


RESEARCH ARTICLE

# A novel Human-Aware Navigation algorithm based on behavioral intention cognition

Jiahao Li, Fuhai Zhang  and Yili Fu

The State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China

**Corresponding author:** Fuhai Zhang; Email: [zfhhit@hit.edu.cn](mailto:zfhhit@hit.edu.cn)

**Received:** 18 August 2023; **Revised:** 3 November 2023; **Accepted:** 10 December 2023;

**First published online:** 4 January 2024

**Keywords:** mobile robot; social behavioral model; navigation; cost map

## Abstract

In order to ensure safe and comfortable human–robot navigation in close proximity, it is imperative for robots to possess the capability to understand human behavioral intention. With this objective in mind, this paper introduces a Human-Aware Navigation (HAN) algorithm. The HAN system combines insights from studies on human detection, social behavioral model, and behavior prediction, all while incorporating social distance considerations. This information is integrated into a layer dedicated to human behavior intention cognition, achieved through the fusion of data from laser radar and Kinect sensors, employing Gaussian functions to account for individual private space and movement trend. To cater to the mapping requirements of the HAN system, we have reduced the computational complexity associated with traditional multilayer cost map by implementing a “first-come, first-served” expansion method. Subsequently, we have enhanced the trajectory optimization equation by incorporating an improved dynamic triangle window method that integrates human behavior intention cognition, leading to the determination of an appropriate trajectory for the robot. Finally, experimental evaluations have been conducted to assess and validate the efficacy of the human behavior intention cognition and the HAN system. The results clearly demonstrate that the HAN system outperforms the traditional Dynamic Window Approach algorithm in ensuring the safety and comfort of humans in human–robot coexistence environments.

## 1. Introduction

As robotics technology advances, mobile robots are no longer confined to public spaces and workplaces but are increasingly finding their way into private homes. The use of mobile service robots in social settings such as airports [1], museums [2], and offices necessitates a heightened emphasis on their ability to navigate safely and carry out tasks in proximity to humans [3, 4]. Current mobile robots demonstrate the capability to navigate safely within their operational environments, effectively avoiding both static and dynamic obstacles [5]. However, from the human perspective, there is a preference for maintaining a certain distance from robots to ensure safety and comfort within the environment [6, 7]. Thus, treating humans merely as dynamic obstacles to be avoided is insufficient. In light of these considerations, it is essential to address the unique challenges associated with human–robot interaction in social environments, where the safety and comfort of humans take precedence.

Human-Aware Navigation (HAN), in brief, is a study focused on the collaborative movement of robots and humans, considering their interaction as a significant social factor. It entails the examination of the various modes of motion when these entities are in close proximity. For instance, common scenarios encountered in the daily activities of mobile robots involve situations where humans and robots cross paths. When people and robots come face to face in a corridor, they engage in a spatial negotiation to navigate past each other successfully. This often involves determining the sequence in which they pass.

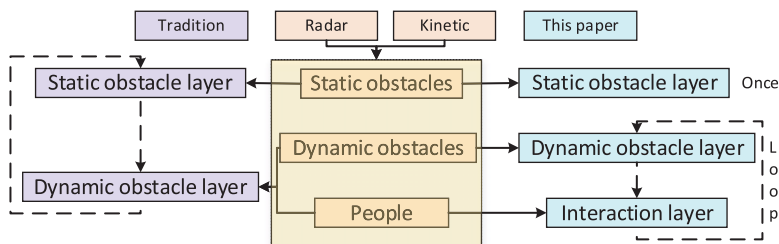
HAN is primarily divided into two core components: the understanding of human behavioral intention and the exploration of robot navigation algorithms. The integration of these two facets yields robotic behaviors that are designed to enhance human comfort and safety.

In the realm of human behavior intention recognition, two predominant approaches are commonly employed: the model method and the dynamic obstacle method. The model method primarily characterizes various human states within an environment by establishing a series of models, including two main models. The first model pertains to the man–robot collision cost model. Melo et al. [8] introduced an adaptive model in which the parameters of personal and group space’s cost functions adjust according to the arrangement of constraints. Sisbot et al. [9] incorporated considerations of safety, comfort, and hidden areas into their developed cost model. The second model is the social force model [10], which influences robot behavior through interactions involving humans, robots, and obstacles. However, a primary limitation of the model method is that it treats individuals within the workspace as static obstacles rather than dynamic ones [11, 12]. Consequently, robots are unable to provide dynamic responses to adapt to changing environments, thus impacting the human experience in human–robot coexistence environments.

This issue has garnered attention in the context of the dynamic obstacle method. This approach guides robots to make dynamic responses by enhancing both global path planning methods (e.g., A\* [13], Dijkstra [14]) and local path planning methods (such as the velocity obstacle method [15] and the dynamic window method [16]). For instance, Goller et al. [17] combined the A\* algorithm with a reactive local planning algorithm in densely populated supermarket environments to plan safe paths for robots. Nevertheless, when conflicts arise between human presence and the global path, robots may only opt for a waiting approach, which can adversely affect the comfort of individuals in the environment.

Another crucial aspect of HAN lies in the domain of navigation algorithms, which can be broadly categorized into global path planning [18–20] and local path planning [21–23] within the field of mobile robot navigation. Global path planning primarily deals with devising an overarching path strategy from the robot’s current location to its target destination, making the inclusion of social factors less impactful. In this study, our focus is on optimizing local path planning algorithms by incorporating social considerations. Among the mainstream local path planning algorithms, the Timed Elastic Band (TEB) algorithm [24, 25] stands out. This method is optimized based on the robot’s state, velocity, and acceleration constraints, as well as environmental obstacles. It exhibits strong real-time performance; however, it is most suitable for vehicle-based robots and may encounter challenges related to local optima. The classical artificial potential field method introduced by Khatib [26] mathematically articulates the relationship between robots and obstacles but is also prone to local optima. On the other hand, random sampling-based path planning algorithms, such as Rapidly-exploring Random Trees (RRT) [27], significantly enhance real-time planning and offer computational efficiency. However, due to limitations in the number of iterations, the path’s validity and optimality are not always guaranteed. The Dynamic Window Approach (DWA) [16] narrows down the robot’s velocity search space to a set of attainable velocity pairs, from which the optimal trajectory is selected [28]. Each velocity pair represents a trajectory with a specific value. Recent work by R. Yuan et al. [29] introduced a novel Q-learning approach for navigation in dynamic environments that incorporates a dynamic window, yielding promising results. Nevertheless, the existing DWA algorithm is primarily designed to avoid robot collisions and does not explicitly consider human factors.

