A Unified Multiple-Motion-Mode Framework for Socially Compliant Navigation in Dense Crowds

Yujing Chen, *Member, IEEE*, and Yunjiang Lou, *Senior Member, IEEE*

***Abstract*— Mobile robots are expected to move safely and efficiently in *socially-compliant* ways in dense crowds. In this paper, a unified multiple-motion-mode framework is proposed to achieve socially-compliant navigation in dense crowds. The proposed framework consists of three successive phases, the identification of pedestrian groups and prediction of their trajectories, the generation of candidate local trajectories of the robot, and the determination of an optimal local trajectory. The pedestrians are grouped according to distances among pedestrians and their velocities and a predictor predicts the groups' trajectories. Three robot motion modes, moving-solo, pedestrian-following, and courteous-stopping, are proposed based on a systematic analysis and classification of dense crowds and they are utilized to generate candidate local trajectories. A composite metric is proposed to determine the best local trajectory by simultaneously considering the robot motion safety, efficiency, and stability of the trajectory. The proposed robot navigation framework is evaluated both in simulations and real-world experiments with various scenarios. The results show that the proposed framework surpasses the state-of-the-art methods in terms of efficiency and social compliance.**

***Note to Practitioners*—This paper is motivated by the socially-compliant navigation problem of mobile robots in densely crowded environments such as hospitals and airports. Robots navigation in these scenes need to follow collective social conventions to ensure ordered and efficient public pedestrian traffics, and individual social conventions for acceptance by human society. Existing approaches generally consider only one type of social conventions cannot accomplish a truly socially-compliant navigation and cannot be naturally accepted in the human pedestrian environment. This paper proposes to imitate the pedestrian's walking behavior for robot navigation by naturally choosing one of the three walking modes, moving-solo, following, and courteous-stopping. We construct the candidate trajectory sampling strategies for different motion modes based on the global path and the two types of social conventions. Experimental results show that the proposed framework can be applied in large-scale, densely crowded, unstructured, and human-robot coexisting environments for socially-compliant navigation. However, the proposed navigation framework has limited improvement in navigation efficiency in a highly crowded environment due to the limited feasible space. Even pedestrians are difficult to walk in these highly crowded environments. In future work, we will study how to auto-adjust the weights to steer the robot to efficient navigation in highly crowded environments.**

***Index Terms*— Mobile robot, socially compliant navigation, dense crowds, motion planning.**

I. Introduction

MOBILE robots are increasingly coexisting with dense crowds in daily-life scenarios, such as shopping malls [1], airports [2], and hospitals [3]. They are expected to navigate in those dense crowds with a safe and socially-compliant motion behavior [4]–[8]. *Safety* is a fundamental requirement and it implies that the robot is able to avoid possible collisions with both stationary obstacles, such as walls, tables and sofas, and moving obstacles, usually pedestrians. *Social compliance* is a more advanced requirement for a mobile robot to be integrated into human social lives. It requires the robot to respect and to follow pedestrians' social conventions by providing a human-like motion behavior [9]–[11]. Nearby pedestrians can thus quickly understand the robot's motion intention and make smooth walking interactions [12]–[14].

Pedestrians' social conventions are those collective and individual rules of pedestrian behaviors created and accepted by human society, which ensure ordered and efficient public pedestrian traffics, and personal space for comfortable, especially in dense crowd scenarios [15]. For example, pedestrians are usually required to walk on one side of the road, and to keep a courteous distance when passing other pedestrian(s) [16] and [17]. According to the conventions, pedestrians are able to understand motion intentions of others so as to plan their collision-free and efficient paths [18]–[20]. A social navigation approach that considers only one type of social conventions cannot accomplish a truly socially-compliant navigation and cannot be naturally accepted in the human pedestrian environment. Therefore, the core problem of the socially-compliant navigation is *how to integrate the two different social conventions simultaneously into the robot's planning formulation to generate a socially-compliant and also safe and efficient motion.*

Existing researches on socially-compliant navigation of robot have tried to steer the robot to keep a courteous distance from pedestrians by considering only the private space of pedestrians based on proxemics model [12], [13],

Manuscript received August 7, 2021; accepted November 1, 2021. This article was recommended for publication by Associate Editor S. Rathinam and Editor C. Seatzu upon evaluation of the reviewers' comments. This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB1313900 and in part by the Shenzhen Science and Technology Program under Grant JCYJ20180508152226630 and Grant JSGG20191129114035610. *(Corresponding author: Yunjiang Lou.)*

The authors are with the School of Mechatronics Engineering and Automation, Harbin Institute of Technology Shenzhen, HIT Campus, University Town of Shenzhen, Shenzhen, Nanshan 518055, China (e-mail: louyj@hit.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TASE.2021.3125367.

Digital Object Identifier 10.1109/TASE.2021.3125367

[21]–[24], where each pedestrian is regarded as an *independent* individual having no relationship with others. However, pedestrians' social conventions are much more complicated practically than just keeping courteous distance, especially in dense crowds. Researches on pedestrians' social conventions show that a pedestrian's walking behavior in dense crowds is related to not only the environment structure, but also his/her social relationships with nearby pedestrians [25] and [26]. According to [26], three typical modes of crowd behavior emerge naturally, corresponding to three scenarios, to follow the pedestrians' collective and individual social conventions simultaneously.

1) **Quasi-static crowds** appear in scenarios where pedestrians gather together to form various groups, such as the groups of pedestrians at the lobby of an airport/a hotel, at a hall, or an open space for conference reception/conference coffee break. A pedestrian walking in this scenario needs to keep a courteous distance from not only individuals but also groups to avoid invading both individuals' and groups' social spaces.
2) **Bi-direction flows** are formed by following the pedestrians' conventions of walking on one side of the road in narrow and long passage scenarios, where pedestrians would quickly pass through, such as in a hallway of a building or in a narrow pedestrian street. A pedestrian walking in this scenario should follow the pedestrian whose in front of him/her in the same moving direction and integrate into the corresponding pedestrian flow.
3) **Chaotically crossing flows** appear in a scenario where pedestrians compete with some others for right-of-way to maximize self-efficiency to reach his/her destination, such as in a crowded intersection. A pedestrian walking in this scenario should avoid other pedestrians, courteously stop and wait to make ways for seniors or people with disabilities, or benefit from following a leader or a group with a similar walking direction.

In order to develop a navigation algorithm to capture the social conventions of variously dense scenarios, Okal and Arras [27] use the inverse reinforcement learning approach that enables a mobile robot to acquire corresponding navigation mode from demonstrations of pedestrian walking behavior in three simulated scenarios. However, the robot can only use one navigation mode in a fixed scenario. If the scenario is changed, the navigation mode of the robot needs to be reset manually. Therefore, the challenge of socially-compliant navigation is how to deal with the navigation constraints imposed by complex social conventions like a human and to formulate them in a unified way, especially when pedestrians pay different degrees of attention to collective and individual social conventions in various scenarios. Moreover, it is also a challenge to develop a comprehensive metric to evaluate possible trajectories by considering both advanced and fundamental requirements, social compliance, safety and efficiency.

This paper explores novel techniques about how to generate a socially-compliant motion of mobile robot by considering pedestrians' social conventions. An intuitive look at pedestrians' walking behaviors led to the social psychological discovery [15] and [18] that pedestrians adapt to various crowded scenarios by naturally choosing one of the three walking modes, moving-solo, following, and courteous-stopping. Therefore, this paper formulates the socially-compliant navigation as a trajectory optimization problem of multiple motion modes. The contributions are summarized as follows.

1) A unified multiple-motion-mode framework of navigation is proposed by simultaneously integrating collective and individual social conventions.
2) Three motion modes of the robot, i.e., moving-solo, pedestrian-following, and courteous-stopping are proposed to imitate the different walking behaviors of the pedestrian in the typically dense crowds. In each mode, we propose a sampling strategy to find local terminal states of the robot, which are utilized to generate a set of candidate local trajectories.
3) A composite metric based on three novel metrics, which reflect the advancement of the robot towards the final destination, the possibility of an oscillation motion, and the possibility of an intrusion into the shared spaces of groups is established to determine the best candidate local trajectories.
4) The proposed robot navigation framework is evaluated in simulations and real-world experiments with various scenarios. The results show that the proposed navigation framework surpasses the state-of-the-art methods in terms of efficiency and social compliance.

In the remaining sections, Section II describes the navigation system and the necessary inputs of the proposed multiple-motion-mode framework. In section III, the candidate trajectory sampling strategies of three motion modes are described in detail, respectively. Section IV describes the comprehensive function and the optimal trajectory generation. The simulations and real-world experimental results are reported and analyzed in Section V. Finally, a conclusion is drawn in Section VI.

II. The Unified Multiple-Motion-Mode Framework

A. Navigation System Description

The proposed M^3 framework consists of three successive phases which execute in terms to determine the control input $\boldsymbol{u} = (v, \omega)^T$ of the robot, as shown in Fig. 1. It is supposed that the map of the environment and the global path from the robot's initial position to destination are known. The pedestrians' states, positions and velocities, can be tracked by an RGB-D camera at each time step. M^3 framework accepts the known map, the global path and the states of the tracked pedestrians at current time step as the input variables. The main purpose of each phase in M^3 framework is as follows, and the details will be discussed in the following subsections.

