



# Robot Navigation Based on Predicting of Human Interaction and its Reproducible Evaluation in a Densely Crowded Environment

Yuichi Kobayashi<sup>1</sup> · Takeshi Sugimoto<sup>2</sup> · Kazuhito Tanaka<sup>5</sup> · Yuki Shimomura<sup>3</sup> · Francisco J. Arjonilla Garcia<sup>1</sup> · Chyon Hae Kim<sup>4</sup> · Hidenori Yabushita<sup>5</sup> · Takahiro Toda<sup>5</sup>

Accepted: 26 April 2021 / Published online: 23 May 2021  
© The Author(s) 2021

## Abstract

To achieve robot navigation in crowded environments having high densities of moving people, it is insufficient to simply consider humans as moving obstacles and avoid collisions with them. That is, the impact of an approaching robot on human movements must be considered as well. Moreover, various navigation methods have been tested in their own environments in the literature, which made them difficult to compare with one another. Thus, we propose an autonomous robot navigation method in densely crowded environments for data-based predictions of robot-human interactions, together with a reproducible experimental test under controlled conditions. Based on localized positional relationships with humans, this method extracts multiple alternative paths, which can implement either following or avoidance, and selects an optimal path based on time efficiency. Each path is selected using neural networks, and the various paths are evaluated by predicting the position after a given amount of time has elapsed. These positions are then used to calculate the time required to reach a certain target position to ensure that the optimal path can be determined. We trained the predictor using simulated data and conducted experiments using an actual mobile robot in an environment where humans were walking around. Using our proposed method, collisions were avoided more effectively than when conventional navigation methods were used, and navigation was achieved with good time efficiency, resulting in an overall reduction in interference with humans. Thus, the proposed method enables an effective navigation in a densely crowded environment, while collecting human-interaction experience for further improvement of its performance in the future.

**Keywords** Mobile robot navigation · Crowded environment · Human interaction · Planning

## 1 Introduction

### 1.1 Research Background and Motivation

One expected application of mobile robots is the task of guiding users in environments with high densities of pedestrians such as railway stations, shopping malls, and stadiums [1]. Due to improvements in human recognition technologies

[2] and simultaneous localization and mapping (SLAM) [3], mapping and navigating environments, while recognizing and avoiding humans, has become feasible for mobile robots. In the field of mobile robot navigation, various approaches that employ sensors placed within the environment have been implemented, but this study achieves navigation by recognizing humans with sensors installed on the robot itself, to reduce the operating costs incurred when the robots are introduced to new environments. When operating in environments where humans are active, robots must avoid causing feelings of discomfort or obstruction in humans, in addition to avoiding collisions with them. Furthermore, pedestrians are not simply moving obstacles, but they also have the unique characteristics of altering their movement in response to the robots, making collision avoidance more challenging.

Meanwhile, to promote development of such navigation in human-interactions, it is important to have a common testbed. Since the humans' behavior is affected by various factors

✉ Yuichi Kobayashi  
kobayashi.yuichi@shizuoka.ac.jp

<sup>1</sup> Department of Mechanical Engineering, Shizuoka University, Hamamatsu, Japan

<sup>2</sup> Hewlett-Packard Japan, Ltd., Tokyo, Japan

<sup>3</sup> Persol Research & Development, Nagoya, Japan

<sup>4</sup> Sky Ocean Technology Co. Ltd., Jeju-si, South Korea

<sup>5</sup> Toyota Motor Corporation, Toyota, Japan

such as density, destination, and obstacles other than a robot, considering scenarios of human's behavior carefully is necessary according to objectives of experiments.

## 1.2 Related Research

When navigating in environments with active humans, route planning and speed control are necessary for avoiding collisions with humans. Considering pedestrians as moving obstacles, Berg et al. implemented multi-agent navigation through a speed control method called reciprocal velocity obstacle (RVO) [4]. While effective, RVO does not explicitly address interactions between pedestrians and robots, so robots exhibited abnormal proximity to the surrounding humans. By introducing a Kalman filter to RVO, Kim et al. enabled uncertainty to be handled, thus solving the problem of abnormal proximity [5]. However, introducing uncertainty into this navigation method triggered the Freezing Robot problem (FRP) [6], where the robot becomes unable to move in high-density, crowded environments, due to high uncertainty of collision with pedestrians.

Trautman et al. proposed to solve the FRP under a cooperative collision avoidance scenario between robot and humans. However, generally speaking, we cannot guarantee that all pedestrians always cooperate with robots. Of course, on the contrary, if pedestrians are completely adversarial to robots, FRP can never be solved. This suggests that some explicit communication, such as asking pedestrians to make room for the robot, will be effective to resolve the problem. Nevertheless, we believe that focusing on navigation against non-cooperative pedestrians will certainly contribute in building navigation strategy in a densely crowded environment. This is because improvement of performance in such non-cooperative environments will reduce necessity of robots to ask pedestrians to make room for them.

From the perspective of consideration of robot-human interaction in navigation, Mehta proposed a method [7] based on expanding Multi-Policy Decision Making [8]. Here, decision policies are designed in advance and may include Go-Solo, Follow, and Stop. By switching between these policies, according to the state of surrounding humans, the robot can select behaviors, such as Stop, rather than simple obstacle-avoidance, Go-Solo, that may cause FRP. In this case, a geometric prediction of human behavior is used when each policy is evaluated, but it is unclear whether humans conform completely to this model. If robots could predict interactions based on the behaviors that humans actually exhibit in response to robots, they could navigate in harmony with the real world. Studies [9] on predicting humans' interactions exist, but focus on one-to-one encounters between humans and robots. Contrarily, our work attempts to predict

interactions when multiple pedestrians are crowded densely together. Furthermore, it is believed that local observations of individual positions can become inaccurate, in environments where humans move in crowds, because of occlusion and density of human legs/bodies. In this case, methods that enable robots to account for overlapping humans when planning motions, such as 'Avoid approaching humans' or 'Follow humans walking ahead in the same direction', can be more effective.

Additionally, Abbeel reported using inverse reinforcement learning to teach robots human behavior [10], which enabled robot navigation in complex environments, such as within crowd flows, to be taught via human instructions. Furthermore, Kretzschmar et al. proposed learning of socially compliant behavior model based on pedestrian behavior or tele-operated robots behavior via inverse reinforcement learning [11]. Based on a probabilistic modeling, their model could capture both discrete navigation decision and continuous (relatively small) variance of human trajectories. However, the approach of inverse reinforcement learning is vulnerable to bias and other undesirable effects, as it relies on the specific traits of the human instructor or environments where data are collected. Our approach, on the contrary, focuses only on local interaction between humans and robots even within the same data-driven approach. Therefore, it is expected that the behavioral model can be constructed in a local and less-environmentally influenced scope.

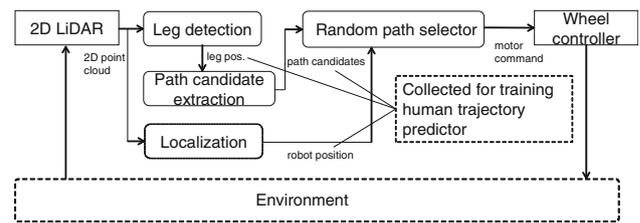
Chen et al. used deep reinforcement learning [12] to teach robots human social norms [13]. These norms, such as "when walking on the right, pass people on the left" are implicit rules for humans in crowds, but were taught to the robot by providing the movement, when passing-each-other, as a reward to the robot. They were thereby able to achieve navigation that did not cause discomfort in the surrounding humans. In environments where humans are present in high densities, however, the robots should move with the flow of pedestrians going in the same direction rather than following the social norms to avoid collisions with every individual. Therefore, teaching social norms is difficult to apply to high-density, crowded environments. Furthermore, this method faces challenges when responding to planning requirements for effective robot motion, considering global optimality.

Finally in most related works, unfortunately, experimental evaluations have been conducted under varying crowd conditions, which directly affects the evaluation of the navigation ability in high-density crowded environments and makes it difficult to objectively verify their effectiveness via reproduction. Thus, another important issue in this field of robotic navigation in crowded environments is that a common reproducible and controlled test has not yet been proposed.