Based on the analysis of the literature mentioned above, the utilization of model-based approaches, which treat individuals as static entities, comes with certain limitations. Besides, exclusively regarding individuals as dynamic obstacles neglects the distinction between humans and inanimate objects. The method proposed in this paper employs laser radar and RGB-D camera to detect human bodies, accurately pinpoint the positions of dynamic individuals, and predict their behaviors. By introducing the concept of social distance, this method establishes a model that aligns more closely with human private space. Concerning navigation, in comparison to the aforementioned literature, our proposed approach incorporates social factors. Unlike the traditional method of treating people as obstacles, our approach



**Figure 1.** Improved multilayer cost map sequence configuration.

in path planning takes into account the distances between the robot and human, resulting in a more comfortable and safer human–robot navigation experience.

The remainder of this paper is structured as follows. In Section 2, we establish a multilayer cost map and introduce the social behavioral model. Section 3 introduces the HAN algorithm for mobile robots designed to enhance human comfort. Subsequently, in Section 4, we conduct experiments to validate the effectiveness and performance of the multilayer cost map, behavioral intent cognition, and the navigation algorithm. Finally, we discuss and conclude this paper by evaluating the performance of the HAN algorithm.

To enhance human comfort and ensure safety and efficiency in human–robot coexistence environments, our paper makes the following contributions:

1. We propose a social behavioral model to discern human behavioral intention;
2. We present a HAN algorithm for mobile robots to derive optimized paths that prioritize human comfort.

## 2. Behavioral intention cognition

HAN relies on efficient map processing and novel human processing methods. Consequently, this paper enhances the matching and computational efficiency of traditional multilayer cost maps. Building upon this foundation, we introduce the social behavioral model, which integrates data from laser radar and Kinect sensors, utilizing sociological distance metrics and Gaussian functions to depict collision costs between robots and humans within the environment. The social behavioral model is incorporated into the improved multilayer cost map, resulting in the creation of a new multilayer cost map that incorporates human behavioral intention cognition.

### 2.1. Improved multilayer cost map

Mobile robots often rely on maps with high accuracy and real-time performance when executing tasks. The traditional multilayer cost map exhibits two primary shortcomings in human behavioral intention cognition. First, when an obstacle is positioned at the edge of the laser radar detection range or extends beyond this range, utilizing a linear algorithm can result in a delayed removal of obstacle information from the previous moment, leading to significant errors in map information and path planning discrepancies. Second, the traditional cost map update process employs a priority queue method, updating all layers within a single cycle, consequently diminishing computational efficiency.

Given that the static obstacle layer is processed only once, this paper employs a preprocessing approach to retain crucial obstacle information within the map and maintain it in the static layer. This information remains unaltered throughout the update process and serves as the foundational data in the cost map as shown in Fig. 1.

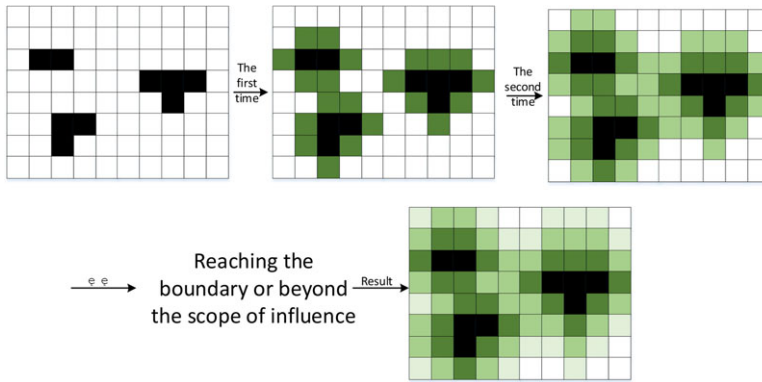


Figure 2. Schematic diagram of expansion mode.

In indoor mobile robot working environments, the map frequently contains an abundance of static obstacle data. Traditional methods involving repeated expansions may entail multiple processes for a single grid, substantially increasing algorithmic complexity and severely impacting the initial map update efficiency. Consequently, this paper adopts the “first-come, first-served” expansion method. In essence, this approach takes into account the position of each obstacle within the map data, expanding to each grid point within the map. Only when a grid is processed for the first time, it can acquire its cost value, wherein the cost value of each grid is solely influenced by the nearest obstacle point to that grid. Subsequently, the affected grid points are iteratively processed, updating the affected area sequentially until the iteration reaches the map’s boundary or extends beyond the influence radius of the obstacle point as shown in Fig. 2.

The algorithm takes the distance as the index value and arranges all grid points in turn to form a quadtree structure:

$$P_n = \{P_i | (x_i, y_i, h_i)\} \tag{1}$$

where  $P_n$  represents the  $n$  grid points,  $P_i$  represents the grid point  $i$ ,  $x_i, y_i$  are the coordinates of the grid point  $i$ , and  $h_i$  is the depth of the quadtree structure.

Taking the depth  $h_i$  of the quadtree structure as the index value of the cost value of the layer, the corresponding cost value constructor is formed:

$$COST_{P_i} = \begin{cases} COST_{OBSTACLE} \cdot \exp(-h_i) & h_i \leq h_{threshold} \\ COST_{FREE} & ELSE \end{cases} \tag{2}$$

where  $COST_{P_i}$  is the cost value of the grid point  $i$ ,  $COST_{OBSTACLE}$  is the cost value of the obstacle,  $COST_{FREE}$  is the cost value of the non-obstacle, and  $h_{threshold}$  is the threshold of the depth of the quadtree structure.

