pp. 175–182, Jan. 2022, doi: [10.1109/LRA.2021.3118078](https://doi.org/10.1109/LRA.2021.3118078).
- [11] M. Nishimura and R. Yonetani, "L2B: Learning to balance the safety-efficiency trade-off in interactive crowd-aware robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 11004–11010, doi: [10.1109/IROS45743.2020.9341519](https://doi.org/10.1109/IROS45743.2020.9341519).
- [12] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10357–10377, 2021, doi: [10.1109/ACCESS.2021.3050338](https://doi.org/10.1109/ACCESS.2021.3050338).
- [13] S. H. Semnani, H. Liu, M. Everett, A. de Ruiter, and J. P. How, "Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3221–3226, Apr. 2020, doi: [10.1109/LRA.2020.2974695](https://doi.org/10.1109/LRA.2020.2974695).
- [14] Y. Suzuki, S. Thompson, and S. Kagami, "Smooth path planning with pedestrian avoidance for wheeled robots: Implementation and evaluation," in *Proc. 4th Int. Conf. Auto. Robots Agents*, Feb. 2009, pp. 657–662, doi: [10.1109/ICARA.2000.4803910](https://doi.org/10.1109/ICARA.2000.4803910).
- [15] N. Pérez-Higueras, F. Caballero, and L. Merino, "Learning human-aware path planning with fully convolutional networks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5897–5902, doi: [10.1109/ICRA.2018.8460851](https://doi.org/10.1109/ICRA.2018.8460851).
- [16] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 874–883, Oct. 2007, doi: [10.1109/TRO.2007.904911](https://doi.org/10.1109/TRO.2007.904911).
- [17] J. V. Gómez, N. Mavridis, and S. Garrido, "Fast marching solution for the social path planning problem," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1871–1876, doi: [10.1109/ICRA.2014.6907105](https://doi.org/10.1109/ICRA.2014.6907105).
- [18] H. Shi, L. Shi, M. Xu, and K. Hwang, "End-to-end navigation strategy with deep reinforcement learning for mobile robots," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2393–2402, Apr. 2020, doi: [10.1109/TII.2019.2936167](https://doi.org/10.1109/TII.2019.2936167).
- [19] R. C. Luo and C. Huang, "Human-aware motion planning based on search and sampling approach," in *Proc. IEEE Workshop Adv. Robot. Social Impacts (ARSO)*, Jul. 2016, pp. 226–231, doi: [10.1109/ARSO.2016.7736286](https://doi.org/10.1109/ARSO.2016.7736286).
- [20] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6015–6022, doi: [10.1109/ICRA.2019.8794134](https://doi.org/10.1109/ICRA.2019.8794134).
- [21] D. Paez-Granados, V. Gupta, and A. Billard, "Unfreezing social navigation: Dynamical systems based compliance for contact control in robot navigation," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 8368–8374, doi: [10.1109/ICRA46639.2022.9811772](https://doi.org/10.1109/ICRA46639.2022.9811772).
- [22] S. Yao, G. Chen, Q. Qiu, J. Ma, X. Chen, and J. Ji, "Crowd-aware robot navigation for pedestrians with multiple collision avoidance strategies via map-based deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 8144–8150, doi: [10.1109/IROS51168.2021.9636579](https://doi.org/10.1109/IROS51168.2021.9636579).
- [23] S. Matsuzaki and Y. Hasegawa, "Learning crowd-aware robot navigation from challenging environments via distributed deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 4730–4736, doi: [10.1109/ICRA46639.2022.9812011](https://doi.org/10.1109/ICRA46639.2022.9812011).
- [24] K. Cai, W. Chen, C. Wang, S. Song, and M. Q.-H. Meng, "Human-aware path planning with improved virtual Doppler method in highly dynamic environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 20, no. 2, pp. 1304–1321, Apr. 2023, doi: [10.1109/TASE.2022.3175039](https://doi.org/10.1109/TASE.2022.3175039).
- [25] D. Dugas, K. Cai, O. Andersson, N. Lawrance, R. Siegwart, and J. J. Chung, "FlowBot: Flow-based modeling for robot navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 8799–8805, doi: [10.1109/IROS47612.2022.9981407](https://doi.org/10.1109/IROS47612.2022.9981407).
- [26] P. Henry, C. Vollmer, B. Ferris, and D. Fox, "Learning to navigate through crowded environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 981–986, doi: [10.1109/ROBOT.2010.5509772](https://doi.org/10.1109/ROBOT.2010.5509772).
- [27] C. Jiang, Z. Ni, Y. Guo, and H. He, "Learning human-robot interaction for robot-assisted pedestrian flow optimization," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 49, no. 4, pp. 797–813, Apr. 2019, doi: [10.1109/TSMC.2017.2725300](https://doi.org/10.1109/TSMC.2017.2725300).
- [28] M. Henderson and T. D. Ngo, "RRT-SMP: Socially-encoded motion primitives for sampling-based path planning," in *Proc. 30th IEEE Int. Conf. Robot Hum. Interact. Commun. (RO-MAN)*, Aug. 2021, pp. 330–336, doi: [10.1109/RO-MAN50785.2021.9515460](https://doi.org/10.1109/RO-MAN50785.2021.9515460).
- [29] Y. Chen, H. Peng, and J. W. Grizzle, "Fast trajectory planning and robust trajectory tracking for pedestrian avoidance," *IEEE Access*, vol. 5, pp. 9304–9317, 2017, doi: [10.1109/ACCESS.2017.2707322](https://doi.org/10.1109/ACCESS.2017.2707322).
- [30] J. Suh, J. Gong, and S. Oh, "Fast sampling-based cost-aware path planning with nonmyopic extensions using cross entropy," *IEEE Trans. Robot.*, vol. 33, no. 6, pp. 1313–1326, Dec. 2017, doi: [10.1109/TRO.2017.2738664](https://doi.org/10.1109/TRO.2017.2738664).
- [31] K. Cai, W. Chen, C. Wang, H. Zhang, and M. Q.-H. Meng, "Curiosity-based robot navigation under uncertainty in crowded environments," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 800–807, Feb. 2023, doi: [10.1109/LRA.2022.3232270](https://doi.org/10.1109/LRA.2022.3232270).
- [32] H. Ma, F. Meng, C. Ye, J. Wang, and M. Q.-H. Meng, "Bi-Risk-RRT based efficient motion planning for autonomous ground vehicles," *IEEE Trans. Intell. Vehicles*, vol. 7, no. 3, pp. 722–733, Sep. 2022, doi: [10.1109/TIV.2022.3152740](https://doi.org/10.1109/TIV.2022.3152740).