### 1.3 Research Objective

We propose here a navigation method wherein robots navigate while making predictions, by learning human interaction in high-density, crowded environments. We propose a method for extracting multiple paths, based on the distribution of nearby humans, as a method that expresses multiple navigation strategies, for both avoiding and following, under specific circumstances where the robot is surrounded by nearby pedestrians. Based on extracting paths and acquiring an interaction model, the robot trajectory upon path selection can be predicted while considering humans' interactions. Using of acquired interaction model, the robot can predict excessive human interference, through low-speed robot movement caused by the interference. By combining a localized interaction prediction and global movement planner, the optimal path can be chosen from multiple options. We tested the proposed navigation method in a real-world environment with many pedestrians to validate its effectiveness in a controlled and reproducible way.

This study provides following contributions to the field of mobile robot navigation in crowded environment. First, we present tests of mobile robot navigation in a continuous flow of humans with an explicitly-controlled density. The conditions and settings make the evaluation process of mobile robot navigation in highly crowded environment reproducible and comparable. Second, not only is the robot able to avoid humans in one-on-one situations, but also it can avoid or follow a crowd of pedestrians through the localized path extraction method. In a previous study [14], navigation was achieved in an environment that was crowded with many people, but a ceiling-mounted camera had to be used to measure the humans. It should be noted that our study uses only sensors embedded on the robot itself, realized by expressing the path from the viewpoint of the robot based on the its local sensing. Second, this work provides a framework that enables data collection about human interactions that can be used for predictions. This enables predictions that incorporate human behavior in actual environments, making this framework distinct from motion planning approaches such as pedestrian optimal reciprocal collision avoidance (PORCA) [15], which addresses human intention and uncertainty as a partially observable Markov decision process, because it can be positioned as an experience-based method. Finally, this method considers optimization in combination with global path planning. This is advantageous in comparison with those methods based on information of control by human operator, such as inverse reinforcement learning.



**Fig. 1** Robot control architecture for data collection. Collected human leg position and robot's trajectories are used for construction of robot trajectory predictor

## 2 Problem Settings

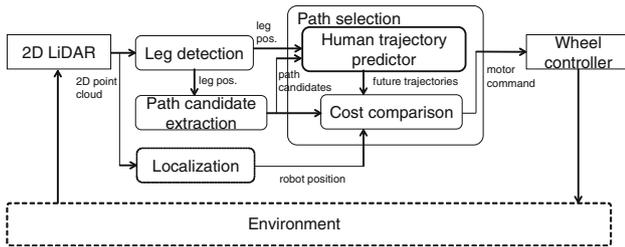
For this study, we assume a crowded environment where humans tend to generate some flows due to a high density. To ensure focus on human interactions, no obstacles were placed in the region around the start and target positions. A mobile robot navigates itself toward its destination using only its onboard sensor information and is not supposed to have an explicit communication with pedestrians (e.g., it does not ask humans to make room for itself by gesture or utterance). The primary goal of the robot is to quickly reach a defined target position, beginning from a fixed initial position, while also avoiding collisions with humans. The secondary goal is to avoid causing any psychologically oppressive feelings because of traversing against pedestrians.

For implementation and validation of the mobile robot navigation, we rely on a simulator for data collection of building a predictor and experiments with real human subjects for validation of the proposed navigation. Details of the simulator are described in appendix, and experimental conditions for validation will be described in Sect. 4.

## 3 Navigation Method Based on Path Extraction and Selection

Figures 1 and 2 provide an overview of the proposed navigation method. Figure 1 outlines the offline data collection process on the simulator, used for configuring the predictor, which finally predicts human interactions. Path candidates are extracted based on 2D LiDAR point cloud via leg detection module. One of paths is randomly selected for a certain duration for navigation. The robot trajectory is recorded while the path remains selected. The resulting data are used to configure a human/robot trajectory predictor. The predicted trajectory reflects interaction between the robot and its nearby pedestrians.

At the online navigation stage, as shown in Fig. 2, the predictor forecasts the trajectories and arrival positions, several seconds into the future, given that the extracted candidate path is selected. Integrating the predicted results and the



**Fig. 2** Robot control architecture for online navigation. Constructed human trajectory predictor is used for path selection

global path planning results ensures that the most efficient path is selected, as determined by the shortest time of arrival to the target position.

### 3.1 Extracting Path Candidates Based on Local Observations

In this study, two types of path extractors, each expressing one of two strategies, are configured to extract the robot paths. One is the Avoidance strategy, in which the robot avoids humans and aims for the goal, while the other is the Follow strategy, in which the robot follows humans. To extract path candidates, a one-dimensional potential function is defined using angle  $\phi$  [rad] in a polar coordinate system with the robot at the origin. This function is a simplification of the well-known artificial potential field method, defined over a 2D plane, to one dimensional (1D), angle, space. It expresses the difficulty of moving forward in each possible driving direction in the robot’s immediate surroundings, where a negative number indicates that the movement is very easy (Fig. 3).

Assuming  $i$  denote the index of human recognized by the robot, (1) shows the calculation of the potential function,  $P_i^{\text{follow}}(\phi)$ , for extracting the path to follow humans that are walking in the same direction as the robot. Here, in the robot coordinate system, the polarization angle of the polar coordinate is  $\phi$ , the human  $i$  direction is  $\theta_i$ , and the direction of the goal is  $\theta_g$ . These angles are 0 deg in the front view of the robot, and the right side is defined as positive.  $A_i^{\text{follow}}$  and  $\sigma_{\text{follow}}$  are each defined below, in (2) and (3), respectively.

$$P_i^{\text{follow}}(\phi) = A_i^{\text{follow}} \exp \left\{ -\frac{(\phi - \theta_i - 0.1\theta_g)^2}{2\sigma_{\text{follow}}^2} \right\} \quad (1)$$

$$A_i^{\text{follow}} = \begin{cases} (R_{v_i}^{\text{follow}} + 0.1) \exp \{-(r_i - 1)\} & \text{if } r_i \geq 1 \\ (R_{v_i}^{\text{follow}} + 0.1) & \text{otherwise} \end{cases} \quad (2)$$

$$\sigma_{\text{follow}} = \begin{cases} \exp(0.1r_i - 2) & \text{if } r_i \geq 6.1 \\ 0.25 & \text{otherwise} \end{cases} \quad (3)$$

This potential function is configured using a Gaussian function and designed so that locally minimum values can be obtained from the direction where humans are moving

away from the robot. The distance to detected human  $i$  is defined as  $r_i$ , the component of the direction toward human  $i$  in the robot velocity is  $R_{v_i}^{\text{follow}}$ , and  $A_i$  is the coefficient that relates to the potential size.

$A_i^{\text{follow}}$  and  $\sigma_{\text{follow}}$  are used to express that moving with a faster speed away from the robot (large  $R_{v_i}^{\text{follow}}$ ) results in a greater influence from the human, and humans that are further away (large  $r_i$ ) exhibit less influence. The amplitude of the Gaussian function is expressed by  $A_i$ , and humans that are further away have correspondingly smaller impacts. Additionally, the coefficient  $\sigma_{\text{follow}}$  relates to the horizontal width of the potential, and a narrower width indicates a narrow range that exerts an effect when (4) is superimposed. Therefore, this coefficient is set so that the potential width of humans narrows with decreasing  $r_i$ . Within a certain range,  $\sigma_{\text{follow}}$  was set constant at 0.25 to prevent a too narrow width. The threshold value of 6.1 in (3) was decided so that two divided cases have almost equal values at  $r_i = 6.1$ , preventing sudden change of  $\sigma_{\text{follow}}$ . The potential,  $P_i^{\text{follow}}$ , which is used to extract paths, is defined from the sum of the potential  $P_i^{\text{follow}}$  for each individual human  $i$ .

$$P^{\text{follow}}(\phi) = \sum_{i=1}^N P_i^{\text{follow}}(\phi) \quad (4)$$

When following humans, the direction  $\phi$  for humans that are moving away from the robot is minimal, and the robot can then engage in following these humans.

Figure 4 shows an example of a path extracted using  $P^{\text{follow}}$ . The left panel shows a distribution of surrounding humans, and the right side shows the corresponding  $P^{\text{follow}}(\phi)$ . The lines on the human dots represent velocity. In the left panel, there is a total of eight humans and two humans are present in front (slightly to the right) and two humans in the front left of the robot. One of the extracted direction, along indicating that paths for following groups of humans have been extracted.