When an obstacle node is positioned within the quadtree to constitute a parent node, four identical child nodes are created within each parent node to store the child nodes surrounding the parent node’s grid. These corresponding nodes are then invoked to verify whether the node is in an available state, typically referred to as an “unprocessed state.” As nodes within layer  $h_i$  undergo processing, the subnodes within layer  $h_{i+1}$  are initialized and transition into a state ready for processing. For nodes that have already been processed, the traversal process will directly bypass the node and all its subnodes, subsequently eliminating branches within the quadtree. Consequently, in the iterative process, the computational time complexity of the traditional cost value method remains stable at:

$$COST_{Complexity} = O(r^n) \tag{3}$$

where  $r$  is the number of child nodes in the iteration process of cost value.

**Table I.** *Equidistant expansion method.*

---

**Algorithm: Equidistant expansion method.**

---

**Input:** Initial\_map**Output:** Static\_obstacle\_layer

1. Initialize map parameters;
  2. Generate Static\_obstacle\_layer;
  3. Analysis of obstacle data  $vector_{process}$  in source map;
  4. **for**  $i < vector_{process}.size$  **do**
  5.     Process  $vector_{process}[i]$ ;
  6.     Initialize child nodes  $\rightarrow vector_{process}$ ;
  7. **end**
  8. Update Static\_obstacle\_layer;
  9. Return Static\_obstacle\_layer;
- 

After using the “first-come, first-served” strategy, the quadtree is pruned, and the computational time complexity is stable at:

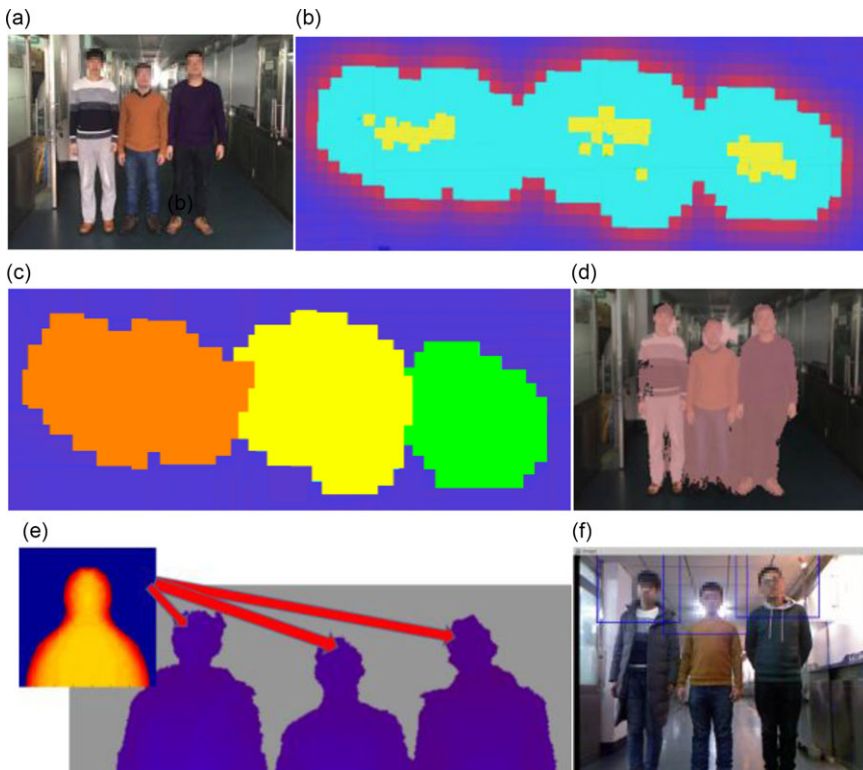
$$COST'_{Complexity} = O(n^2) \quad (4)$$

It is evident that as the amount of information regarding static obstacles in the map increases, the number of grid points, denoted as “ $n$ ,” occupied by these static obstacles also rises. The algorithm presented in this paper offers distinct advantages over traditional methods. When the robot enters an indoor environment, the information related to static obstacles becomes superimposed, resulting in an increased overlap between obstacle grids and a consequent reduction in the number of processed grids. This contributes to a slight improvement in the efficiency of the algorithm proposed in this paper. The pseudocode for this method is provided in Table I.

## 2.2. Human detection

The primary task of human behavior intention cognition is the detection of human positions. Among the sensors commonly utilized, laser radar excels in terms of high accuracy and robustness when conducting large-scale measurements on a 2D horizontal plane. However, it may not provide a complete representation of the human body. In indoor environments, obstacles often possess geometric characteristics similar to human legs, leading to potential misidentification when solely relying on laser radar for human leg detection. Kinect, on the other hand, offers three-dimensional data, allowing for the representation of stereo images and depth information. Nevertheless, Kinect is susceptible to interference from lighting conditions. To address this, we combine laser radar with Kinect to avoid the error detection of human detection.

To begin, 2D laser radar is employed for the initial detection of human legs. This is achieved by defining specific features of human legs, as outlined in [18], including attributes such as data dispersion, geometry, and movement characteristics. The segmentation domains, derived from the point cloud data generated by the laser radar, are subjected to classification. Using the Adaboost algorithm, an adaptive boosting algorithm that modifies the weight distribution at each round of the training process, it is determined whether a given segment corresponds to a human leg.



**Figure 3.** Human detection procedure. (a) Original image. (b) 2D projection. (c) cluster segmentation. (d) Point cloud marking. (e) Upper body detection. (f) Classification result.

A weak classifier is designed using a single value feature  $g_k$  as the following forms:

$$i_k(f) = \begin{cases} +1 & q_k g_k(f) < q_k \lambda_k \\ -1 & \text{otherwise} \end{cases} \quad (5)$$

where  $\lambda_k$  is the threshold value,  $q_k$  is  $+1$  or  $-1$ , and represents the direction of the inequality. In each round of training, the value of  $\lambda_k$  and  $q_k$  are updated, and the error classification in the weighted training sample is minimized.