1) In the first phase, the pedestrians are grouped according to distances among pedestrians and their velocities, and a predictor predicts the groups' trajectories.
2) In the second phase, we propose a sampling strategy to generate the candidate local trajectories of the robot corresponding to the three motion modes based on the

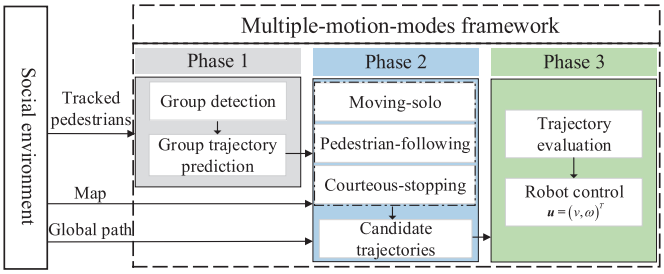


Fig. 1. Description of the proposed multiple-motion-modes (M^3) framework.

global path guidance and the predicted groups' trajectories obtained in the first phase.

3) In the third phase, a composite metric is used to evaluate the candidate trajectories obtained in the second phase.

At the end of the last phase, the best local trajectory is determined and the control input $\boldsymbol{u}$ of the robot can be obtained from the best local trajectory. At the next time step, M^3 framework gets the new input variables and re-computes the control input $\boldsymbol{u}$ of the robot.

B. Pedestrian Motion Prediction Under Group Constraints

In order to avoid collision with moving pedestrians nearby, the robot requires to predict their trajectories. In dense crowds, it is more appropriate to predict the trajectory of surrounding pedestrians by groups than by individuals. The reason is that 70% pedestrians move in groups of two or more individuals in dense crowds [25]. The benefits of group based trajectory prediction are as follows. First, the group based trajectory prediction can handle different situations of surrounding pedestrians because an individual can also be considered as a group. Second, it might happen that some pedestrians form a group with a shared space among them to have a private conversation or maintain a closed relationship. Robots need to identify such group shared spaces and avoid to invade them. An outsider needs to gain the permission of the insiders to become members of the group. Only when the newcomer can be allowed to join the group so that he/she can be positioned in the shared space of the group. Therefore, robots should not invade the group space. Third, to represent the crowd by groups of individuals can help to reduce the computational complexity of the pedestrian's trajectory prediction. In the proposed framework, we use the group tracker presented in [28] to detect groups in dense crowds. The working principle of the group tracker is to construct a social relationship graph to score the relationship of the tracked pedestrians by considering distances among pedestrians and their velocities. Pedestrians are labeled as a group if the score of their relationships are greater than a given threshold.

After the pedestrians are labeled as groups, we will discuss their characteristics. For convenience, we represent the state of a group by a quin tuple, i.e., $\boldsymbol{g}_m = (\boldsymbol{p}_m, \boldsymbol{v}_m, p_{rm})^T$, where $\boldsymbol{p}_m = (p_{mx}, p_{my})^T$ and $\boldsymbol{v}_m = (v_x, v_y)^T$ are position and velocity in x- and y- axis, respectively, and p_{rm} is the radius of the group. The position $\boldsymbol{p}_m$ are the average that are computed from each pedestrian's position in $\boldsymbol{g}_m$, $\boldsymbol{v}_m$ are the average of all pedestrians' velocities, and p_{rm} is the Euclidean distance from the center of the group to the furthest pedestrian position in $\boldsymbol{g}_m$. We denote the states of the obtained M groups in a crowd at time step k using a sequence $\mathcal{G}^k = \{\boldsymbol{g}_1^k, \boldsymbol{g}_2^k, \ldots, \boldsymbol{g}_M^k\}$ where $\boldsymbol{g}_m^k = (\boldsymbol{p}_m^k, \boldsymbol{v}_m^k, p_{rm}^k)^T$ is the state of the m-th group at time step k.

In order to predict a group's trajectory, we utilize the constant velocity model (CVM). Despite of the existence of other more complex models, such as Social-LSTM [29], CVM is preferable in practice due to its three advantages. First, CVM assumes that the group moves forward with a constant velocity in each time step. It implies that the movements of pedestrians are supposed not to be affected by the existence of the robot. Thus the robot is prompted to actively avoid pedestrians. Second, CVM is parameterless, so that it is applicable in many different environments. Third, the computational complexity of CVM is low. Therefore, CVM is suitable for a mobile robot with an onboard computing unit. It is verified in [30] that the performance of the trajectory prediction based on CVM in dense crowds is competitive to other state-of-the-art algorithms.

At the beginning of time step k, the predicted walking distance during the time interval is can be obtained as follows,

$$\hat{\boldsymbol{g}}_m^{k+1} = \left(\boldsymbol{p}_m^k + \Delta_m^k, \boldsymbol{v}_m^k, p_{rm}^k\right)$$

where $\Delta_m = \boldsymbol{p}_m^k - \boldsymbol{p}_m^{k-1}$ is the walking distance during the last time interval. Therefore, the future trajectory of the group $\boldsymbol{g}_m$ can be calculated as

$$\mathcal{T}_m = \left(\boldsymbol{g}_m, \hat{\boldsymbol{g}}_m^1, \hat{\boldsymbol{g}}_m^2, \ldots, \hat{\boldsymbol{g}}_m^N\right) \tag{1}$$

where $\hat{\boldsymbol{g}}_m^n$ is the predicted state of the group $\boldsymbol{g}_m$ at predicted time step n.

C. Global Path Generation

With the predicted trajectories of the identified groups in the dense crowd, M^3 framework proceeds to the second phase, which targets at providing a series of candidate trajectories of the robot. As shown in Fig. 1, the inputs of the second phase include the known map and the global path, together with the predicted trajectories of each group of pedestrians obtained in the first phase.

Since the robot is supposed to fulfill a specific task from one place to another, we denote the initial state and the terminal state of the robot by $\boldsymbol{x}_I$ and $\boldsymbol{x}_F$, respectively, where $\boldsymbol{x} = (p_x, p_y, \theta)^T$ and θ is the orientation of the robot. With the information of the environment, i.e., the map of the environment, the shortest path between initial state $\boldsymbol{x}_I$ and terminal state $\boldsymbol{x}_F$ can be obtained using the A* method. We define such a shortest path as the global path and it is used in the M^3 framework to guide the robot toward the terminal state which takes the static obstacles of the environment into consideration.

Because the crowded scenario is constantly changing, the robot needs to adjust its motion behavior at each time step. It implies that during each time step in the M^3 framework, a local trajectory of the robot has to be specified in order to meet the social compliance requirement. Denote the initial state of the robot at time step k by $\boldsymbol{x}_I^k$. By finding a terminal

state $\boldsymbol{x}_F^k$, the local trajectory can be generated based on $\boldsymbol{x}_I^k$, and $\boldsymbol{x}_F^k$. The typical method to obtain $\boldsymbol{x}_F^k$ is to take the robot as the center of a circle and sample in a circular space with a given radius in two-dimensional planar space. However, this method might fail to respect the social compliance requirement and be relatively inefficient. Moreover, the orientation θ of the terminal state $\boldsymbol{x}_F^k$ cannot be obtained.

In order to imitate the pedestrians' walking behaviors together with consideration on the efficiency to reach the destination, the limited onboard computing ability, and the kinematic constraints of the robot, we propose a lattice sampling strategy to find the best local trajectory. More specifically the following characteristics will be explained as follows.

First, in order to satisfy the requirement of reaching destination as quickly as possible without collision, the global path is used to establish the lattice sampling strategy. The candidate trajectories generated by the proposed strategy are based on the global path so that these trajectories can guide the robot to efficiently reach the destination by moving along the global path as far as possible. Second, in order to satisfy the requirement of the efficient calculation, the lattice sampling strategy is used to scatter widely the candidate trajectories while imitating pedestrians' walking behavior. The candidate trajectories need to be separated because nearly identical trajectories will likely intersect with the same obstacles. Thus it would be a poor and inefficient choice if two trajectories are unnecessarily close to each other. Third, in order to generate trajectories satisfying the kinematic constraint of the robot, we use a closed-loop control law to generate trajectory between the initial state of the robot $\boldsymbol{x}_I^k$ and the sampling terminal state $\boldsymbol{x}_F^k$ at time step k.

In order to use the global path to establish lattice sampling strategy, we build the Frenet coordinate system based on the global path. In the Frenet coordinate system, the s-axis represents the distance along the reference path (also known as longitudinal displacement), the d-axis represents the side-to-side position on the reference path (also known as lateral displacement), and the origin locates on the global path of the robot at the initial state $\boldsymbol{x}_I$, as shown in Fig. 2. Therefore, the position of the robot in the Cartesian coordinate system $\mathcal{X}$ and Frenet coordinate system $\mathcal{S}$ can be transformed as follows.

$$h : \mathcal{S} \mapsto \mathcal{X} \tag{2}$$

$$h^{-1} : \mathcal{X} \mapsto \mathcal{S} \tag{3}$$

where the expression of $h(\cdot)$ and $h^{-1}(\cdot)$ can be found in [31].