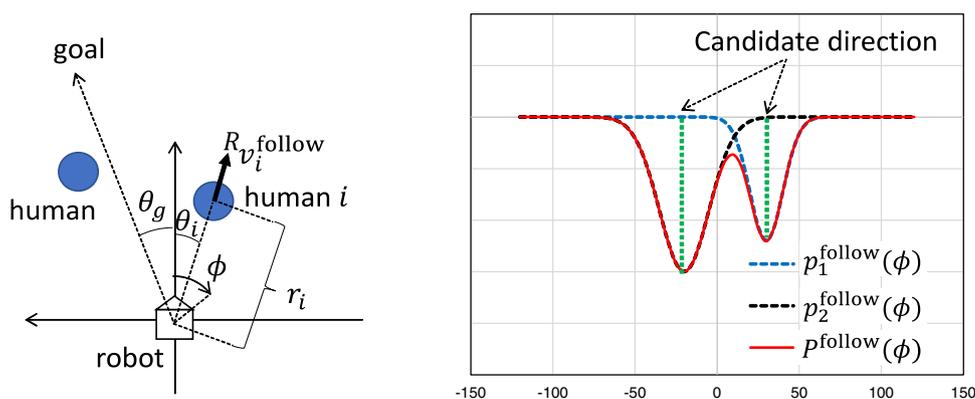
Two types of avoidance path candidates are also extracted. One is the path passing between humans, based on  $P_i^{\text{avoid}}(\phi)$ , and the other is the path going around the outside of a group of humans at the edge of the robot’s field of vision. Regarding the first candidates,  $P_i^{\text{avoid}}(\phi)$ ,  $A_i^{\text{avoid}}$ , and  $\sigma_{\text{avoid}}$  are defined as

$$P_i^{\text{avoid}}(\phi) = A_i^{\text{avoid}} \exp \left\{ -\frac{(\phi - \theta_i)^2}{2\sigma_{\text{avoid}}^2} \right\} \quad (5)$$

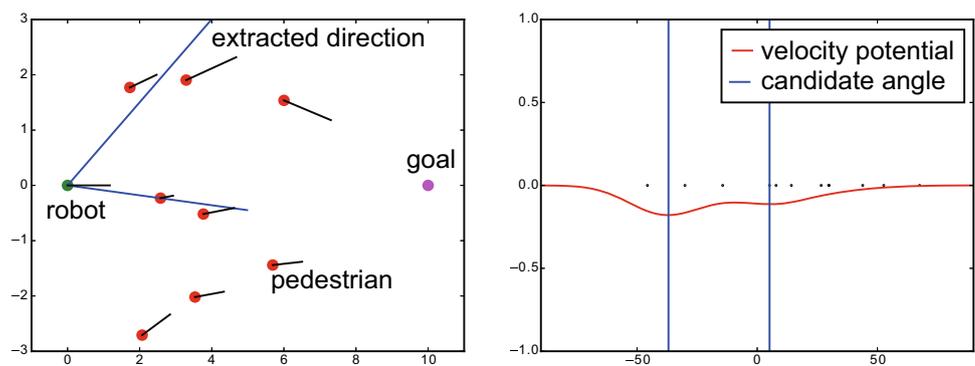
$$A_i^{\text{avoid}} = \begin{cases} \exp \{-(r_i - 1)\} & \text{if } r_i \geq 1 \\ 1 & \text{otherwise} \end{cases} \quad (6)$$

$$\sigma_{\text{avoid}} = \begin{cases} \exp(0.1r_i - 2) & \text{if } r_i \geq 2.9 \\ 0.18 & \text{otherwise.} \end{cases} \quad (7)$$

**Fig. 3** Parameter definitions for potential  $P^{follow}(\phi)$ (left) and an example of its component (right). local minima are used for candidates of paths



**Fig. 4** Example of following directions. There are two groups of humans walking in similar directions to the robot and two directions are extracted to follow them. The two directions in the left panel correspond to two local minima in the potential in the right panel



The calculated 1D potential minimum value yields a path that passes between people. The 1D potential used to extract this path is obtained from the sum of the detected human  $i$  potentials shown in (5). Further, when  $r_i$  decreases below 2.9 m,  $\sigma_{avoid}$  assumes a constant value of 0.18 to prevent  $\sigma_{avoid}$  from becoming too small, which could cause the extraction of a path too close to a human. Similar to the case of (3), a threshold value  $r_i = 2.9$  was decided for maintaining continuity of  $\sigma_{avoid}$  in the divided cases.

$$P^{avoid}(\phi) = \sum_{i=1}^N p_i^{avoid}(\phi) + p_g(\phi) \tag{8}$$

The first term in this equation is the potential function with a minimal gap between humans. The second term,  $p_g(\phi)$ , is the potential function, defined in (9), and it minimizes the goal direction.

$$p_g(\phi) = -B \exp \left\{ -\frac{(\phi - \theta_g)^2}{\beta} \right\} \tag{9}$$

Here, the parameters  $B$  and  $\beta > 0$  determine the shape of the potential function.

Regarding path candidates going outside of humans, in many cases, the potential decreases monotonically at both side limits of the 2D LiDAR field of vision, and the minimum cannot be defined. Thus, to extract a path that goes around the outside of a group of humans, the potential function value on

the outer side of the field of view from the maximum nearby potential is approximated by the following equation since the path away from the maximum nearby potential is extracted at a constant interval angle.

$$\hat{p}(\phi) = A_{round} \exp \left( -\frac{\phi^2}{\sigma_{round}^2} \right) \tag{10}$$

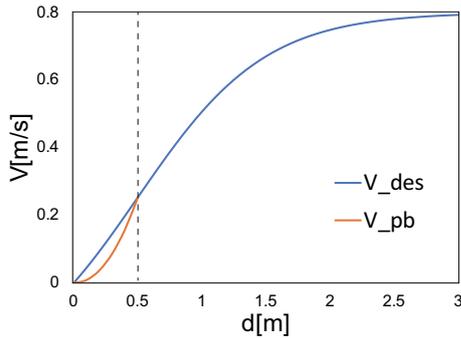
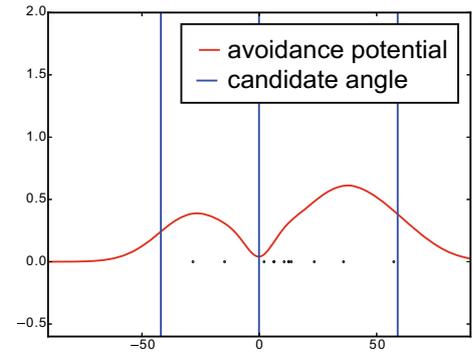
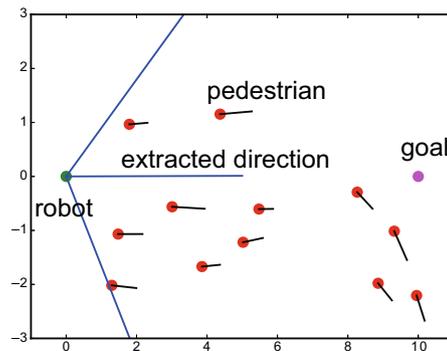
Here,  $A_{round}$  sets the maximum nearby potential and estimates  $\sigma_{round}$  through the least-squares method. Based on this estimated value, the direction where  $\sigma_{round}$  is displaced, towards the edge of the field of view, from the maximum direction, becomes the path candidates  $\phi_{round}^L$  and  $\phi_{round}^R$ .

The left panel of Fig. 5 shows an example distribution of humans and the corresponding potential,  $P^{avoid}(\phi)$  is shown in the right panel. With  $P^{avoid}(\phi)$  minimized, a path between human groups can be extracted, but the minimum value cannot be detected on either side. Clearly, a direction of travel that goes around the outside of the group has been extracted in Fig. 5 as two paths going around the group,  $\phi_{round}^L$  and  $\phi_{round}^R$  can be readily observed.

### 3.2 Robot Velocity Control

In addition to determining the direction of travel, as described in the previous section, we used the optimal velocity (OV) model [16] to ensure that the robot moves at velocities appropriate for its distance from humans. This model is typically

**Fig. 5** Example of obstacle avoidance and round directions. Three candidate directions are extracted, center of which directs to the gap between two groups of humans. The left and right directions are to go around the human groups



**Fig. 6** Desired velocity profile of OV model  $V_{des}(d)$

used as a vehicle control model and it well describes steady flow and congestion, not only for traffic system but also for pedestrians’ flow (e.g., [17]). Thus, the OV model is used as a method for controlling the robot velocity, so that the robot can smoothly join a human’s flow in the case of the Follow strategy.