After detecting the human leg, the upper body can be identified through the analysis of RGB-D data obtained from the Kinect sensor as shown in Fig. 3. This process involves several key steps. To begin, the acquired point cloud data is meticulously filtered to ensure a high level of accuracy. Subsequently, the RANSAC (Random Sample Consensus) algorithm is applied to ascertain the horizontal plane. This plane serves as a crucial reference for classifying the point cloud into four distinct categories based on height information: ground, objects, transition spaces, and high-altitude building structures. The remaining point cloud, post-removal of ground and building structures, proves to be highly suitable for region of interest (ROI) extraction. In this regard, the ROI extraction method entails projecting the 3D point cloud onto a horizontal plane to generate a two-dimensional histogram. A Gaussian filter is then utilized to refine the histogram, followed by threshold processing. For efficient region segmentation, the Quick-Shift fast mode search algorithm is employed.

Using the RGB-D information as input, the robotic system runs the video frame and fully integrates the methods covered in this article into ROS at the efficiency of 18fps in step (a) to step (f). In order to verify the accuracy of Kinect’s upper body detection, an experiment was conducted within a corridor scenario where three individuals freely traversed the area, with the detected individuals being clearly marked. The experimental results of Kinect’s upper body detection showed that in the field of vision of



**Table II.** Private space.

Parameter	Value/m
Private area	0–0.45
Personal area	0.45–1.2
Social area	1.2–3.6
Public area	>3.6

**Figure 4.** Human upper body detection.

Kinect, with the movement of pedestrians, the robot can recognize the upper body of the human body as shown in Fig. 4.

After obtaining the laser radar detection results  $h_l$  and the Kinect detection results  $h_k$ , the information from laser radar and Kinect is fused to complete the human tracking system based on GNN algorithm. Both  $h_l = (x_l, y_l)$  and  $h_k = (x_k, y_k)$  are taken as the observation values of particle filter so that the likelihood  $L$  in the particle filter can be calculated as follows:

$$L = \beta_l \times L_{laser} + \beta_b \times L_{rgb} \quad (6)$$

where  $\beta_l$  represents the confidence of laser-based human leg detection and  $\beta_b$  represents the confidence of camera-based human detection algorithm.

$$L_{laser} = \exp(-(\sqrt{(x_l - x_t)^2 + (y_l - y_t)^2})) \quad (7)$$

$$L_{rgb} = \exp(-(\sqrt{(x_k - x_t)^2 + (y_k - y_t)^2})) \quad (8)$$

where  $x_t$  represents the  $x$  coordinate of human position and  $y_t$  represents the  $y$  coordinate. According to the tracking information of human body, the posture of human is extracted.

Assume that there are  $N$  people around the robot and the state of the  $i$ th people is  $p_i = (x_i^p, y_i^p, \theta_i^p, v_i^p)$ , in which  $(x_i^p, y_i^p)$  represents the human location,  $\theta_i^p$  represents the human direction, and  $v_i^p$  represents the human speed. Then, the constant velocity model is used to estimate the human trajectory that is used to predict the human motivation and to describe the possible trajectory.

### 2.3. Social behavioral model

Once human position detection has been successfully accomplished, the subsequent objective is to enhance human comfort, adhere to social norms, and enable various human behaviors within the human–robot coexistence environment. This paper introduces a social behavioral model to address the requirements for human comfort in such a context. The spatial relationship between humans and robots, which signifies the social distance that individuals maintain in diverse social situations, stands as a primary determinant of human comfort. The concept of private space was initially posited by Hall in 1966 [30], considering the influence of interpersonal distances on comfort. This concept categorizes the space surrounding individuals into four distinct areas: private space, personal space, social space, and public space, as illustrated in Table II.

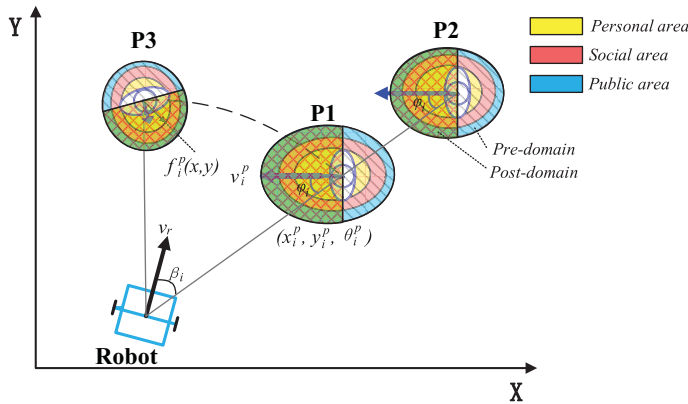


Figure 5. Social behavioral model in man-robot coexistence environment.

The limitation of Hall’s classification standards is that they maintain a constant definition of private space, without accounting for the relative pose and motion direction between the robot and individuals. Therefore, this paper proposes a new concept called collision probability that can be represented with collision probability function  $f_i^{cp}$ , which describes the relationship between human body  $p_i$  and the robot. This paper combines the collision probability and private space model, as shown in Fig. 5.  $p_i = (x_i^p, y_i^p, \theta_i^p, v_i^p)$  represents the state of human, and  $r = (x_r, y_r, \theta_r, v_r)$  represents the state of robot, and then the collision probability function is written as:

$$f_i^{cp} = 0.5 + 1 / \left( 2 + \exp \left( \gamma \frac{v_{rp}}{d_{rp}} \right) \right) \tag{9}$$

$$d_{rp} = \sqrt{(x_i^p - x_r)^2 + (y_i^p - y_r)^2} \tag{10}$$

$$v_{rp} = v_i^p \cos(\varphi_i) + v_r \cos(\beta_i) \tag{11}$$

where  $f_i^{cp}$  denotes the probability of collision between human and robot, with a range of values 0.5~1, and  $\gamma$  is a constant default value. Hence, as  $v_{rp}$  increases and  $d_{rp}$  decreases, the collision probability  $f_i^{cp}$  increases.