With the coordinate transformation, the proposed lattice sampling strategy tries to find a set of reasonable terminal state of the robot that covers the most reachable state space and has a minimal overlap and then generate a candidate trajectory based on $\boldsymbol{x}_I^k$ and each terminal state $\boldsymbol{x}_F^k$. In order to fulfill the lattice sampling, we introduce a set of lattice sampling parameters $\mathcal{Q}^k = \{s_{\min}, s_{\max}, d_{\min}, d_{\max}\}$, where the sampling space is from $s_{\min}$ to $s_{\max}$ in s-axis and from $d_{\min}$ to $d_{\max}$ in d-axis in Frenet coordinate system. With the specified lattice sampling parameters, the candidate terminal state $\boldsymbol{s}_c = (s_c, d_c)^T$ in the Frenet coordinate system can be randomly

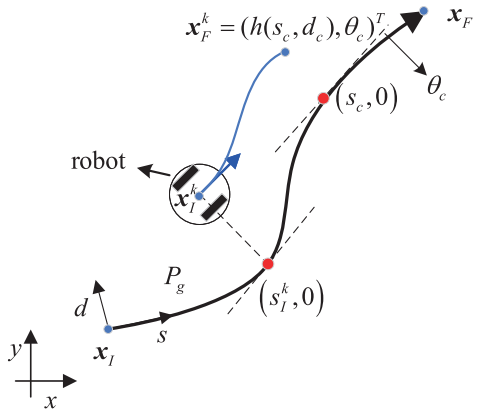


Fig. 2. The relationship between the Cartesian and Frenet coordinate systems. The solid black line is the global path, the blue line is the local trajectory, the blue dots are the position in Cartesian coordinate system, the red dots are the position in Frenet coordinate system.

generated by uniformly sampling as follows.

$$\begin{aligned} s_c &\in [s_I + s_{\min}, s_I + s_{\max}] \\ d_c &\in [d_I + d_{\min}, d_I + d_{\max}]. \end{aligned} \tag{4}$$

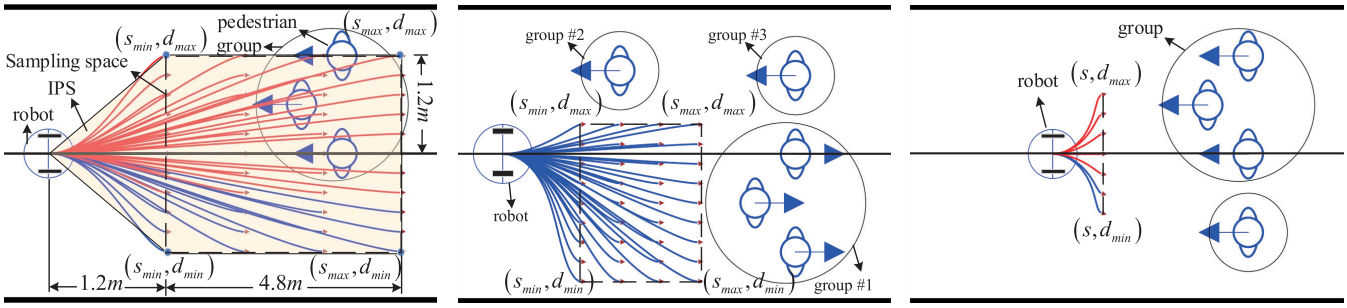
Therefore, the candidate terminal states in the Cartesian coordinate system can be obtained using (3),

$$\boldsymbol{x}_c = \begin{pmatrix} h(\boldsymbol{s}_c) \\ \theta_c \end{pmatrix} \tag{5}$$

where θ_c is the angle of the tangential direction of the point $h(s_c, 0)$ on the global path, as shown in Fig. 2. Note that the obtained terminal state of the robot may deviate from the global path in order to avoid intruding into groups. Based on (5), the robot selects any of the candidate local trajectory moving parallelly to the global path. It takes advantages over the typical option of moving forward to the global path and then moves along the global path again because the robot can maintain the most efficient longitudinal movement without additional lateral movement. The frequent back and forth movement of lateral direction should cause zigzagged motion of the robot in dense crowds.

From the above discussion on generating a local trajectory of the robot based on the global path, it can be found that the key is to determine the parameters $\mathcal{Q}^k$. In the proposed M^3 framework, we formulate the navigation problem as a lattice sampling based motion planning problem and determine the sampling parameters based on three typical motion modes of the pedestrians. In the remainder of the subsection, we will discuss the details of the motion modes, the determination of the sampling parameters, and the calculation of the obtained candidate local trajectories corresponding to each set of parameters.

As described in Section I, the complexity of pedestrians' social conventions comes from the requirement of the ordered and efficient public pedestrian traffics with interactions among a large number of groups. A mobile robot needs to imitate an individual's walking behavior to integrate into the dense crowds without negatively affecting the efficiency of public pedestrian traffics. More specifically these characteristics will be explained as follows.



(a) Moving-solo mode (b) Pedestrian-following mode (c) Courteous-stopping mode

Fig. 3. Terminal states sampling for candidate trajectory sets generation. The direction and the sampling range of the state space can be guided by global path guidance. The area in the dashed rectangle is the sampling space. The red trajectories are discarded with collision. The blue trajectories are the collision-free trajectories. **Left:** Candidate trajectories of moving-solo. The yellow area is the information process space of the robot. **Center:** Candidate trajectories of pedestrian-following. The robot follows group #1 in this situation. **Right:** Candidate trajectories of courteous-stopping.

III. Candidate Trajectory Sampling Strategies

In this section, we propose to imitate the pedestrian's walking behavior to choose a suitable motion mode naturally for robot navigation and thus formulate socially-compliant navigation as a trajectory optimization problem. We construct the candidate trajectory sampling strategies for different motion modes based on social conventions.

In order to autonomously choose a suitable motion mode and find the optimal motion, the sampling strategies of three motion modes pay different degrees of attention to the collective and individual social conventions. Then, the robot can evaluate the candidate trajectories of any motion modes by a comprehensive objective function, and autonomously choose an optimal trajectory for adapting to ever-changing scenarios.

A. Moving-Solo Mode

In the moving-solo mode, a robot imitates a pedestrian's walking behavior that tries to reach the destination while keeping courteous distances from pedestrians and respecting the groups' shared space. According to [20] pedestrians are observed to be more interested in the exact front having a shortly relative lateral distance from the walking direction. Therefore, it is more reasonable to utilize the information process space (IPS) to depict the interested areas of the pedestrians. The concept of IPS is proposed in [20] and further developed in [17]. It is the space in which pedestrians consider obstacles and other pedestrians in order to compute next motions and where psychological comfort is evaluated.

In this paper, we propose to utilize the IPS to determine the parameter $\mathcal{Q}_a^k$ of the moving-solo mode. As shown in Fig. 3(a), the pentagon in yellow is the IPS and then we can sample the candidate local terminal state in the dashed rectangle of the IPS. The application of IPS improves the efficiency in computation when sampling the candidate local terminal state since it is much smaller than the typically circular spaces. In addition, the robot is able to find the optimal strategy of the moving-solo mode in IPS due to the kinematic constraints of the robot and the requirement of the efficiency for the navigation task. With the principles of IPS, the sampling parameters which determine the generation of candidate local terminal states of the robot can be set as $\mathcal{Q}_a^k = \{1.2, 4.8, -1.2, 1.2\}$.

Following the sampling strategy mentioned previously, we can then determine the state $\boldsymbol{s}^k$ of the robot in the Frenet coordinate system and correspondingly the state $\boldsymbol{x}^k$ in the Cartesian coordinate system at time step k. In order to make a well balance between the computational efficiency and the sufficiency in sampling in IPS, we sample 68 times in rectangular IPS. Each sample lies at the intersections of the lattice graph whose vertical lines are distributed uniformly between $s_{\min}$ and $s_{\max}$, and whose horizontal lines are distributed between $d_{\min}$ and $d_{\max}$. Therefore, a set of candidate local terminal states is obtained and is denoted by $\mathcal{S}_a^k$.

B. Pedestrian-Following Mode

In the pedestrian-following mode, a robot imitates a pedestrian's walking behavior that follows a leader to benefit from the collective motion. In this mode, the key role is the leader, which can be either individual or a group. In the general situation where there are multiple groups in the dense crowd, it is important to evaluate the goodness of groups, with which a best leader can be selected. In this paper, the goodness of a group is evaluated from the following aspects.

The first feature is slowness. The groups moving too slowly are inappropriate to be selected as the leader. We neglect the groups that are moving slower than 0.3 m/s and set the remaining groups as candidate leaders.

The second feature is similarity of the moving direction. We need to evaluate the similarity of the moving directions of the robot and all the groups and then to determine the leader. At time step k, the moving direction θ_{gm} of the m-th group can be easily obtained with its velocity $\boldsymbol{v}_m^k$, which is included in its current state $\boldsymbol{g}_m^k$. As to the moving direction of the robot, one of the common methods is to use the direction from the robot's current position pointing to its final destination [32]. However, this method does not consider the characteristics of the environment, such as corners, and will fail to select the suitable moving direction of the robot in these situations. For example, in a situation where the destination is behind a corner, a reasonable moving direction of the robot would be to first reach the corner before turning toward the destination, rather than directly towards the destination.

In order to overcome the shortcoming of the existing method of determining the robot's moving direction, we propose a

method based on the global path. As mentioned above, the global path is the shortest path from the initial state of the robot $\boldsymbol{x}_I$ to the destination $\boldsymbol{x}_F$ considering on the static obstacles in the environment. Therefore, the suitable moving direction of the robot at a state $\boldsymbol{x}_c$ would be the tangent direction θ_c of the global path in the Cartesian coordinate system, as shown in Fig. 2. Therefore, the similarity of the moving direction is defined as $\Delta\theta_m = \theta_{mc} - \theta_{gm}$, where θ_{gc} is the angle of the tangential direction of the point $(s_{mc}, 0)$ on the global path and s_{mc} is the s -axis position of the $\boldsymbol{g}_m^k$ at Frenet coordinate system. We neglect the groups with corresponding $\Delta\theta_m$ are bigger than 30° and set the remaining groups as candidate leaders.

The third feature is Minimum switching. It may happen that multiple groups are candidate leaders satisfying the above mentioned criteria. Since it may cause an oscillation motion when frequently switching between different leaders, it is more reasonable to remain the leadership of a group if it is among the many candidates. If the leader in the previous time step is not a candidate leader currently, we choose the one from the set of the candidate leaders that has the smallest $\Delta\theta_m$.