Given  $d$ , the distance from the robot to the closest human, the target velocity  $V_{des}(d)$  is determined as

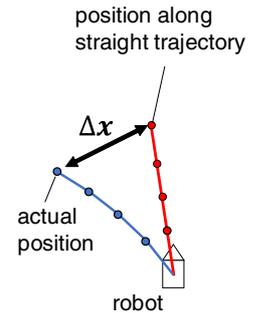
$$V_{des}(d) = \frac{V_{max}}{1 + \tanh(\beta c)} \{ \tanh(\beta d - c) + \tanh(\beta c) \}, \quad d > 0.5 \tag{11}$$

$$V_{pb}(d) = 2dV_{des}(d), \quad 0 \leq d \leq 0.5 \tag{12}$$

Figure 6 shows  $V_{des}(d)$  and  $V_{pb}(d)$ . At 0.5 m or below, marked by the dotted line,  $V_{pb}$  causes the robot to decelerate rapidly to avoid a collision.

Furthermore, we consider the  $\pm 30$  deg direction of the global path from the robot viewpoint, which is the direction leading straight to the goal if there are no obstacles except for pedestrians. To avoid wasting motion when there are no obstacles near the robot, if the sensor value indicates that no obstacles exist within 5 m, the robot advances at 1.0 m/s in the direction of this global path.

**Fig. 7** Definition of trajectory of robot



### 3.3 Data Collection and Predictor Configuration

On the simulator, data are collected for interactions between the robot and humans, when they travelled in proximity according to paths determined by the extraction method described above. The trajectory predictor is then configured from that data (Table 1). As described in Sect. 3.1, the path candidates are classified into two groups: “Avoid,” which passes through the gaps between and around humans, and “Follow,” which follows humans. Consequently, two corresponding types of trajectory predictors are created using multi-layered neural networks. The predicted interactions, shown in Fig. 7, are expressed as displacement from the trajectory, when the robot is moving straight forward at a steady velocity in the direction of the path selected at any point in time, the ‘straight-ahead’ route (Table 1).

The definitions for these element inputs are shown in Fig. 8. Displacement from the straight-ahead route is expressed as  $\Delta \mathbf{x}_n \in \mathbb{R}^2, n = 1, \dots, N$ , where  $n$  denotes a discrete time with time step  $\Delta t$ , and  $N$  denotes the maximum discrete time for the prediction. Thus, the output of the approximator for each function is a  $2N$ -dimensional vector. Table 2 shows common settings of the avoiding/following motion predictor neural networks. To collect a variety of moving directions, random selection from multiple candidate directions was controlled so that directions with less samples collected in the past are selected more frequently. Ten percent of the training samples were used for testing. In the process of training, loss functions of both training data and test data equally converged, which indicated absence of overfitting.

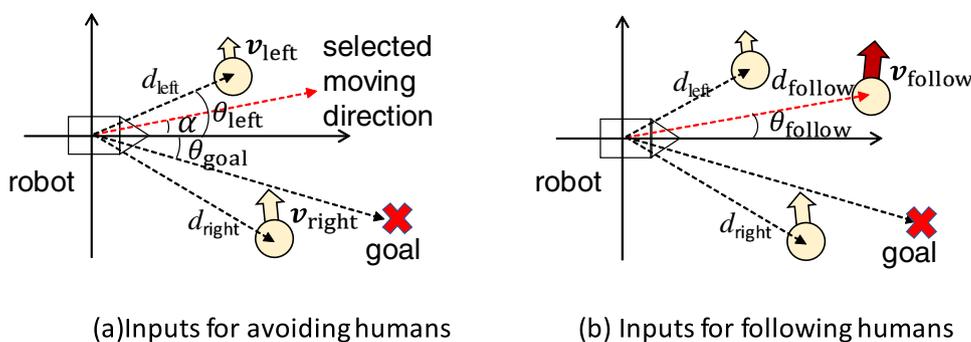
**Table 1** Inputs for robot trajectory predictors

Avoiding motion predictor		Following motion predictor	
Selected direction	$\alpha$	Selected direction	$\alpha$
Goal direction	$\theta_{goal}$	Velocity of followed human	$v_{follow}$
Velocity of left & Right human	$v_{left}, v_{right}$	Position of followed human	$d_{follow}, \theta_{follow}$
Position of left person	$d_{left}, \theta_{left}$	Position of left person	$d_{left}, \theta_{left}$
Position of right person	$d_{right}, \theta_{right}$	Position of right person	$d_{right}, \theta_{right}$

**Table 2** Settings of motion predictors

Number of middle layers	1
Number of neurons in the middle	20
Optimization method	Ada Grad method [26]
Loss function	Average squared error
Number of training samples	14,300 (avoiding), 14,640 (following)

**Fig. 8** Inputs for robot trajectory prediction

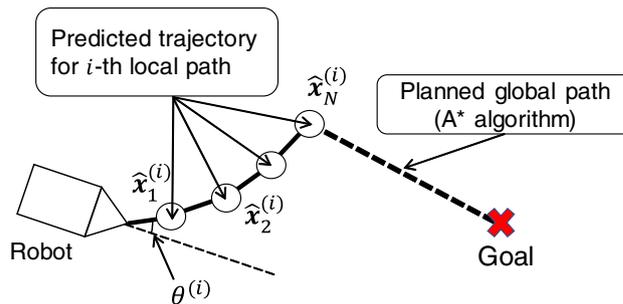


Among the inputs for each predictor in Fig. 8, the position and velocity of humans on either side of the travel direction serve as inputs for Avoid. Of the humans on either side, those close to the robot’s direction of travel and to the robot itself are selected. For Follow, the position and velocity of humans to be followed, as well as the positions of humans on either side, are input. In both cases, one of the two indicators of human distance and direction to the left and right is randomly selected from the Pareto Optimal Solution. If no humans are present to the left or right, the distance and angle are each set as 1000.

### 3.4 Online Navigation Method

During online navigation, the robot extracts candidate paths using the extraction method described in Sect. 3.1. After the path selection described below has been conducted, drive control, based on velocity control, is conducted as described in Sect. 3.2.

Figure 9 shows the predicted trajectory when the  $i$ th path candidate is selected. Based on the  $N$ th position,  $\hat{x}_N^{(i)}$ , of the positions from the predictor described in Sect. 3.3, the global path planner calculates the path to the goal position, using the A\* algorithm. The path length is  $L^{(i)}$ .



**Fig. 9** Predicted local trajectory and global path

When the candidate path  $i$  is selected, the total time cost,  $C_{total}^{(i)}$ , to reach the goal is defined by

$$C_{total}^{(i)} = C_{local}^{(i)} + C_{global}^{(i)} = \left( \frac{\Delta\theta^{(i)}}{\omega_{robot}} + N\Delta t \right) + \frac{L^{(i)}}{v_{robot}} \tag{13}$$

Here,  $C_{local}^{(i)}$  is the time cost required to move  $N$  steps along the selected path, and it corresponds to the parenthetical term on the second line of (13). The first item inside these parentheses is the time cost for the robot turning direction, starting from the direction it is currently facing. The quotient of the

angle of deflection  $\theta^{(i)}$  toward path  $i$  is used, with the rotational velocity  $\omega_{\text{robot}}$  [rad/s] kept constant. The second item inside the parentheses is the time cost for  $N$  steps. Since the observational period  $\Delta t$  is constant, this value is a constant common to all candidate paths. The third term, after the parenthesis, expresses the time cost to arrive at the goal position after localized path selection has occurred, and is obtained by dividing the length of the route  $L^{(i)}$ , obtained from the global planner, by the velocity of the robot,  $v_{\text{robot}}$ . To simplify the global cost estimation, we assume that the robot moves to the goal at a constant velocity. A selection is made, among the candidate paths, using the following equation to minimize the time cost described above.

$$i^* = \underset{i}{\operatorname{argmin}} C_{\text{total}}^{(i)} \quad (14)$$

This path selection is performed at  $\Delta t$  [s] intervals. During this time, the robot follows the path candidates described in Sect. 3.1, which are refreshed with each observational frame, and implements the velocity control described in Sect. 3.2.

## 4 Experiment

We confirmed, both in the simulator and the actual mobile robot, that the proposed method enabled the robot to arrive at its destination, without collisions, in an environment with a human flow of approximately 0.2 humans/m<sup>2</sup>. We then validated these robot-human interactions with an increased density of 0.5 humans/m<sup>2</sup>, which corresponds to an environment full of moving crowds, such as a crowded railway station. This presents a significant challenge for navigation methods simply designed to avoid collisions with single humans.