In order to create a private space model, a 2D Gaussian function based on collision probability is developed in this section. The function  $f_i^{cp}(x, y)$  is used to indicate the social behavioral model that has the maximum value at the center of human  $(x_i, y_i)$  and decreases as the distance increases gradually from  $(x_i, y_i)$ . The private space is represented as:

$$f_i^p(x, y) = A^p \exp \left( - \left( \left( \frac{d \cos(\theta - \theta_i^p)}{\sqrt{2}\sigma_i^{px}} \right)^2 + \left( \frac{d \sin(\theta - \theta_i^p)}{\sqrt{2}\sigma_i^{py}} \right)^2 \right) \right) \tag{12}$$

Here,  $d$  and  $\theta$  are represented as:

$$d = \sqrt{(x - x_i^p)^2 + (y - y_i^p)^2} \tag{13}$$

$$\theta = \text{atan2}((y - y_i^p), (x - x_i^p)) \tag{14}$$

where  $(x_i^p, y_i^p)$  represents the central position of human  $p_i$ ,  $\theta_i^p$  represents the direction of human,  $\sigma_i^{px}$  and  $\sigma_i^{py}$  are the standard deviations of the Gaussian distribution, and  $A_p$  represents the average value.



**Table III.** Social behavioral model parameter.

Parameter	Value
Average $A_p$	255
Standard deviation $\sigma_0^{px}$	0.25
Standard deviation $\sigma_0^{py}$	0.45
Collision probability factor $\gamma$	-3.0
Speed factor $f_v$	0.8
Pre-domain effect factor $f_{front}$	0.2
Forbidden radius $r_0$	0.25

**Table IV.** The results of parameter calculation in four states.

Parameter	$f_i^{cp}$	$\sigma_i^{py}$	$\sigma_i^{px}$ front	$\sigma_i^{py}$ back
1	0.8333	0.2083	0.4500	0.3750
2	0.9498	0.2375	0.6839	0.4274
3	0.9045	0.2261	0.6512	0.4070
4	0.9498	0.2375	0.8548	0.4274

As shown in Eq. (12), the size and shape of private space are related to the basic parameters  $A^p$ ,  $\sigma_i^{px}$ , and  $\sigma_i^{py}$ . In this paper, the human state information and collision probability will be incorporated into the social behavioral model to affect  $\sigma_i^{px}$  and  $\sigma_i^{py}$ . The space around people can be divided into pre-domain and post-domain. In the pre-domain, the new velocity effect factor  $f_v$  and the pre-domain effect factor  $f_{front}$  are introduced, and  $\sigma_i^{px}$  and  $\sigma_i^{py}$  are calculated as follows:

$$\sigma_i^{py} = \begin{cases} (1 + v_i^p f_v + f_{front}) f_i^{cp} \sigma_0^{py} & \text{front} \\ f_i^{cp} \sigma_0^{py} & \text{back} \end{cases} \tag{15}$$

$$\sigma_i^{px} = f_i^{cp} \sigma_0^{px} \tag{16}$$

Under the framework of HAN, the parameters of the social behavioral model are shown in Table III.

To validate the accuracy of the social behavioral model, we conducted a simulation under specific initial conditions. When the human states are defined as  $p_1 = (0, 2, -\frac{\pi}{2}, 0)$ ,  $p_2 = (0, 3.5, -\frac{\pi}{2}, 0)$ , and  $p_3 = (0, 2, -\frac{\pi}{2}, 0.5)$ , respectively, the corresponding robot state is consistently represented as  $r = (0, 0, 0, 0)$ ; when the human state is  $p_4 = (0, 2, -\frac{\pi}{2}, 0.5)$ , the robot state shifts to  $r = (0, 0, 0, 0.5)$ . According to the parameters in Table IV, the social behavioral model of the above four states can be obtained, respectively, as shown in Fig. 6. Table IV lists the calculated collision probability  $f_i^{cp}$  and standard deviations  $\sigma_0^{px}$  and  $\sigma_0^{py}$ . Notably, our observations reveal that as relative velocity increases and relative distance decreases, the model’s range expands accordingly.

### 2.4. Behavior prediction

Behavior prediction has a profound impact on the performance of HAN in the context of mobile robots. Among the elements influencing behavior intention, the human movement trajectory stands out as the most crucial. For short-term predictions, both uniform velocity models and constant acceleration models are widely employed due to their ability to ensure adequate accuracy. However, it is important to acknowledge that observational data concerning the object’s movement trajectory is often subject to noise and errors. To address this challenge, we introduce the use of a particle filter into the trajectory prediction model, aimed at mitigating the impact of observation noise. The motion prediction model we employ consists of both a uniform velocity model and a particle filter, as illustrated in Fig. 7.