Following the criteria base on slowness, similarity in moving direction, and minimum switching, the leader which the robot intends to follow in the current time step can be selected. Then it is necessary to design a tracking strategy for the robot to maintain a temporal safety distance to the leader. We determine the sampling parameters as follows, $\mathcal{Q}_o^k = \{s_o^k - 4.8, s_o^k - 1.2, d_o^k - p_{ro}^k, d_g^k + p_{ro}^k\}$, where $\boldsymbol{s}_o^k = (s_o^k, d_o^k)^T = h(\boldsymbol{p}_o^k)$ is the position of the selected leader in Frenet coordinate system, and p_{ro}^k is the radius of the shared space of the selected leader. Similarly, we randomly generate a set of candidate local terminal states, denoted by $\mathcal{S}_o^k$. In Fig. 3(b), we demonstrate the generation of local candidate terminal state in the pedestrian-following mode, where the position and the direction of the red arrows are the sampling local terminal states based on the leader group #1.

C. Courteous-Stopping Mode

In the courteous-stopping mode, a robot imitates the pedestrian's walking behavior that is supposed to slow down or even stop to make ways for groups, seniors, or people with disabilities. Therefore, the sampling space should be smaller than those in other motion modes together with consideration on the kinematic constraints of the robot. We set the sampling parameters at the time step k as $\mathcal{Q}_e^k = \{1.2, 1.2, -0.5, 0.5\}$, as shown in Fig. 3(c). We randomly generated a set of 7 candidate terminal state denoted by $\mathcal{S}_e^k$.

D. Trajectory Generation Satisfying Robot Constraints

At the end of the second phase, three sets of candidate local terminal states, denoted by $\mathcal{S}_a^k$, $\mathcal{S}_o^k$, and $\mathcal{S}_e^k$, are generated according to the three typical motion modes. Let $\mathcal{S}^k = \mathcal{S}_a^k \cup \mathcal{S}_o^k \cup \mathcal{S}_e^k$ and we need to generate a local trajectory for each candidate local terminal state in $\mathcal{S}^k$. The typical ways to generate trajectory between two points are based on Bezier splines or curvature polynomials of an arbitrary order [31], and then a controller is designed to follow the generated trajectory. However, since the trajectory generation and controller design are independent processes, it may fail to simultaneously achieve the optimal local trajectory and the optimal control of the robot. Moreover, the controller needs to check the collision again, resulting in additional computational costs. Besides, the θ of the terminal state $\boldsymbol{x}_F^k$ is not considered.

In order to integrate the local trajectory generation with the controller design, we utilize a real-time steering function to generate a local trajectory between the initial state of the robot $\boldsymbol{x}_I^k$ and sampling terminal state $\boldsymbol{x}_F^k$ at time step k as follows. Without loss of generality, we assume that the robot is a differential-drive robot. The open loop kinematic model of the robot can be expressed as

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{6}$$

where $\boldsymbol{u}(t) = (v(t), \omega(t))^T \in \mathbb{R}^2$ is the control input at time step t, $v(t)$ is the linear velocity, $\omega(t)$ is the angular velocity, $\boldsymbol{x}(t)$ is the robot state at time step t, and $f(\cdot,\cdot)$ is the kinematics of the robot. Following the transformation from world Cartesian coordinate system to the polar coordinate system, the kinematic model (6) can be transformed into the following model

$$\begin{aligned} \dot{\rho} &= -v\cos\alpha \\ \dot{\alpha} &= \frac{\sin\alpha}{\rho}v - \omega \\ \dot{\varphi} &= -\omega \end{aligned} \tag{7}$$

where ρ is the Euclidean distance between a pair of state $\boldsymbol{x}_I^k$ and $\boldsymbol{x}_F^k$, β is the angle between $\boldsymbol{x}_I^k$ and $\boldsymbol{x}_F^k$ and $\varphi = (\pi/2) - \beta$, α is the angle between the orientation of the robot to the vector from $\boldsymbol{x}_I^k$ to $\boldsymbol{x}_F^k$, v_l and v_r are the wheel velocity in left and right, respectively, $v_{\max}$ is the maximum velocity of the wheel and b the robot's diameter. The relationship in (7) as shown in Fig. 4.

The following control law which is proposed in [33] and has been proved to assure asymptotically heading convergence and system's local stability, and it guarantees the smoothness of the generated trajectories without cusps. The control law can be expressed as

$$\begin{aligned} v &= K_\rho \rho \\ \omega &= K_\alpha \alpha + K_\phi \phi \\ K_\rho &> 0 \\ K_\phi &< 0 \\ K_\alpha + K_\phi - K_\rho &> 0 \end{aligned} \tag{8}$$

where K_ρ, K_α, and K_ϕ are the controller gains. Substituting the control law (8) in to (7), we can obtain the closed-loop model as

$$\begin{aligned} \dot{\rho} &= -K_\rho \cos\alpha \\ \dot{\alpha} &= K_\rho \sin\alpha - K_\alpha \alpha - K_\phi \phi \\ \dot{\phi} &= -K_\alpha \alpha - K_\phi \phi. \end{aligned} \tag{9}$$

Therefore, given an initial state $\boldsymbol{x}_I$, an integration time Δt and a terminal state $\boldsymbol{x}_F$, the state of the robot $\hat{\boldsymbol{x}}^{t+\Delta t}$ is

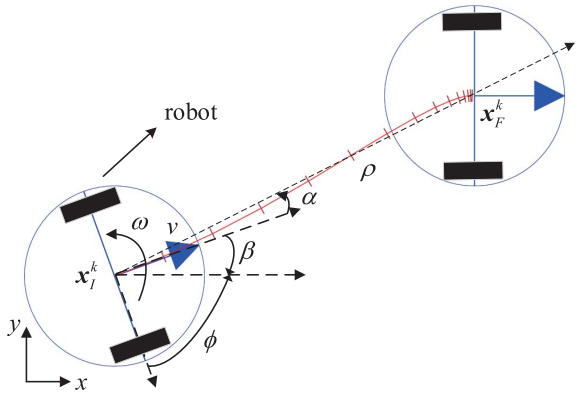


Fig. 4. The state relationship based on $\boldsymbol{x}_I^k$ and $\boldsymbol{x}_F^k$. The red line is the generated trajectory when steering the robot from $\boldsymbol{x}_I^k$ to $\boldsymbol{x}_F^k$.

generated by integrating (8) and (9)

$$\hat{\boldsymbol{x}}(t+\Delta t) = \sum_{t}^{t+\Delta t} f(x(t), u(t))\, dt \; + \boldsymbol{x}(t). \tag{10}$$

In Fig. 4, the red line is the generated trajectory $\mathcal{T}_r = (\boldsymbol{x}_I, \hat{\boldsymbol{x}}^1, \hat{\boldsymbol{x}}^2, \ldots, \hat{\boldsymbol{x}}^N, \ldots, \boldsymbol{x}_F)$ between the initial state $\boldsymbol{x}_I$ and terminal state $\boldsymbol{x}_F$, where $\hat{\boldsymbol{x}}^n = \boldsymbol{x}(t + n\Delta t)$ is the predicted state of the robot. The control input $\hat{\boldsymbol{u}}^n$ generated by (9) is based on the robot's kinematic model, and it can be applied directly to steer the robot. Moreover, in M^3 framework, the trajectory generation and control design are tightly integrated and allowed to directly optimize the robot's trajectory to the preference of its user. The overall M^3 framework is much more flexible and can handle the variability of the real environments. In this paper, we set $K_\rho = 1.0$, $K_\alpha = 2.5$, and $K_\phi = -0.5$ based on our empirical experiences.

IV. Trajectory Optimization

At the third phase, a set of candidate local trajectories are generated by imitating the pedestrians' walking behaviors. In order to evaluate the goodness of each local trajectory, we propose three metrics that reflect the performance of the trajectories from different aspects.

The first metric is **progress**. Since the main task of the robot is to move to a pre-specified destination $\boldsymbol{x}_F$, it is more preferable for the robot to move as far as possible in each time step. In this paper we propose the metric, progress, to measure the distance that the robot moves along each candidate trajectory. The progress of a given local trajectory in the k-th time can be expressed as follows.

$$C_1 = s_F^k - s_I^k. \tag{11}$$

It can be noticed that the progress is evaluated in the Frenet coordinate system. Since the global path considers static obstacles of the environment, the metric of progress design based on the global path can also consider the static obstacles.

The second metric is **similarity**. One of the fundamental requirements for mobile robots is the stability, i.e., to avoid an oscillation motion. In order to evaluate the possibility of oscillation and the stability of the robot, we evaluate

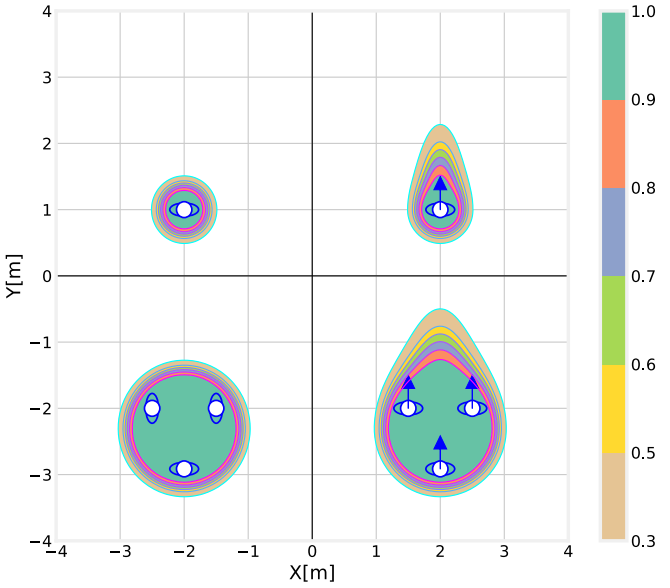


Fig. 5. Group space model. The different color area represent different blame value. **Top left**: a static individual. **Top right**: a moving individual. **Bottom left**: a static group. **Bottom right**: a moving group.

the similarity of each candidate local trajectory to the last trajectory, which can be obtained as follows.

$$C_2 = \frac{\Delta \boldsymbol{x}^{k\,T} \cdot \Delta \boldsymbol{x}^{k-1}}{\| \Delta \boldsymbol{x}^{k-1} \|} \tag{12}$$

where $\Delta \boldsymbol{x}^k = \boldsymbol{x}_F^k - \boldsymbol{x}_I^k$.