Our proposed method was constructed by the data collected by the simulator, and evaluated by actual mobile robots and pedestrians. There are two primary reasons for collecting data in this way:

- It cannot be said that human behavior on a ROS-stage-based simulator adequately represents true human behavior. Specifically, it is difficult to bridge the differences in behavior with actual movements, including avoidance behaviors when near other humans or a robot, and large repulsion-based movements. Thus, we do not believe that the detailed verification of robot-human interference is meaningful in this simulation.
- It can, however, be acknowledged that within the range where humans do not get too close to each other and to robots, avoidance movements in the simulator have a certain degree of realistic spontaneity. Therefore, we believe that there is value in verifying a learner, which

has learned behaviors, before approaching point-blank range.

### 4.1 Robot Condition

The robot model used in the simulator was assumed to be an omnidirectional mobile robot with a diameter of 0.45 m and a height of 1 m, while these values were 0.56 and 0.9 m for the actual experimental robot. Followings are the robot and simulation settings: walking speed of 0–1.0 m/s, turning angular velocity of  $-90$  to  $90$  deg/s, generatable acceleration of  $-1$  to  $1$  m/s<sup>2</sup>, and angular acceleration of  $-90$  to  $90$  deg/s<sup>2</sup>. A two-dimensional (2D) LiDAR sensor with a view angle of  $-90$  to  $90$  deg, a measurable distance of 0.1–30 m, and a resolution of 0.25 deg was embedded in the robot. Within the sensor measurement range, human positions were measured, and a differential was used to estimate their speeds. If humans overlapped, as is the case with robot navigation in a densely crowded environment, only the front human was viewed. The simulator equipment was based on a ROS Stage [18], and details of this simulator is described in appendix.

### 4.2 Design of Reproducible Environmental Condition

Figure 10 shows the experimental environment. Humans move in the direction of the arrow, within the range indicated by the 2 m wide and 8 m long gray rectangle. There are eight humans inside this range, arranged so that their density is always 0.5 humans/m<sup>2</sup>. Six points are defined on each side of the pedestrian flow, denoted A1–A3 and B1–B3, and the robot goes back and forth between each A point and the three B points. Crossing the flow once and arriving at the destination on the opposite side counts as one trial. In this way, each method is tested 18 times. We requested the following of the subjects while they were walking: to maintain the regulated density, enter the area as soon as another person leaves; when within the designated area, walk at a natural velocity; choose where to walk inside the area at random, preferably not along the middle line; do not alter behavior when approaching the robot and keep this response consistent; and do not learn or predict the robot behavior by its repetition.

When returning to the inflow position after leaving the experimental area, subjects were to return to their original position through the side opposite to the goal position of the robot at that point in time to avoid approaching, and subsequently affecting, the robot after it passed through the area. Subjects were also not provided with a description of the evaluated navigation methods or the experiment intention.

The robot detected and measured pedestrian legs with a ROS leg detector [19] and 2D LiDAR point group information. Human detectors, which detect and measure the human positions, based on leg detectors are currently available, but

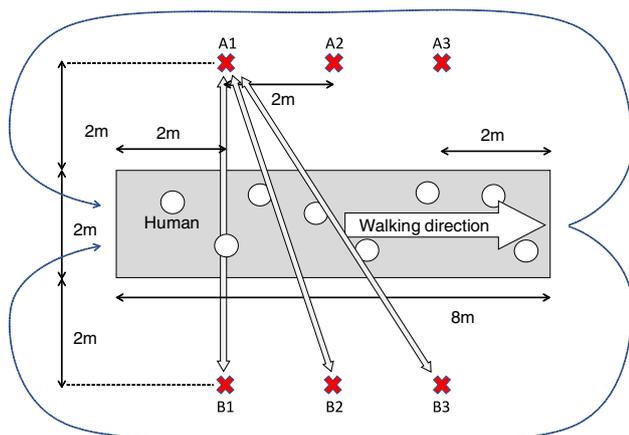


Fig. 10 Experimental environment settings

they were not used in this work because they produce frequent false-negatives. Leg detectors recognize humans by sensing each leg and transmitting the information to the path extractor. Even when humans are walking at a steady velocity, their legs do not necessarily move at the same steady velocity, so this information was filtered and smoothed over time. After smoothing, the estimated time and velocity values were obtained at 25 ms cycles. Accordingly, candidate paths were extracted in 30 ms cycles, and the robot wheel velocity control was implemented in 5 ms cycles.

The process for selecting the route with minimal cost, based on (14), was implemented in 330 ms cycles. The predictor time steps were  $N = 5$ . In the simulation, the path selection cycle was  $\Delta t = 600$  ms, so 3 s of movement was predicted.

### 4.3 Comparison Methods

We refer to our proposed method as navigation with local human-interaction prediction (NLHP). Table 3 shows three alternative methods, used for comparison.

All three of these methods utilize the  $A^*$  method, which is widely used for motion planning.  $A^*_{diff}$  offers the most promising performance, whereas  $A^*_{omni}$  and  $A^*_{omni35}$  were included to investigate the overall performance impact of their varied motion characteristics. A description of the three methods is provided below.

- $A^*_{diff}$ : With the robot radius set as 0.5 m, a route to the goal position was planned. A velocity command was assigned, assuming that the robot runs along the planned trajectory by its differential two-wheel mechanism.
- $A^*_{omni}$ : The same radius and route planning as in  $A^*_{diff}$ . The velocity command was assigned while gradually shifting toward the tangential direction of the generated route, assuming an omnidirectional movement

mechanism. Therefore,  $A^*_{omni}$  exhibits salient behavior differences from  $A^*_{diff}$ , when  $A^*$  provides a path with a direction that is perpendicular to the current robot posture.  $A^*_{diff}$  causes the robot to first stop and turn toward the perpendicular direction (along the path), whereas  $A^*_{omni}$  keeps moving along the route with its current posture and turn gradually toward the perpendicular direction.

- $A^*_{omni35}$ : The route was planned with the radius set as 0.35 m, and the velocity command was assigned as in  $A^*_{omni}$ .

Under  $A^*_{omni}$  and  $A^*_{omni35}$ , frequent collisions occurred with obstacles that were 180 deg outside of the frontal field of view of the robot. Contrarily, in the NLHP method, the robot always moved while facing the direction of the selected path, minimizing the risk of colliding with humans that are right beside or in front of it. Because it reduces the risk of collision,  $A^*_{diff}$  was used as the primary comparison method. Notably,  $A^*_{diff}$  is a direction control method that, due to translational movement, does not move horizontally or diagonally backwards because of its motion model based on a differential two-wheel mechanism. We verified performances of  $A^*_{omni}$  and  $A^*_{omni35}$  as examples of prioritizing time-efficiency over safety in high-density environments.

### 4.4 Evaluation Index

We used the following indices to evaluate and compare the various navigation methods.

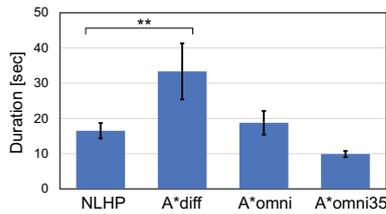
- Duration: the time required to arrive at the goal position [s]
- Human flow efficiency: number of humans who passed a specific point (entrance of the gray region in this measurement) per time [humans/s]
- Collisions: the number of collisions or near-collisions, identified based on pedestrians' emergency avoidance behaviors
- Disturbances: the number of times the flow of pedestrians was disturbed, identified based on pedestrians' changing their movement direction to avoid the robot
- Blame: psychological intimidation experiences when pedestrians were encountered

Because the robot could only measure humans in a 180 deg frontal range, collisions and flow disturbances were identified and measured visually, by us, based on videos taken during the experiment.