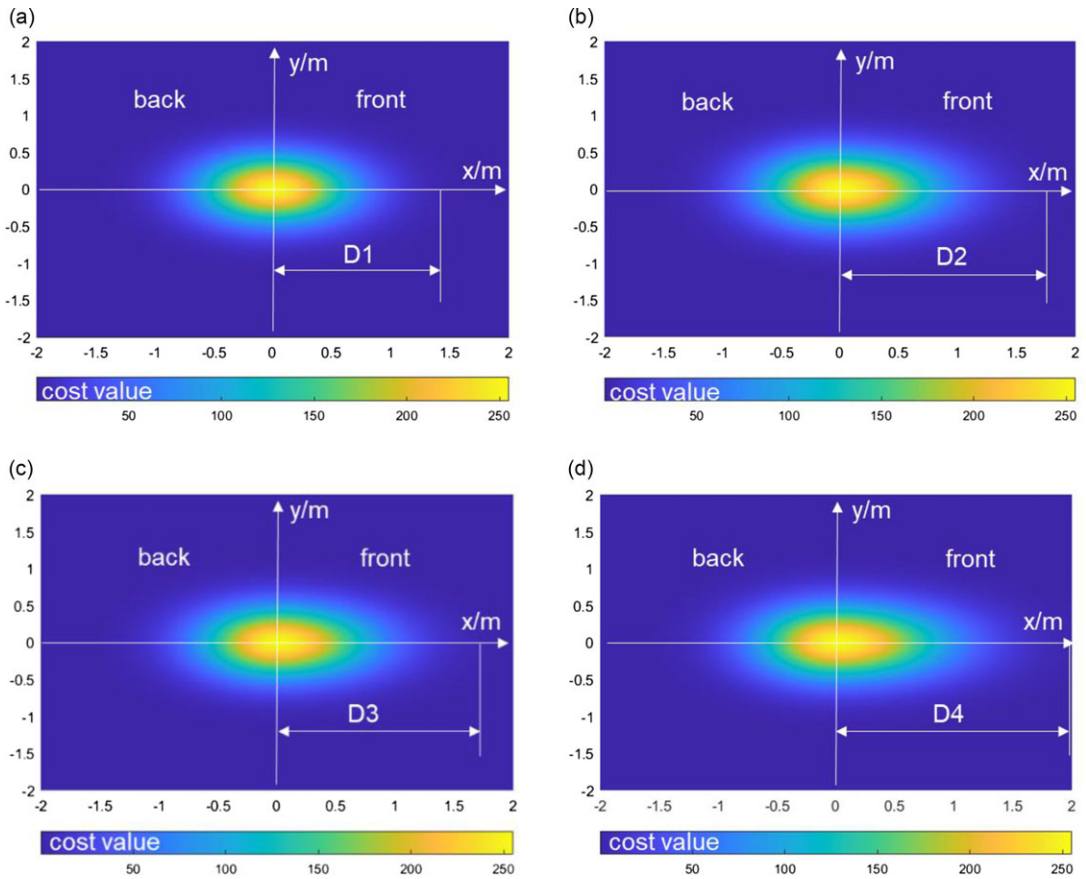


Figure 6. Collision cost. (a) State 1. (b) State 2. (c) State 3. (d) State 4.

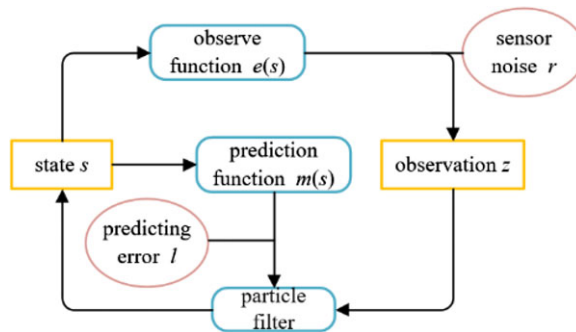


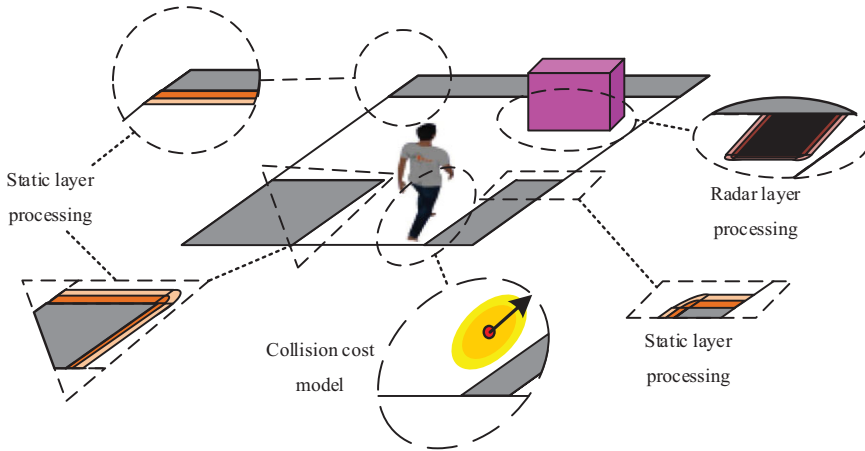
Figure 7. Flow chart of behavior prediction.

Assuming that the robot runs in a spacious environment with fewer obstacles, the uniform velocity model is reasonable. State  $s_k$  and time  $t_k$  of the robot can be predicted using the known state at  $t_{k-1}$ :

$$s_k = m(s_{k-1}) + l_k \tag{17}$$

where  $l_k$  is prediction error and  $m$  is prediction equation. The state  $s$  of robot in 2D environment is

$$s = \begin{bmatrix} p \\ v \end{bmatrix} \tag{18}$$



**Figure 8.** A new multilayer cost map with behavioral intention cognition.

where  $p = [x \ y \ \theta]^T$  is the state matrix of robot, including robot’s position and posture and  $v = [v_x \ v_y \ w]^T$  is a velocity matrix, and prediction equation  $m$  is

$$m \left( \begin{bmatrix} p \\ v \end{bmatrix} \right) = \begin{bmatrix} p + v \cdot \Delta t \\ v \end{bmatrix} \tag{19}$$

In addition, the robot observes the motion state of the person through the observation function  $e$  and the observation value  $z_k$  of robot depends on the real state  $s_k$ . There will inevitably be sensor noise  $r_k$ . Thus, the observation function can be expressed as:

$$z_k = e(s_k) + r_k \tag{20}$$

In general, the visible information is position  $p$ , which means that  $z_k = [x \ y \ \theta]^T$  and  $e$  can be expressed as:

$$e \left( \begin{bmatrix} p \\ v \end{bmatrix} \right) = p \tag{21}$$

In order to deal with prediction error  $l$  and sensor noise  $r$ , since the distribution of the two does not have a fixed form, such as a Gaussian distribution, a particle filter is preferred.

**2.5. Behavioral intention cognition layer**

We utilize a multilayer dynamic cost map, composed of multiple dynamic layers, to establish the behavioral intention cognition layer and represent dynamic social constraints associated with temporal factors, as illustrated in Fig. 8. Each layer is a two-dimensional grid map. Once all layers are integrated, the attributes of each grid represent the cost value to be assigned when encountering navigation constraints.