The third metric is **blame**. In order to evaluate the potential disturbance caused to the nearby pedestrians by the robot, the proxemics model which quantifies pedestrians' comfort and safety based on distances from the robot to the closest pedestrians. The proxemics model is expressed as a two-dimensional asymmetric Gaussian function [13],

$$h_p(x, y) = \exp\left(-\frac{\bar{x}^2}{2\sigma_x^2} - \frac{\bar{y}^2}{2\sigma_y^2} \right) \tag{13}$$

where $\bar{x}$, $\bar{y}$ are the distance from the robot to a pedestrian and σ_x, σ_y are the variables related to his/her velocity in x- and y- axis, respectively [13]. It can be seen that such a space is based on the states of each pedestrian. However, as discussed previously, the group based method is more reasonable. Pedestrians are more likely to walk in groups in dense crowds, and the shared space should be respected by others. In this paper, we propose a group space model that can depict the collective motion of a specific group of pedestrians and also respect the shared space in groups. The group space model is expressed as follows.

$$h_g(x_g, y_g) = \exp\left(-\frac{\Delta \bar{x}_g^2}{2\sigma_x^2} - \frac{\Delta \bar{y}_g^2}{2\sigma_y^2} \right) \tag{14}$$

where $\Delta \bar{x}_g$, $\Delta \bar{y}_g$ are the distances from the robot to the group in x- and y- axis, respectively. In Fig. 5, we illustrate the contours of the standing still individual, a standing still group,

a moving individual, and a moving group of the group space model (14), respectively.

Therefore, the metric, blame, which measures the potential disturbance caused to the nearby pedestrians by the robot can be calculated as follows.

$$C_3 = \max\{h_g^n(\hat{\boldsymbol{x}}^n; \hat{\boldsymbol{g}}_m^n)\}. \tag{15}$$

Since it implies that the robot disturbs the group by intruding into its shared space when $h = 1$ (the green areas in Fig. 5), the corresponding candidate local trajectory will be discarded. Thus, the robot does not invade the shared space of any group based on (15).

Based on the proposed three metrics, we utilize the following composition to evaluate the candidate trajectories,

$$C = w_1 C_1 + w_2 C_2 + w_3 C_3 \tag{16}$$

where w_1, w_2, and w_3 are weights for the three metrics, respectively. In practice the weights are problematic as a result of the order of prioritization of each metric in the navigation task. The composite metric C in (16) allows M^3 framework to evaluate the robot motion and choose the optimal one,

$$\underset{\mathcal{S}^k}{\operatorname{argmin}} \sum_{n=0}^{N} C^n. \tag{17}$$

The best local trajectory of each time step is determined by (17) and the $\hat{\boldsymbol{u}}^1$ of this local trajectory is the control input of the robot. Through the continuous execution, the robot is finally to reach the destination.

The weight of the progress metric is set $w_1 = -1$ to guarantee the robot move to the goal destination. Compared with $w_2 = -1$, when the value of the similarity weight is $w_2 = -0.1$, the trajectory selection is more sensitive to the value of w_3. It is hard to choose the value of w_3 because the robot may cause oscillation motion due to the sensitivity of the blame metric. In contrast, when the value of similarity weight is $w_2 = -2$, the trajectory selection is more insensitive to the value of w_3. It is hard to choose the value of w_3 because the robot may cause the aversion of pedestrians due to the insensitivity of the blame metric. Therefore, the weight of similarity is set to $w_2 = -1$. Moreover, the robot should keep enough distance from pedestrians and thus the weight of blame is set to $w_3 = 4$.

It is worth noting that the robot is supposed to navigate in a socially-compliant and also safe and efficient way in dense crowds regardless of the scenarios. Therefore, the weights w_1, w_2 and w_3 are fixed during entire navigation. However, with the increase of crowd density, the feasible space for the robot decreases, and thus w_2 and w_3 need to be tuned more carefully. The density of the crowd can be divided into three levels [34], low density (<0.2 person/m^2), medium density (0.2 − 0.8 person/m^2), and high density (>0.8 person/m^2). This paper aims at low densely and medium densely crowded scenarios and the set of weights $w_1 = -1$, $w_2 = -1$, $w_3 = 4$ is applied to various scenarios in simulations and real-world experiments.

TABLE I

PARAMETERS OF THE ROBOT IN THE SIMULATIONS AND THE REAL-WORLD EXPERIMENTS

Parameter	Size	v_{max}	ω_{max}
Simulation	$0.508m \times 0.430m$	$1.2m/s$	$1.0rad/s$
Real-world	$0.699m \times 0.515m$	$0.7m/s$	$1.0rad/s$

V. EVALUATION

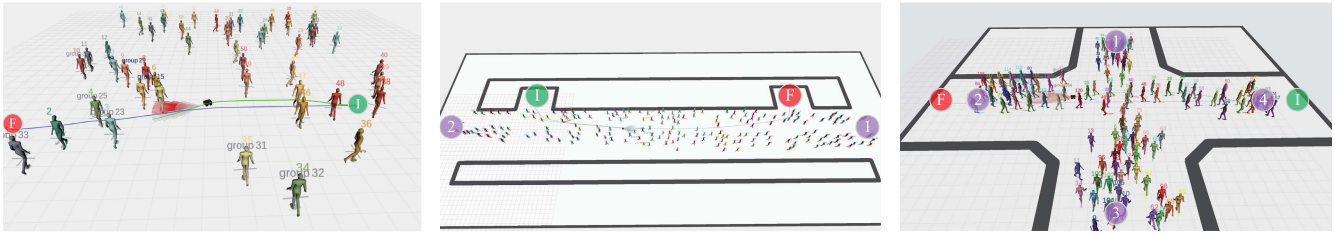
A. Setup

In this paper, we carry out both simulations and real-world experiments based on the Robot Operating System (ROS) with Ubuntu 16.04 on an Intel i5-8250U CPU with 8GB of RAM. All programs were coded in C++. For real-world experiments, a SUMMIT-XL STEEL differentially-driven mobile robot is utilized for the experimental studies, as shown in Fig. 13. The size of the robot is 0.699 m × 0.515 m, and the height of the robot is 0.8 m. A Kinect V1 camera is mounted on top, and a HOKUYO-10LX laser is mounted on the front of the robot. The parameters of the robot utilized in the simulation and the real-world experiments are listed in Table I, where the $v_{\max}$ and $\omega_{\max}$ are the linear and angular velocity of the robot, respectively. In real-world experiments, the maximum linear velocity of the robot is reduced for safety reasons.

The standard A^* global planner in ROS is used to eliminates experimental variation, but an advanced global planner can also be used in practice. In the first phase of M^3 framework, we utilize the package implementation from SPENCER project [3] to track the pedestrians and the groups. This pedestrian tracking module can run at 5 Hz on our computer. The second and third phases of M^3 framework can run at 50 Hz with 111 candidate trajectories being evaluated, where the number of trajectories is 68 in $\mathcal{S}_a^k$, 36 in $\mathcal{S}_o^k$ and 7 in $\mathcal{S}_e^k$. Therefore, in our simulations and experiments, we set the calculation time of each time step of the M^3 framework as 0.2 s by considering the update frequency of the pedestrians tracking module. The proposed M^3 framework can run at **50** Hz if there is a faster pedestrians tracking module. A pedestrian's walking speed is usually 1.2 m/s and the length of his/her interesting areas of the information process space (IPS) is 4.8 m. Therefore, the predicted time is set to 4 s and N is set to $N = 20$ when the update frequency of the M^3 is 0.2 s in the experiments. We set $\sigma_x = 0.3$ and $\sigma_y = 0.3 + v_m \times 0.4$ based on our empirical experiences, where v_m is the velocity of the group $\boldsymbol{g}_m$.

B. Simulations

M^3 framework is tested in three densely crowded scenarios in simulations. The three scenarios, as shown in Fig. 6, are the quasi-static crowds scenario (QSC) which the pedestrians are standing still, the bi-direction flows scenario (BDF) which the pedestrians are walking with opposing flows, and the chaotically crossing flows scenario (CCF) which the pedestrians walking with a randomly crossing interaction. The parameters of the scenarios in the simulations are listed in Table II, where the maximum velocity of the pedestrians in the simulator is 1.3 m/s. We utilize the simulator implementation from SPENCER project [3] to simulate the dense crowd. For



(a) Quasi-static crowded scenario (b) Bi-direction flows scenario (c) Chaotically crossing flows scenario

Fig. 6. The three scenarios are used in the simulations. Group social relations are visualized with text among pedestrians. The circularly purple labels are the destinations of the pedestrians. The circularly green label is the initial position of the robot and the circularly red label is the destination of the robot. The grey thick line is the wall. The blue line is the global path, the grey lines started from the robot are the candidate trajectories, while the red ones are the collision trajectories, and the green one is the best trajectory. The current robot's motion mode is shown above the best trajectory. **Left:** Quasi-static crowds scenario with 50 pedestrians and 18 groups. **Center:** Bi-direction flows scenario with 200 pedestrians moving in two distinct "flows." **Right:** Chaotically crossing flows scenario with 120 pedestrians crossing each other's paths.