Blame, at a given time  $t$ , is quantified by the following equation, in which the human movement direction was added to the evaluation method [7] based on information measured

**Table 3** Robot movement of comparison method

	$A_{\text{diff}}^*$	$A_{\text{omni}}^*$	$A_{\text{omni35}}^*$
Obstacle avoidance radius (m)	0.50	0.50	0.35
Motor command model	Differential drive	Omni wheel	Omni wheel

**Fig. 11** Duration for each of the evaluated methods

and considered by the robot.

$$\begin{aligned}
 B(t) &= \max_i \beta_i \frac{2}{1 + e^{\alpha_i}} \\
 \alpha_i &= \|\Delta t \mathbf{v}_{\text{robot}} - \mathbf{x}_i\| \\
 \beta_i &= \exp \left\{ -\frac{(\phi_i - \theta_i)^2}{2\sigma_{\text{blame}}^2} \right\}
 \end{aligned} \quad (15)$$

Here,  $i$  represents humans within a 1.5 m radius of the robot center, and the blame is evaluated for those humans exhibiting the highest blame levels.  $\alpha_i$  represents the difference between human positions,  $\mathbf{x}_i$ , and the robot position after  $\Delta t$  [s]. Closer distances are represented by smaller denominators, and subsequently larger quotients.  $\beta_i$  is set so that smaller differentials between the direction of human motion,  $\theta_i$ , and the direction of the robot seen by the humans,  $\phi_i$  [rad], result in larger  $\beta_i$  values. The parameter  $\sigma_{\text{blame}} = 0.5$  sets the range of influence, in the form of a Gaussian function. Thus, blame is an indicator that increases when the robot approaches humans or is positioned in the direction of human movement. The blame at each time instant is evaluated as blame per time, which represents an average value for the time a human spends within the 1.5 m radius of the robot.

#### 4.5 Experimental Results

Figure 11 shows the time required for our NLHP method and the other comparative methods to reach the goal position. The error bars indicate a 95% confidence interval and \*\* indicates statistically significant difference at the  $p < 0.01$ . We observe that  $A_{\text{omni35}}^*$ , which did not hesitate to pass through small gaps among humans, reached the goal in the shortest time, and NLHP reached it next. Specifically, the NLHP method provided a 43% time reduction compared with  $A_{\text{diff}}^*$ , the primary comparison method.

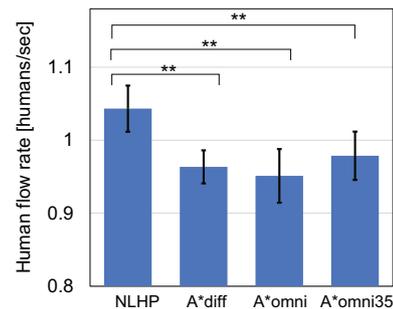
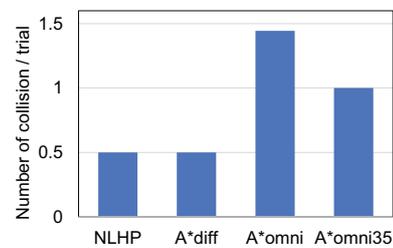
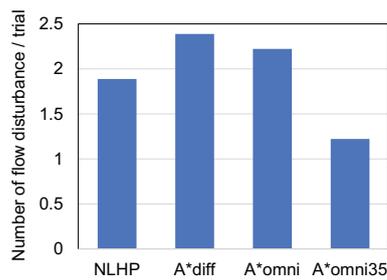
**Fig. 12** Comparison of human flow efficiency during navigation**Fig. 13** Number of collisions produced by the evaluated methods

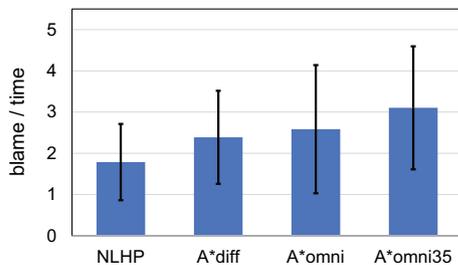
Figure 12 shows the efficiency of human flow, which was found to be approximately 1 humans/sec for all cases. Since the robot affects humans' behaviors only in some limited period in each trial, differences between human flows were small. However, there were statistically significant differences between the proposed method and the comparison methods.

Collisions and disturbances are shown in Fig. 13 and 14, respectively. While the NLHP method and  $A_{\text{diff}}^*$  exhibit similar numbers of collisions,  $A_{\text{omni}}^*$  and  $A_{\text{omni35}}^*$  produced large numbers of both. This is primarily attributed to interference with humans outside of the robot's range of view. Regarding the number of disturbances,  $A_{\text{omni35}}^*$  achieved the least number. It was due to its aggressive and quick strategy, causing very short duration of interaction against pedestrians. Thus, as a total balance including other indices, it cannot be overestimated. Comparison of blame indicated in Fig. 15 shows that the proposed method realized the best performance in reducing psychological influence to humans, though statistical significance was not obtained due to large variation of the blame values in each trial.

All robot behaviors recorded in videos can be accessed from the author's HP [20]. As examples, snapshots of navigations with starting point B1 and destination A3 are shown



**Fig. 14** Number of disturbances produced by the evaluated methods



**Fig. 15** Blame per time for each of the evaluated methods

in Figs. 16, 17, 18 and 19 (resolution was lowered for privacy). Each shot in the sequences is drawn with 3 s interval. First, it can be observed that the density of pedestrians in the setting is quite high and challenging when humans walk in their natural speed without specific consideration to the robot. For cases of  $A^*_{omni}$  and  $A^*_{omni35}$ , the robot took path closer to the direct diagonal line toward the destination. However, it is sometimes human-unconscious, and the robot collided with a pedestrian at the third frame in Fig. 19. With the proposed method, NLHP, the robot was flew downstream of the pedestrian's flow while crossing it. This behavior is caused by the mechanism of extracting human-following direction. In the case of  $A^*_{diff}$  in Fig. 17, the robot was always trying to be safe, but it was too slow and still could not avoid the 'blamed' situation sufficiently.

The comparisons between the five indices with the comparison methods are summarized in Table 4. It was confirmed that NLHP, the proposed method, showed better performance in comparison with  $A^*_{diff}$ , the primary comparison method. On the contrary, as can be seen in duration and number of disturbance comparisons,  $A^*_{omni35}$ , the aggressive and quick strategy, showed a better performance. This means that there can be some room for the proposed method to improve its performance while maintaining the low interference property against humans.

Even though NLHP showed lower blame / time than the other methods in average, there was no statistically significant differences due to large deviations. It can be partly caused by fluctuations in measurement of human velocity because the blame defined in (15) includes estimated direction of human motion. Therefore, decreasing noise in human

motion estimation is expected to increase the reliability of the measurement of blame.

## 4.6 Discussion

The path extraction described in Sect. 3.1 does not utilize human velocity information when setting the robot driving direction because we were unable to measure human velocities with sufficient accuracy in our current experimental system. Additionally, we did not observe significant improvements for methods that did incorporate this information. If the leg and human detectors used in this experiment were improved, and human velocity could be estimated more accurately, we anticipate that using this information would improve the candidate path extractions and subsequently the navigation performance.

Furthermore, in this experimental system, LiDAR provided the robot with a 180 deg frontal range of observation, so contact with humans at the rear and sides could not be avoided. The robot was sufficiently able to stop when humans were detected in the point-blank range, so improving this vision to avoid all collisions will make the robot and our NLHP method viable real-world operation.

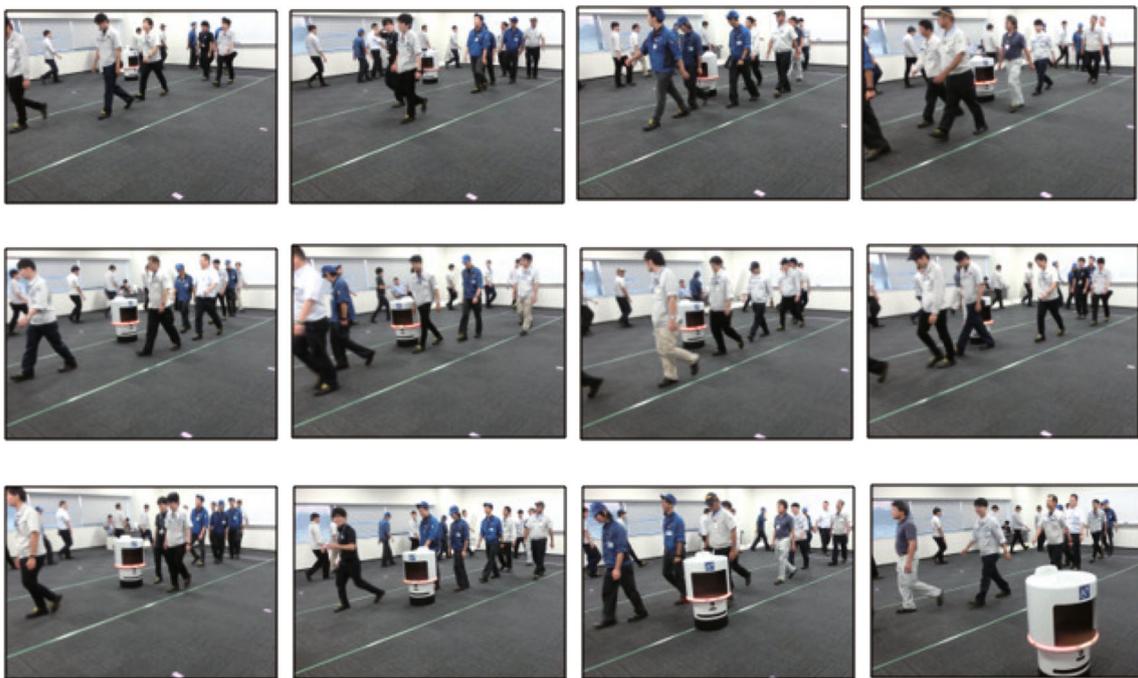
Though the basic behavior of the robot with the proposed method was sufficient in the experiment, there were many cases where the robot could act against humans more appropriately. Parameters for the potential function defined in Sect. 3.1 were mainly tuned based on the robot's behavior in the simulation by human hand. However, it is desired to tune those parameters automatically through both simulations and real-robot experiments for further improving robot navigation performance. Optimization techniques (e.g., [21,22]) can be applied for the automatic tuning of the parameters.