The behavioral intent cognition layer provides a carrier for the social behavioral model and contains the cost value related to some social constraints. This layer evolves over time, giving rise to a sequence of dynamic layers. Each dynamic layer represents the prediction of human trajectories from the time step  $i$  to  $i + 1$  and fills cost value in the current layer. The main cost map can be formed by superimposing the  $i$ th multilayer dynamic cost map of this section on the multilayer cost map mentioned in Section 2.1 as shown in Table V. As a result, in the process of path planning by the robot, the estimated path cost value at time  $t_i$  can be obtained from the main cost map. If  $t_i$  exceeds the time that can be expressed by the multilayer dynamic cost map, the cost value will only be obtained from the original multilevel

**Table V.** Formation of the main cost map.

---

**Algorithm: Dynamic Layered Costmap**

---

**Input:** PeoplePrediction {P}, master\_map, applitude, variance\_x, variance\_y;

**Output:** Dynamic layered Costmap;

```

1 Initialize the time_resolution, time_steps, dynamic_layers_plugin;
2 for count k < time_steps do
3     Initialize (onemap, k);
4     timed_costmap.push_back(onemap);
5 end
6 Update Map;
7 Initialize intermediate variavle k;
8 for k < time_steps do
9     gauss=calcGussian(x ,y ,P.x, P.y, applitude, variance_x,
10    variance_y, angle);
11    Identify pre-domain and post-domain;
12    Update Map;
13    MarkHumanInCostMap(P.x, P.y, timed_costmap);
14 end
15 Coordinate system joint processing;
16 Return Dynamic layered Costmap_joint calibration

```

---

cost map. Therefore, the estimated total length of time should be as long as possible to deal with more complex situations.

The superposition map can effectively represent the time-related navigation task. Furthermore, the resolution and prediction time can be adjusted according to the complexity of the specific environment to optimize storage space usage.

At this stage, the robot can perceive pedestrian space and motion trends based on the fused multilayer cost map and navigate with human information.

### 3. Human-Aware Navigation

#### 3.1 Navigation velocity sampling

The mobile robot employed in this paper is a two-wheel differential drive platform, which allows it to alter its azimuth angle by independently adjusting the speed of its left and right wheels.

In an ideal condition, the next state of the robot  $s_{t+1} = (x', y', \theta')^T$  at  $t + 1$  moment is related to the last state  $s_t = (x, y, \theta)^T$  and the control signal  $u_t$  during the time interval  $\Delta_t$ . The control  $u_t$  at  $t$  moment is expressed as  $u_t = (v, w)^T$ .

Assuming that the robot keeps constant  $(v, w)^T$  during a brief time interval  $\Delta_t$  with an initial state of  $s_t = (x, y, \theta)^T$ , the state  $s_{t+1} = (x', y', \theta')^T$  is derived as:

$$\begin{pmatrix} x' \\ y' \\ \theta' \end{pmatrix} = \begin{pmatrix} x_c + \frac{v}{w} \sin(\theta + w\Delta_t) \\ y_c - \frac{v}{w} \cos(\theta + w\Delta_t) \\ \theta + w\Delta_t \end{pmatrix} = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix} + \begin{pmatrix} -\frac{v}{w} \sin \theta + \frac{v}{w} \sin(\theta + w\Delta_t) \\ \frac{v}{w} \cos \theta - \frac{v}{w} \cos(\theta + w\Delta_t) \\ w\Delta_t \end{pmatrix} \tag{22}$$

where  $(x_c, y_c)$  is the center of the circular path of the robot and  $x_c = x - \frac{v}{w} \sin \theta$ ,  $y_c = y + \frac{v}{w} \cos \theta$ .

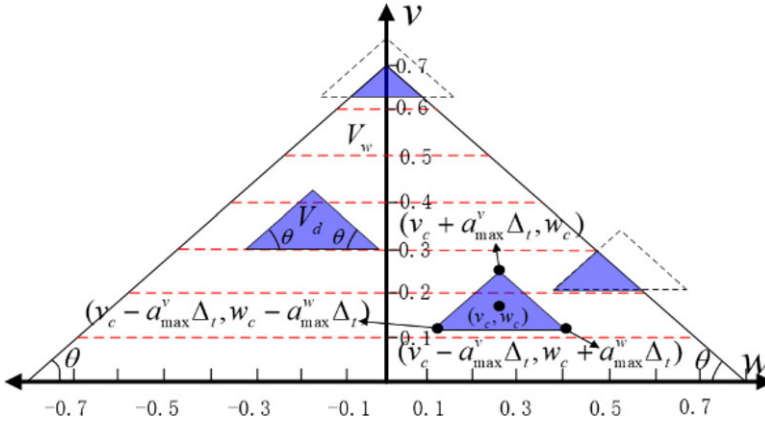


Figure 9. Dynamic triangle window.

In fact, there are limited relationships among linear speed, angular velocity and motor performance of robot. The maximum linear speed is Eq. (23), but the angular velocity at this time is zero as Eq. (24):

$$v = \lim_{\substack{v_l \rightarrow v_{\max} \\ v_r \rightarrow v_{\max}}} (v_r + v_l)/2 = v_{\max} \tag{23}$$

$$w = \lim_{\substack{v_l \rightarrow v_{\max} \\ v_r \rightarrow v_{\max}}} (v_r - v_l)/l = 0 \tag{24}$$

where  $v_l$  is the speed of the left wheel,  $v_r$  is the speed of the right wheel,  $v_{\max}$  is the maximum speed for each wheel, and  $l$  is the distance between two wheels.

Thus, the robot cannot increase angular velocity merely by increasing the speed of a certain wheel. Similarly, the linear speed will be zero at the point of maximum angular velocity. Therefore, based on this characteristic, this paper proposes a dynamic triangle window that aligns with the actual situation, as shown in Fig. 9. It is considered that all the optional speed pairs of the robot are included in the accessible space within a triangle during the moving process, and the model is used to study the local path planning algorithm to improve the efficiency of planning.