- [33] Z. Tahir, A. H. Qureshi, Y. Ayaz, and R. Nawaz, "Potentially guided bidirectionalized RRT\* for fast optimal path planning in cluttered environments," *Robot. Auto. Syst.*, vol. 108, pp. 13–27, 2018, doi: [10.1016/j.robot.2017.12.008](https://doi.org/10.1016/j.robot.2017.12.008).
- [34] Y. Chen, M. Liu, and L. Wang, "RRT\* combined with GVO for real-time nonholonomic robot navigation in dynamic environment," in *Proc. IEEE Int. Conf. Real-Time Comput. Robot. (RCAR)*, Aug. 2018, pp. 479–484, doi: [10.1109/RCAR.2018.8621737](https://doi.org/10.1109/RCAR.2018.8621737).
- [35] K. Cai, C. Wang, S. Song, H. Chen, and M. Q.-H. Meng, "Risk-aware path planning under uncertainty in dynamic environments," *J. Intell. Robot. Syst.*, vol. 101, no. 3, pp. 1–15, Mar. 2021, doi: [10.1007/s10846-021-01323-3](https://doi.org/10.1007/s10846-021-01323-3).
- [36] D. Meyer-Delius, M. Beinhofer, and W. Burgard, "Occupancy grid models for robot mapping in changing environments," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2012, pp. 2024–2030, doi: [10.1609/aaai.v26i1.8377](https://doi.org/10.1609/aaai.v26i1.8377).
- [37] F. Grzeskowiak et al., "Crowd against the machine: A simulation-based benchmark tool to evaluate and compare robot capabilities to navigate a human crowd," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 3879–3885, doi: [10.1109/ICRA48506.2021.9561694](https://doi.org/10.1109/ICRA48506.2021.9561694).
- [38] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1928–1935, doi: [10.1109/ROBOT.2008.4543489](https://doi.org/10.1109/ROBOT.2008.4543489).
- [39] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997, doi: [10.1109/100.580977](https://doi.org/10.1109/100.580977).