TABLE II

PARAMETERS OF THE SCENARIOS IN THE SIMULATIONS

Scenario	Size	Number of pedestrians	Length of global path	Maximum velocity
QSC	$30m \times 30m$	50	$20m$	$1.3m/s$
BDF	$120m \times 10m$	200	$70m$	$1.3m/s$
CCF	$40m \times 40m$	120	$30m$	$1.3m/s$

each simulation scenario, we generate 50 random cases for evaluation. In each test case, the initial position and velocity of the pedestrians are randomly generated and the destinations of the pedestrians are fixed, as shown in Fig. 6.

For comparative purposes, two existing state-of-the-art approaches for social compliant navigation, the anticipative kinodynamic planning framework (AKP) [21] and the inverse reinforcement learning-based framework for socially normative navigation (LBN) [27] are implemented as the benchmark to demonstrate the performance of M^3 framework. AKP considers the individuals' social space and steers the robot to provide a minimum impact on nearby individuals for social navigation. In AKP, the robot is repulsed by the nearby pedestrians and attracted by the destination. The repulsive forces are calculated based on the distance between the robot and the nearby pedestrians. The parameters and the open-source code of the AKP are provided in [21]. LBN learns different robot navigation behaviors with the same set of features in different scenarios from expert demonstrations. The robot can only use one navigation behavior in a fixed scenario. If the scenario is changed, the navigation behavior of the robot needs to be re-set manually. According to [27], LBN learns the sociability motion behavior in quasi-static crowds scenario, the merge motion behavior in bi-direction flows scenario, and the slipstream motion behavior in chaotically crossing flows scenario, respectively. The parameters of all motion behaviors and the open-source code of the LBN are provided in [27].

In order to evaluate the performance of the efficiency of the mobile robot when undertaking the navigation task, the navigation time (NT) to reach the robot's destination is compared. The ways to evaluate the performance of social compliance metrics are different in the quasi-static crowds scenario and the other two scenarios.

In the quasi-static crowds scenario, three metric criteria for measuring social compliance are as follows.

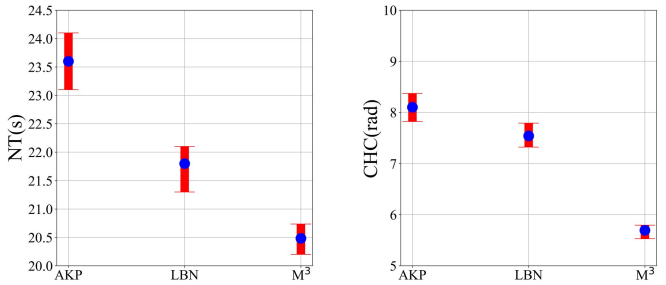
1) Cumulative heading changes (CHC), estimating the degree of path oscillation by adding up the heading changes of the robot during the whole navigation process.
2) Pedestrian distance (PD), recording the number of times the distances from the robot to a pedestrian are less than 1.2 m. Based on the proxemics, this distance is likely to attract the attention of pedestrians.
3) Relation of group disturbance (RGD), recording the number of times the robot invades the groups' share space.

It is inaccurate to use CHC, PD, and RGD to evaluate the social compliance of the robot in the bi-direction flows scenario and the chaotically crossing scenario due to the very large crowds and the randomly generated initial state of moving pedestrians of the simulator. We use the success rate of the navigation task (TSR) to evaluate the social compliance performance of three frameworks in these two scenarios. The reasons are that a mobile robot considering social compliance is better able to adapt to different density scenarios, and complete navigation tasks without impacting the ordered and efficient public pedestrian traffics. This metric criteria is defined as

1) Task success rate (TSR), the number of times the robot successfully reached the destination over multiple tests (task is aborted if the robot is "stuck" for more than 30 seconds).

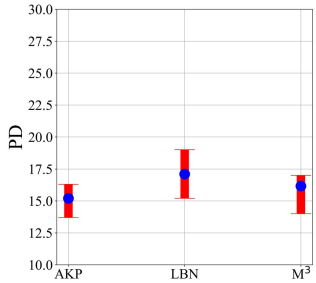
The performance comparisons in simulation studies are as follows.

1) Quasi-Static Crowds Scenario: The average performance of the NT, CHC, PD, and RGD of 10 times of quasi-static crowds scenario simulations are shown in Fig. 7. From Fig. 7(a), (b) and (c), we can find that the AKP has the highest mean of NT value, 23.7 s, the highest mean of CHC value, 8.1 rad, the lowest mean of PD value, 15.5, and the mean of RGD value, 0.3. AKP uses the social force to determine the robot velocity. It is easy to be trapped into a local extremum as the crowd density increases and thus may invade the shared space of the groups. The robot would even move backward in the situation when there are many pedestrians in front of it. This strategy makes large space between the robot and pedestrians, however, it leads to very

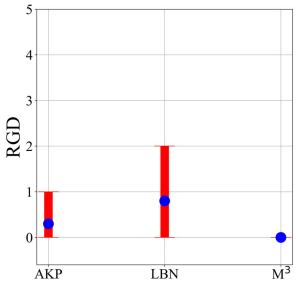


(a) Time to reach destination

(b) Cumulative heading changes

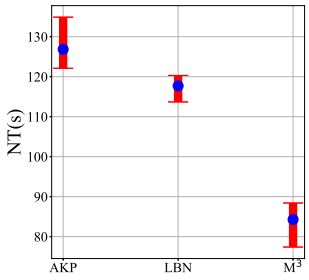


(c) Individual disturbance

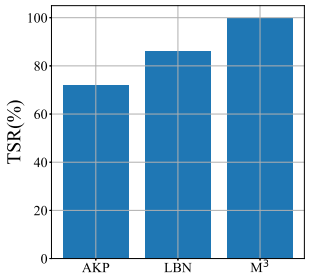


(d) Relation of group disturbance

Fig. 7. The performance of the NT, CHC, PD, and RGD of AKP, LBN and M^3 are evaluated in the quasi-static crowded scenario. All metrics are given as means and 95% intervals from running in 10 test cases.



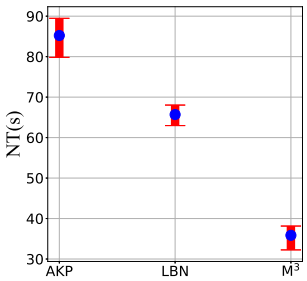
(a) Time to reach destination



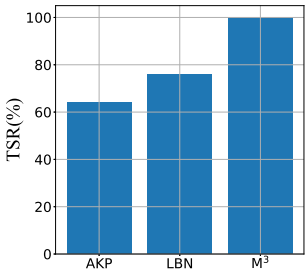
(b) Task success rate

Fig. 8. The performance of the NT and TSR of AKP, LBN and M^3 are evaluated in the bi-direction flows scenario. All metrics are given as means and 95% intervals from running in 10 test cases.

low efficiency. LBN has the mean of NT value, 21.8 s, the mean CHC value, 7.7 rad, the mean of PD value, 17.1, and the mean of RGD value, 0.7. LBN learns the navigation policy from the expert demonstration. However, the learned motion is suboptimal due to the limited demonstration data. Moreover, the expert demonstrations do not consider the shared space of the group. M^3 has the lowest mean of NT value, 20.5 s, which performs 15.6% better than AKP and 6.3% better than LBN. It has the lowest mean of CHC value, 5.7 rad, which performs 42.1% better than AKP and 35.0% better than LBN. In M^3 framework, the robot needs to trade off the progress, similarity and blame of the trajectory. Thus, with the set of weights of progress, similarity and blame in this paper, the NT is 15.6% and the CHC is 42% better than AKP, and the PD is only 10% not as good as AKP. Unlike AKP and LBN, M^3 does not steer the robot into the group space. The results of RGD show that the robot will avoid invading the group space when the group space is modeled in the planning.



(a) Time to reach destination



(b) Task success rate

Fig. 9. The performance of the NT and TSR of AKP, LBN and M^3 are evaluated in the chaotically crossing flows scenario. All metrics are given as means and 95% intervals from running in 10 test cases.

TABLE III
PARAMETERS OF THE SCENARIOS IN THE REAL-WORLD EXPERIMENTS

Scenario	Size	Number of pedestrians	Length of global path	Maximum velocity
QSC	$10m \times 3m$	6	$10m$	$1.3m/s$
BDF	$10m \times 3m$	6	$10m$	$1.3m/s$
CCF	$10m \times 9m$	6	$12m$	$1.3m/s$

2) Bi-Direction Flows Scenario: The average performance of the NT and TSR of 10 times of bi-direction flows scenario simulations are shown in Fig. 8. The results in Fig. 8(a) show the run time during the navigation and the results in Fig. 8(b) show the task success rate of three methods. We can find that the AKP has the highest mean of NT value, 126.7 s, the lowest task success rate, 72%. LBN has the mean of NT value, 118.4 s, the task success rate, 86%. M^3 has the lowest mean of NT value, 88.1 s, which performs 43.8% better than AKP and 34.3% better than LBN. It has the highest task success rate, 100%, which performs 28% better than AKP and 14% better than LBN. AKP is easy to get stuck as the crowd density increases. LBN learned the merge motion of the robot from the expert demonstrations and it still subjects to the limited demonstration data. As mention in Section I, the robot can benefit from utilizing the appropriate motion mode based on the social conventions of the scenario. In the bi-direction flows scenario, the robot utilizes M^3 can follow a flow that has the same moving direction to improve the performance of the robot.