## 5 Conclusion and Future Challenges

In this study, we proposed a method that combines a local planner for route prediction, based on embedded sensor information and a global planner that calculates the shortest distance to the goal, based on map information, to achieve time-efficient robot navigation in high-density crowded environments. First, candidate paths were extracted from the embedded sensor, and the positions what would be reached after approximately 3 s were predicted for each of the path candidates. We used a predictor to forecast the deceleration and the impacts of humans' interaction to the left and right. Combining the time cost of reaching the predicted position and the time cost of reaching the goal, which was obtained by the global planner, we built a route selection system based on global time efficiency. The proposed method was implemented and verified in an environment with a high crowd density of 0.5 humans/m<sup>2</sup> in a reproducible and con-



**Fig. 16** Navigation with NLHP (18.7 s)



**Fig. 17** Navigation with  $A^*_{diff}$  (32.1 s)

trolled way, which helps different methods to be compared against each other in a common condition. These experiments demonstrated that navigation could not only be achieved in this environment, but could be done in a shorter time and with less interference, while maintaining a similar collision risk, compared with established, standard  $A^*$ -based obstacle avoidance methods. The high-density environment employed in our experiment continuously reproduced its high-density with more consistency than other related studies. While these conditions might not occur with this frequency in real-world situations, it provides a common problem setting for evaluating navigation methods in crowded environments.

In the future, the prediction accuracy should be improved. Specifically, the human behavior reproduced in the simulator must be more similar to true human behavior. We expect that, by changing input parameters from those representing individual humans to those representing groups, we can enable prediction of human flows and not just individual travel, thus improving the prediction accuracy. More precise prediction will allow us to realize prediction in longer time span. Though we did not focus on behavior of individual pedestrian in the current implementation, this improvement will improve the local interaction between the robot and a pedestrian such as cooperative collision avoidance [6]. Addi-

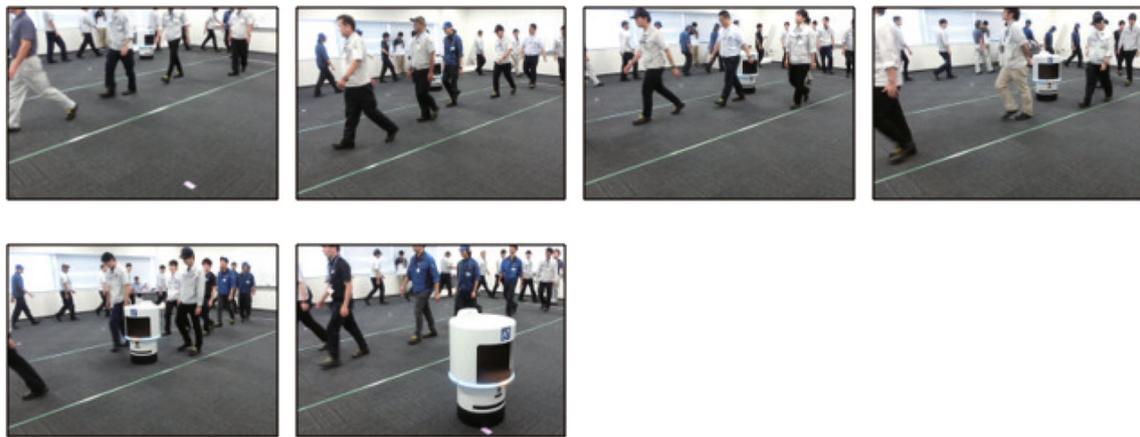


Fig. 18 Navigation with  $A^*_{omni}$  (14.9 s)



Fig. 19 Navigation with  $A^*_{omni35}$  (12.2 s)

**Table 4** Summary of comparisons between the proposed and comparison methods

	NLHP	$A^*_{diff}$	$A^*_{omni}$	$A^*_{omni35}$
Duration	○	×	○	⊙
Human flow	○	×	×	×
# of collision	○	○	×	×
# of disturbance	×	×	×	○
Blame / time	○	△	△	△

‘○ ↔ ×’ and ‘⊙ ↔ ○’ indicate (statistically) significant difference and ‘○ ↔ △’ indicates weak difference

tionally, an option to stop (wait), which is not implemented in the current method, could further improve the performance.

The proposed NLHP method offers the unique advantage of using data collected from learning. This must be extended to data collected in the real world [23] and verified. For example, highly accurate human psychological models that account for individual differences [24] could be introduced while continuing to confirm the significance of these for robot navigation. Finally, while  $A^*$ -based methods are

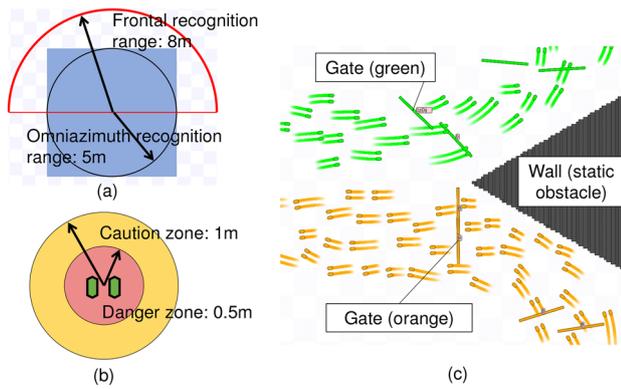
a reasonable standard as a basic comparison, there are other approaches for reciprocal collision avoidance [15,25]. Further comparisons against such navigation methods are to be conducted in our future works.

**Acknowledgements** The authors would like to thank Mr. Shuntaro Nakajima, Mr. Ryota Kaneko and Mr. Takanobu Miyaguchi for supporting implementation of simulations and experiments.

**Declarations**

**Conflict of interest** The authors declare that they have no conflict of interest.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the



**Fig. 20** **a** Pedestrian’s recognition range. **b** Caution zone and danger zone for switching behavior. **c** Pedestrians’ flows generated in simulator by locating gates. A pedestrian has two feet in the simulator but center of each pedestrian is measured as its position by other pedestrians or a robot

permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

### Appendix: Pedestrian Behavior Model in Simulator

In the simulator, each pedestrian is supposed to have a virtual distance sensor that detects distances to other pedestrians and walls in its neighborhood. The detection range of the distance sensor is 8 m in the front (−90 deg to 90 deg around its forward direction) and 5 m for other directions (see Fig. 20a).

A pedestrian’s translational velocity  $v(t)$  [m/s] and rotational angular velocity  $\omega(t)$  [rad/sec] are decided by

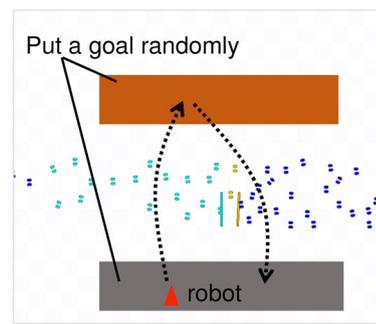
$$v(t) = \max(v_{max}, v(t - \Delta t) + a\Delta t) \tag{16}$$

$$\omega(t) = \alpha\theta_s + \beta\theta_p + \gamma\theta_g, \tag{17}$$

where  $\Delta t$  denotes control cycle.  $v_{max}$  is an individual human parameter which is assigned inherently to each pedestrian by a uniformly random value within range [0.8 1.5] m/s. Acceleration  $a$  is set as 1 m/s<sup>2</sup>.  $\theta_s$ ,  $\theta_p$  and  $\theta_g$  denote angles from the front of human to center of open space, target direction and direction of a neighbor human that has the same target, respectively. The last feedback term  $\gamma\theta_g$  has an effect that humans aiming at the same target behave as a group by sharing the same moving direction. Gain parameters  $\alpha$ ,  $\beta$  and  $\gamma$  corresponding to the angles are set as indicated in Table 5.