**Kuanqi Cai** (Graduate Student Member, IEEE) received the B.E. degree from Hainan University in 2018 and the M.E. degree from the Harbin Institute of Technology in 2021. He is currently pursuing the Ph.D. degree with the Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich. Prior to the Ph.D. study, he was a Research Assistant with The Chinese University of Hong Kong, Hong Kong, from 2019 to 2020. Following that, he was a Visiting Student Researcher with the Southern University of Science and Technology.

In 2021, he was a Robotics Student Fellow with ETH Zürich. In 2022, he was with The Chinese University of Hong Kong (Shenzhen Research Institute). His current research interests include motion planning and human–robot interaction.



**Weinan Chen** was born in 1991. He received the B.Sc. and Ph.D. degrees from the Guangdong University of Technology. During the Ph.D. degree, he spent a year with the University of Alberta, Edmonton, Canada, as a Visiting Scholar. After a year of Post-Doctoral Training with The Chinese University of Hong Kong, he joined the Southern University of Science and Technology, as the Principal Outstanding Post-Doctoral Fellow. He is currently a Lecturer with the Guangdong University of Technology. His research interests include SLAM and mobile robots.



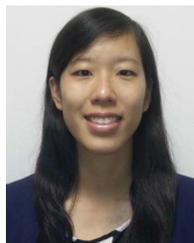
**Daniel Dugas** received the B.E. degree from EPFL in 2014 and the M.E. and Ph.D. degrees from ETHZ in 2017 and 2022, respectively. His research interests include the application of deep learning to robotics, embodied artificial intelligence, and human–robot interaction.



**Roland Siegwart** (Fellow, IEEE) is a Professor of autonomous mobile robots with ETH Zürich; the Founding Co-Director of the Technology Transfer Center, Wyss Zurich; and a board member of multiple high-tech companies. From 1996 to 2006, he was a Professor with EPFL Lausanne. He has held a visiting positions with Stanford University and NASA Ames. He was the Vice President of ETH Zürich, from 2010 to 2014. His research interests include the design, control, and navigation of flying and wheeled and walking robots operating in complex and highly

dynamical environments.

He received the IEEE RAS Pioneer Award and the IEEE RAS Inaba Technical Award. He is among the most cited scientist in robots world-wide, a co-founder of more than half a dozen spin-off companies, and a strong promoter of innovation and entrepreneurship in Switzerland.



**Jen Jen Chung** (Member, IEEE) received the B.E. and first Ph.D. degrees from The University of Sydney, in 2010 and 2014, respectively, and the second Ph.D. degree in information-based exploration-exploitation strategies for autonomous soaring platforms from the Australian Centre for Field Robotics, The University of Sydney. She is an Associate Professor of mechatronics with the School of Electrical Engineering and Computer Science, The University of Queensland (UQ). Prior to working with UQ, she was a Senior Researcher with the

Autonomous Systems Laboratory (ASL), ETH Zürich, from 2018 to 2022; and a Post-Doctoral Scholar with Oregon State University, researching on multiagent learning methods, from 2014 to 2017. Her current research interests include perception, planning, and learning for robotic mobile manipulation, algorithms for robot navigation through human crowds, informative path planning, and adaptive sampling.