3) Chaotically Crossing Flows Scenario: The average performance of the NT and TSR of 10 times of chaotically crossing flows scenario simulations are shown in Fig. 9. The results in Fig. 9(a) show the run time during the navigation and the results in Fig. 9(b) show the task success rate of three methods. We can find that the AKP has the highest mean of NT value, 86.1 s, the lowest task success rate, 64%. LBN has the mean of NT value, 66.1 s, the task success rate, 76%. M^3 has the lowest mean of NT value, 36.4 s, which performs 136.5% better than AKP and 81.6% better than LBN. It has the highest task success rate, 100%, which performs 36% better than AKP and 22% better than LBN. The robot utilized M^3 benefits from taking social conventions into account to enhance its navigation capabilities, similar to the performance in the bi-direction flows.

In order to better show the details of the M^3 framework in dense crowds, we take an example of a running process in

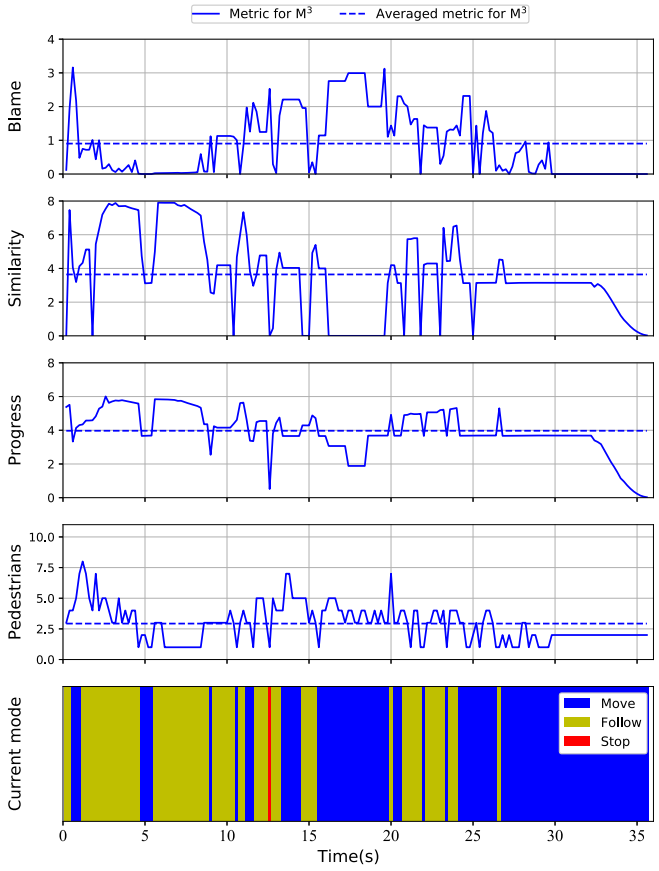


Fig. 10. The cost of blame, similarity, and progress of the robot with the proposed M^3 framework in the chaotically crossing flows scenario in simulation are given. The number of pedestrians within 2 m and the mode selected in each time step of M^3 framework are also given.

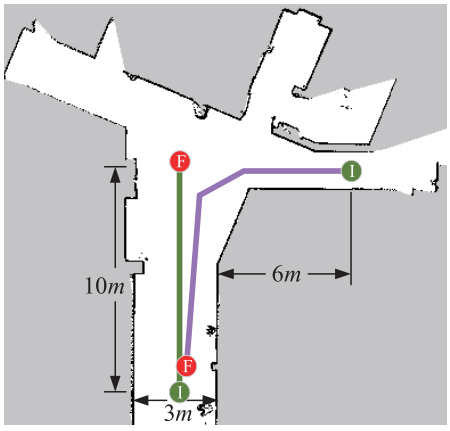


Fig. 11. The map of the real-world experiments. The green line is the global path for experiments of the quasi-static crowds scenario and the bi-direction flows scenario. The purple line is the global path for experiments of the chaotically crossing flows scenario. The robot starts from the circular label of green to the circular label of red, respectively.

chaotically crossing flows scenario, as shown in Fig. 10. This example is selected because the navigation time of the robot in this example is closest to the average value over 10 simulations. Moreover, the chaotically crossing flows scenario is the most complex and crowded scenario of the three simulation scenarios. In Fig. 10, the cost of blame, similarity, and progress of M^3 are given. The robot has lower blame cost, higher

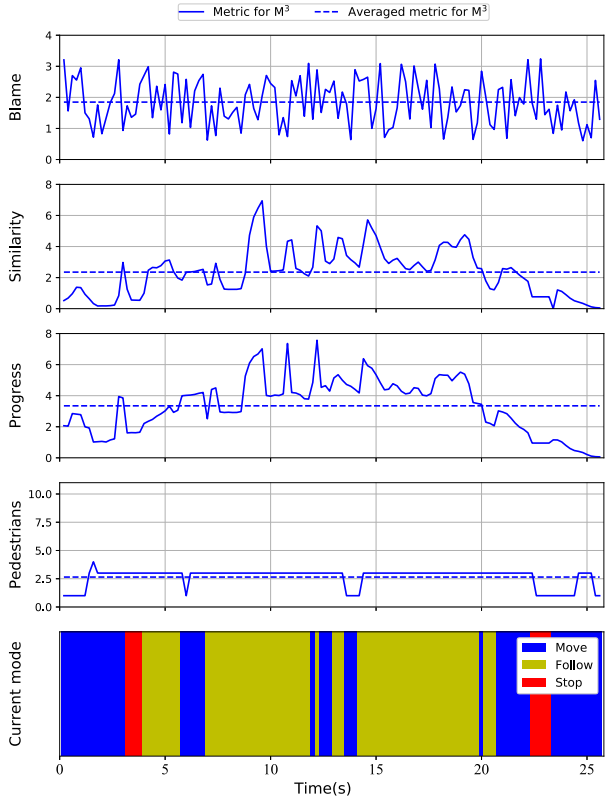


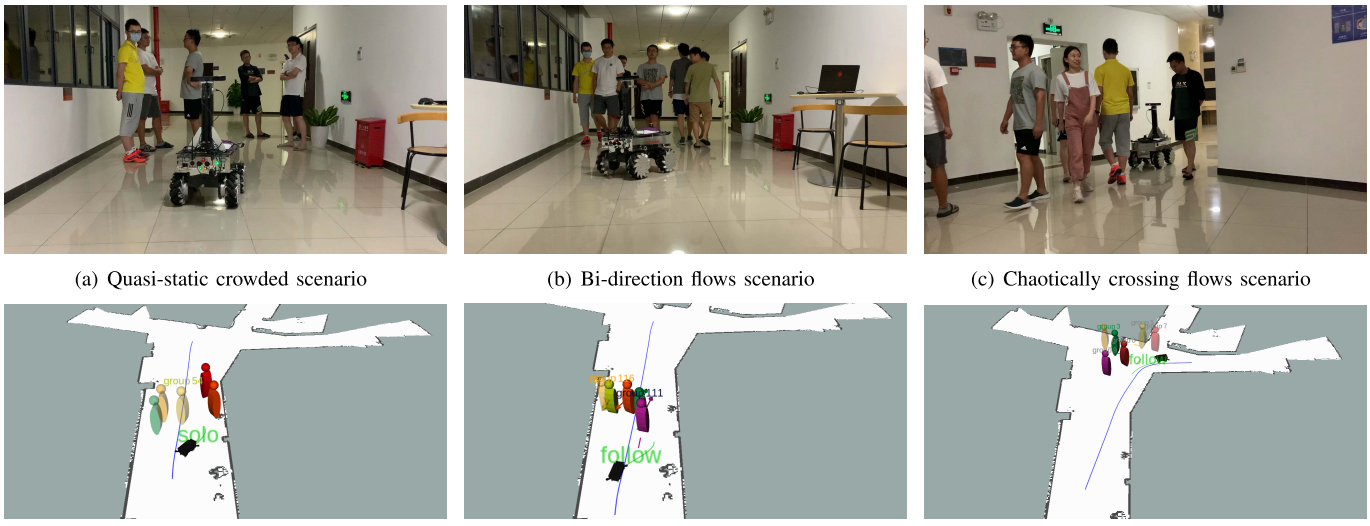
Fig. 12. The cost of blame, similarity, and progress of the robot with the proposed M^3 framework in the chaotically crossing flows scenario in real-world experiment are given. The number of pedestrians within 2 m and the mode selected in each time step of M^3 framework are also given.

similarity cost and progress cost, and fewer pedestrians around when it follows a leader. The robot can benefit from following the leader and then conform to similar social conventions as the leader for socially-compliant navigation. Moreover, the robot can autonomously adopt an appropriate navigation mode in densely crowded scenario.

C. Real-World Experiments

In these experiments, we test the mobile robot using the proposed M^3 framework to complete the navigation task in a real-world environment. The parameters of the scenarios in the experiments are listed in Table III, where the maximum velocity of the pedestrians is 1.3 m/s. There are 6 pedestrians to act as pedestrians. The map of the real-world is given (Building G at HITSZ), as shown in Fig. 11. The snapshots of each scenarios are given in Fig. 13.

In the real-world experiments, we test the proposed M^3 framework in three scenarios, which are consistent with the setup of scenarios in the simulation. In the first scenario of quasi-static crowds, the robot passes through two groups, as shown in Fig. 13(a). Six people divide into two groups. The robot moves through these two groups without invading the groups' space. In the second scenario of bi-direction flows, the robot passes through two flows walking in opposite directions. Seven pedestrians divide into two flows. The robot follows the flow with the same direction of movement. In the third scenario of chaotically crossing flows, the robot



(a) Quasi-static crowded scenario (b) Bi-direction flows scenario (c) Chaotically crossing flows scenario

(d) Navigation in quasi-static crowded (e) Navigation in bi-direction flows (f) Navigation in chaotically crossing flows

Fig. 13. Three scenarios are tested in the experiments. Group social relations are visualized with text among pedestrians. The global path is shown in blue while its executed trajectory is shown in green. The current robot's motion mode is shown above the best trajectory. **Left:** Quasi-static crowded scenario. **Center:** Bi-direction flows scenario with pedestrians moving in two distinct "flows." **Right:** Chaotically crossing flows scenario with pedestrians crossing each other's paths.

passes through a crossroad with pedestrians walking from one intersection to another. The robot autonomously chooses the optimal motion mode, as shown in Fig. 12. In the semicircle area of $2m$ in front of the robot, the average number of pedestrians is 2.7. So the density of the scenario reaches 0.43 person/m^2 (medium density).