**Table 5** Gain parameters of rotation velocity controller

	$\alpha$ (open space)	$\beta$ (target)	$\gamma$ (group)
Normal mode	0.3	0.6	0.3
Moving obstacle in caution zone	0.9	0.3	0.0
Static obstacle in caution zone	1.2	0.1	0.0



**Fig. 21** Pedestrians’ flow in data collection. Gates are placed so that pedestrians in the simulator generate a flow between the robot’s initial position and the destination

There are two zones around a pedestrian that switch its emergent behaviors, named caution zone and of danger zone. Caution zone is defined by radius 1 m and danger zone is defined by 0.5 m (see Fig. 20b). When there is an obstacle in the caution zone, velocities are decided as

$$v(t) = \max\left(v_{min}, \frac{v(t - \Delta t)}{2}\right) \tag{18}$$

$$\omega(t) = \alpha\theta_s + \beta\theta_p + \gamma\theta_g, \tag{19}$$

where  $v_{min}$  is set as 0.2m/s. But in case when there is an open space in its front even when there is an obstacle in the caution zone, velocity is decided as

$$v(t) = \max\left(\frac{v_{max}}{2}, v(t - \Delta t) + a\Delta t\right). \tag{20}$$

If there is an obstacle in danger zone, velocities are decided as

$$v(t) = 0 \tag{21}$$

$$\omega(t) = \text{sign}(\theta_s) \cdot \text{random}(0 \sim 1), \tag{22}$$

where  $\text{sign}()$  denotes sign function and  $\text{random}()$  denotes uniformly random value generator within the indicated range.

In order to guide pedestrians along defined routes, gates are located in the environment. Each pedestrian has its assigned sequence of gates to pass. It search for the next target gate and when it is found, it passes through the gate, guided by the feedback term  $\theta_p$  in (17) and (19). When it has passed the target gate, the next gate is searched. Pedestrians’ flow is generated by locating gates sequentially as shown in Fig. 20c as an example.

In the process of sample collection described in section 3.3, the gates described above are located so that pedestrians generate a chain of flow. Initial positions and destinations of the robot are placed randomly to cross the flow iteratively as shown in Fig. 21. Once the robot reaches its destination,

the goal is newly placed randomly in the opposite side over the flow of pedestrians.

## References

1. Triebel R, Arras K, Alami R, et al (2016) SPENCER: a socially aware service robot for passenger guidance and help in busy airports. In: David SW, Timothy DB (eds) *Field and service robotics*, vol. 11. Springer, Cham, pp 607–622
2. Jafari OH, Mitzel D, Leibe B (2014) Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras. In: *Proceedings of 2014 IEEE international conference on robotics and automation*. IEEE, pp 5636–5643
3. Durrant WH, Bailey T (2006) Simultaneous localization and mapping (SLAM): part I, the essential algorithms. In: Bram V (ed) *IEEE robots and automation magazine*, vol. 13. IEEE, pp 99–110
4. Berg V, Lin M, Manocha D (2008) Reciprocal velocity obstacles for real-time multi-agent navigation. *Robotics and automation*. In: *Proceedings of 2008 IEEE international conference on robotics and automation*. IEEE, pp 1928–1935
5. Kim S, Guy SJ, Liu W, et al (2013) Predicting pedestrian trajectories using velocity-space reasoning. In: Emilio F, Tomas LP, et al (eds) *Algorithmic foundations of robotics X*, vol. 86. Springer, Berlin, Heidelberg, pp 609–623
6. Trautman P, Krause A (2010) Unfreezing the robot: navigation in dense, interacting crowds. In: *Proceedings of 2010 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, pp 797–803
7. Mehta G, Ferrer G, Olson E (2016) Autonomous navigation in dynamic social environments using multi-policy decision making. In: *Proceedings of 2016 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, pp 1190–1197
8. Cunningham AG, Galceran E, Eustice RM, et al. (2015) MPDM: multipolicy decision-making in dynamic, uncertain environments for autonomous driving. In: *Proceeding of 2015 IEEE international conference robots and automation*. IEEE, pp 1670–1677
9. Tamura Y, Terada Y, Yamashita A, Asama H (2013) Modelling behaviour patterns of pedestrians for mobile robot trajectory generation. *Int J Adv Robotic Syst* 10(8): 1–11
10. Abbeel P, Andrew YN (2004) Apprenticeship learning via inverse reinforcement learning. In: *Proceedings of 21st international conference on machine learning*. ACM, pp 1–8
11. Kretschmar H, Spies M, Sprunk C, Burgard W (2016) Socially compliant mobile robot navigation via inverse reinforcement learning. *Int J Robot Res* 35(11):1289–1307
12. Mnih V, Kavukcuoglu K, Silver D, et al (2015) Human-level control through deep reinforcement learning. *Nature* 518(7540): 529–533
13. Chen Y, Liu M, Everett M, et al (2017) Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning. In: *Proceedings of 2017 IEEE international conference on robotics and automation*. IEEE, pp 285–292
14. Trautman P, et al (2015) Robot navigation in dense human crowds: statistical models and experimental studies of human-robot cooperation. *Int J Robot Res* 34(3): 335–356 (John MH (ed))
15. Luo Y, Cai P, Bera A, Hsu D, Lee WS, Manocha D (2018) PORCA: modeling and planning for autonomous driving among many pedestrians. *IEEE Robot Autom Lett (RA-L)* 3(4): 3418–3425
16. Bando M, Hasebe K, Nakayama A, et al (1994) Structure stability of congestion in traffic dynamics. *Jpn J Indus Appl Math* 11(2): 203–223 (Yamaguchi M et al (eds)), Springer-Verlag
17. Tordeux A, Schadschneider A (2016) A stochastic optimal velocity model for pedestrian flow. In: Wyrzykowski R, Deelman E, Dongarra J, Karczewski K, Kitowski J, Wiatr K (eds) *Parallel processing and applied mathematics. Lecture notes in computer science*, vol 9574. Springer
18. [http://wiki.ros.org/simulator\\_stage](http://wiki.ros.org/simulator_stage)
19. [http://wiki.ros.org/leg\\_detector](http://wiki.ros.org/leg_detector)
20. <https://sensor.eng.shizuoka.ac.jp/~koba/crowdednavigation.html>
21. Botzheim J, Toda Y, Kubota N (2012) Bacterial memetic algorithm for offline path planning of mobile robots. *Mimetic Comput* 4:73–86
22. Csík Á, Botzheim J, Balázs J, Csoknyai T, Hontvári JL (2012) Energy and cost optimal design for the reconstruction of residential building envelopes by bacterial memetic algorithms. In: *Proceedings of the 6th international conference on soft computing and intelligent systems, and the 13th international symposium on advanced intelligence systems*, pp. 1226–1231
23. Morales Y, Akai N, Murase H (2018) Personal mobility vehicle autonomous navigation through pedestrian flow: a data driven approach for parameter extraction. In: *2018 IEEE/RSJ international conference on intelligent robots and systems (IROS 2018)*
24. Bera A, Randhavane T, Prinja R, Manocha D (2017) SocioSense: robot navigation amongst pedestrians with social and psychological constraints. In: *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. Vancouver, BC, pp 7018–7025. <https://doi.org/10.1109/IROS.2017.8206628>
25. van den Berg J, Guy Stephen J, Ming L, Dinesh M (2011) Reciprocal n-body collision avoidance, robotics research. In: Pradalier C, Siegwart R, Hirzinger G (eds) *The 14th international symposium ISRR*, Springer tracts in advanced robotics, vol. 70. Springer-Verlag, pp 3–19
26. John D, Elad H, Yoram S (2011) Adaptive subgradient methods for online learning and stochastic optimization. *J Mach Learn Res* 12:2121–2159

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.