The cost of blame, similarity, and progress of the robot in the experiment in chaotically crossing flows scenario are given in Fig. 12. The experimental results are coincident with those in the simulations, as shown in Fig. 10. The value of the blame metric is high because the pedestrians are very close during the robot navigation. It means that the robot is moving in dense crowds. The results show that the robot is more efficient when it is in the following motion mode. This is because the robot can benefit from collective motion by following pedestrians. However, the robot chooses to stop sometimes due to the chaotic interactions in this scenario. The efficiency of the robot declines when the robot stop.

VI. Conclusion

In this paper, we consider the navigation problem of a mobile robot in a dense crowd. In order to ensure the social-compliance of the robot so that it can complete the task without affecting the order and efficiency of the traffic, we propose a unified multiple-motion-mode framework of planning for socially-compliant navigation. The proposed framework consists of three successive phases, which accept the known map of the static environment, the global path from the initial state of the robot to the destination, and the nearby pedestrians detected in each time step as inputs and yields the linear and angular velocity of the robot as outputs.

The proposed framework consists of three phases, the identification of pedestrian groups and prediction of their trajectories, the generation of candidate local trajectories of the robot, and the determination of an optimal local trajectory. In the proposed framework, we propose to take a group of pedestrians as a whole in the navigation problem rather than taking each pedestrian into consideration, which can keep the robot not invading the groups' privately shared space and reduces the computational complexity of the pedestrians' trajectories prediction. Considering the different motion behaviors of the pedestrians, we propose three motion modes of the robot, the moving-solo mode, the pedestrian-following, and the courteous-stopping. In each mode, we propose different sampling strategies to find local terminal states of the robot, which are utilized to generate a set of candidate local trajectories. In order to determine the best local trajectory among the many candidates, we establish a composite metric based on three novel metrics, which reflect the advancement of the robot towards the final destination, the possibility of an oscillation motion, and the possibility of an intrusion into the shared spaces of groups. The proposed M^3 framework is evaluated both in simulations and real-world experiments with various densely crowded scenarios and the results show that the proposed framework surpasses the state-of-the-art methods in terms of efficiency and social compliance.

The proposed M^3 framework can be used for the various crowded and chaotic environments by following collective and individual social conventions. A possible challenge situation is to navigate in highly crowded environments. Even pedestrians are difficult to walk in these highly crowded situations. Sometimes they need to use their flexibility to move in these environments. Due to the constraints of the robot's dynamic and size, the robot's efficiency may decline.

In our future work, we want to explore how to incorporate more semantic information of dense crowds into the M^3 framework to make a robot behave like humans in a social environment. The robot can autonomously adjust the weights of the comprehensive function based on the semantic information of dense crowds.

ACKNOWLEDGMENT

The authors would like to thank Dr. Xiangming Xi, Harbin Institute of Technology, Shenzhen, for his valuable discussions and advices in manuscript preparation.

REFERENCES

[1] M. Shiomi, T. Kanda, D. F. Glas, S. Satake, H. Ishiguro, and N. Hagita, "Field trial of networked social robots in a shopping mall," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 2846–2853.

[2] A. Vasquez, M. Kollmitz, A. Eitel, and W. Burgard, "Deep detection of people and their mobility aids for a hospital robot," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2017, pp. 1–7.

[3] R. Triebel *et al.*, "SPENCER: A socially aware service robot for passenger guidance and help in busy airports," in *Field and Service Robotics* (Springer Tracts in Advanced Robotics), vol. 113. Springer, 2016, pp. 607–622.

[4] G. Ferrer and A. Sanfeliu, "Proactive kinodynamic planning using the extended social force model and human motion prediction in urban environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 1730–1735.

[5] H. Kretzschmar, M. Spies, C. Sprunk, and W. Burgard, "Socially compliant mobile robot navigation via inverse reinforcement learning," *Int. J. Robot. Res.*, vol. 35, no. 11, pp. 1289–1307, 2016.

[6] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 1343–1350.

[7] Y. Luo, P. Cai, A. Bera, D. Hsu, W. S. Lee, and D. Manocha, "PORCA: Modeling and planning for autonomous driving among many pedestrians," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3418–3425, Oct. 2018.

[8] R. Bresson, J. Saraydaryan, J. Dugdale, and A. Spalanzani, "Socially compliant navigation in dense crowds," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 64–69.

[9] S. Gulati, "A framework for characterization and planning of safe, comfortable, and customizable motion of assistive mobile robots," M.S. thesis, Mech. Eng., Univ. Texas Austin, Austin, TX, USA, 2011.

[10] Y. Morales, T. Miyashita, and N. Hagita, "Social robotic wheelchair centered on passenger and pedestrian comfort," *Robot. Auto. Syst.*, vol. 87, pp. 355–362, Jan. 2017.

[11] X.-T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 4, pp. 1743–1760, Oct. 2017.

[12] 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.

[13] R. Kirby, "Social robot navigation," Ph.D. dissertation, School Comput. Sci., Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, 2010.

[14] M. Kuderer, H. Kretzschmar, C. Sprunk, and W. Burgard, "Feature-based prediction of trajectories for socially compliant navigation," in *Proceedings of Robotics: Science and Systems*, vol. 26. Cambridge, MA, USA: MIT Press, 2012, pp. 193–200.

[15] W. H. Warren, "Collective motion in human crowds," *Current Directions Psychol. Sci.*, vol. 27, no. 4, pp. 232–240, Aug. 2018.

[16] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 51, no. 5, p. 4282, 1995.

[17] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *Int. J. Social Robot.*, vol. 7, no. 2, pp. 137–153, Sep. 2015.

[18] D. Helbing, P. Molnár, I. J. Farkas, and K. Bolay, "Self-organizing pedestrian movement," *Environ. Planning B, Planning Des.*, vol. 28, no. 3, pp. 361–383, Jun. 2001.

[19] S. Pellegrini, A. Ess, K. Schindler, and L. van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *Proc. IEEE 12nd Int. Conf. Comput. Vis.*, Sep. 2009, pp. 261–268.

[20] K. Kitazawa and T. Fujiyama, "Pedestrian vision and collision avoidance behavior: Investigation of the information process space of pedestrians using an eye tracker," in *Pedestrian and Evacuation Dynamics 2008*. Berlin, Germany: Springer, 2010, ch. 7, pp. 95–108.

[21] G. Ferrer and A. Sanfeliu, "Anticipative kinodynamic planning: Multi-objective robot navigation in urban and dynamic environments," *Auto. Robots*, vol. 43, no. 6, pp. 1473–1488, Aug. 2019.

[22] W. Chi, C. Wang, J. Wang, and M. Q.-H. Meng, "Risk-DTRRT-based optimal motion planning algorithm for mobile robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1271–1288, Jul. 2019.

[23] X. Zhang, J. Wang, Y. Fang, and J. Yuan, "Multilevel humanlike motion planning for mobile robots in complex indoor environments," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 3, pp. 1244–1258, Jul. 2019.

[24] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "Understanding human interaction for probabilistic autonomous navigation using risk-RRT approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2011, pp. 2014–2019.

[25] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, "The walking behaviour of pedestrian social groups and its impact on crowd dynamics," *PLoS ONE*, vol. 5, no. 4, Apr. 2010, Art. no. e10047.

[26] D. C. Duives, W. Daamen, and S. P. Hoogendoorn, "State-of-the-art crowd motion simulation models," *Transp. Res. C, Emerg. Technol.*, vol. 37, pp. 193–209, Dec. 2013.

[27] B. Okal and K. O. Arras, "Learning socially normative robot navigation behaviors with Bayesian inverse reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2016, pp. 2889–2895.

[28] T. Linder and K. O. Arras, "Multi-model hypothesis tracking of groups of people in RGB-D data," in *Proc. IEEE Int. Conf. Inf. Fusion (FUSION)*, Jul. 2014, pp. 1–7.

[29] A. Vemula, K. Muelling, and J. Oh, "Modeling cooperative navigation in dense human crowds," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2017, pp. 1685–1692.

[30] C. Scholler, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1696–1703, Apr. 2020.

[31] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a frenet frame," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 987–993.

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

[33] A. Astolfi, "Exponential stabilization of a wheeled mobile robot via discontinuous control," *J. Dyn. Syst., Meas., Control*, vol. 121, no. 1, pp. 121–126, Mar. 1999.

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

Yujing Chen (Member, IEEE) received the B.E. and M.E. degrees in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree in control science and engineering with the Harbin Institute of Technology, Shenzhen, China.

His current research interests include motion planning and socially compliant robot navigation.

Yunjiang Lou (Senior Member, IEEE) received the B.S. and M.E. degrees in automation from the University of Science and Technology of China, Hefei, China, in 1997 and 2000, respectively, and the Ph.D. degree in electrical and electronic engineering from The Hong Kong University of Science and Technology, Hong Kong, in 2006.

He is currently with the State Key Laboratory of Robotics and Systems, School of Mechatronics Engineering and Automation, Harbin Institute of Technology, Shenzhen, China, and the Shenzhen Key Laboratory for Advanced Motion Control and Modern Automation Equipments, Shenzhen. His research interests include motion control, mechanism design, and industrial robots.