The dynamic triangle window first determines the reachable space  $V_w$  of linear speed and angular speed according to the motor speed, which includes all the actual speed pairs  $(v, w)$  as:

$$V_w = \left\{ (v, w) \mid -\frac{v_{\max}}{w_{\max}}|w| + v_{\max} - v \geq 0 \wedge v \geq 0 \right\} \tag{25}$$

Due to the maximum linear acceleration  $a_{\max}^v$  and angular acceleration  $a_{\max}^w$  have been known, the optional velocity pair window (blue block shown in Fig. 9) is a small triangle similar to the overall reachable space, which contains all the speed pairs  $(v, w)$  that the robot can make at the next moment.

The key point of the connection between robot navigation and movement is to select a point as the next speed control instruction in the dynamic triangle window  $V_d$ . However, since  $V_d$  is a continuous Euclidean space and cannot be used for the selection and evaluation of speed pairs, it is necessary to divide  $V_d$  into grids and complete the sampling process of speed pairs  $(v, w)$ .

Taking specific numerical values as an example, based on the parameters in Table VI, where  $n_v$  and  $n_w$  represent the number of discrete divisions for the  $v$  and  $w$  in the dynamic triangle window  $V_d$ , and using the current speed pair  $(v_c, w_c)$  from Table VII, we can obtain five discrete velocities for the triangular area  $V_d$  after discretization, as shown in Table VII.

After velocity sampling, assuming that the current pose of the robot is  $[P_x^t, P_y^t, \theta_t]^T$ , and the next state after the interval  $\Delta$ , can be derived according to Eq. (22). However, due to the value  $\Delta_t$  is small, it

**Table VI.** Parameters in dynamic triangle window.

$v_{\max}$ (m/s)	$w_{\max}$ (rad/s)	$a_{\max}^v$ (m/s <sup>2</sup> )	$a_{\max}^{wv}$ (m/s <sup>2</sup> )	$\Delta_t$ (s)	$n_v$	$n_w$
0.7	0.8	0.7	1.5	0.1		3

**Table VII.** Speed pairs in dynamic triangle window.

	$v$ (m/s)	$w$ (rad/s)
$(v_c, w_c)$	0.2	0.2
$(v, w)_1$	0.13	0.05
$(v, w)_2$	0.13	0.20
$(v, w)_3$	0.20	0.20
$(v, w)_4$	0.27	0.20
$(v, w)_5$	0.13	0.35

is not enough to get features that can be used to assess. In order to generate the virtual path, we set the simulation time  $t_{sim}$  to show obvious trajectories. The trajectory is defined as a hypothetical circular arc made by the robot moving while following the line speed and angular speed commands within  $t_{sim}$ .

But actually, the circular arc is unable to be expressed in the program, so we will use time interval  $t_{sim}$  to linearize the circular arc. Using every straight segment of the robot during time interval  $t_{sim}$  to fit the circular, the robot state  $[P_x^e, P_y^e, \theta_e]^T$  at the end of  $t_{sim}$  is calculated by Eq. (22). The iteration and processing are integrated into the trajectory optimization equation as shown in Table VIII. Finally, a collection of trajectories is obtained.

### 3.2 Navigation trajectory optimization

After obtaining all feasible trajectories, it is necessary to evaluate the cost value of each trajectory to obtain the speed control instruction corresponding to the optimal trajectory.

#### 3.2.1. Trajectory optimization preprocessing

Navigation using only local paths is limited to local minima, necessitating its combination with global paths. However, when optimizing local paths within the local cost map, referencing global paths that extend beyond the local cost map becomes irrelevant. Therefore, we define the local guidance point which is the junction point of the global path and the local cost map and is updated with the moving process of the robot, so as to guide the robot never deviating from the global path as shown in Fig. 10.

Based on the local guidance point, this paper introduces the calculation methods of  $d^{path}$  and  $d^{goal}$ . We take the side length of the map grid as the distance unit, take the local guidance point and the subsegment of the global path in the local cost map as two goals, and take the minimum distance value between the predicted trajectory from the two goals under the influence of the global path subsegment and local path as the reference value of  $d^{path}$  and  $d^{goal}$ , respectively. As shown in Fig. 11, the larger the value, the greater the tendency of the robot to deviate from the local path and global path.

#### 3.2.2. Trajectory optimization equation

Through preprocessing, the dynamic triangle window is combined with the human social behavioral model to predict the human trajectory, and the trajectory optimization equation is proposed.



**Table VIII.** The configuration of the navigation algorithm.

**Algorithm: Planner implementation.**

**Input:**  $S_h(t); V_h(t)$ ; sets of positions and velocities of all humans at time  $t$   
 $P_x^t, P_y^t, \theta_t, v_t, w_t$ ; robot's position, orientation and velocity  
 $G_{path} \leftarrow A^*(global\_costmap, Start, Goal)$ ;

**Output:**  $BestTr$ , the optimal track;

Initialize:  $S_r(t_c) \leftarrow Start$ ;

```

1  While  $S_r(t_c) \neq Goal$  do
2      //Velocity Sampling
3       $V_r = CalcVelSamples(P_x^t, P_y^t, \theta_t, v_t, w_t)$ 
4      //Update pose/vel of the robot and humans
5       $[S_r(t_c), V_r(t_c)] \leftarrow UpdateRobotOdometry()$ 
6       $[S_h(t_c), V_h(t_c)] \leftarrow HumanDetection()$ 
7       $V_r \leftarrow$  list of dynamically admissable velocities
8      for all  $V(r, w) \in V_r$  do
9          for  $t = t_c + t_{sim} \cdot step \Delta t$  do
10             if  $Collision(S_r(t), costmap, S_h(t) = true)$  then
11                 collision = true;
12                 break
13             else
14                  $Tr_i \leftarrow push(S_r(t))$ 
15                  $[S_h(t+\Delta t), V_h(t+\Delta t)] = HAM(S_h(t), V_h(t), S_r(t), costmap)$ 
16                  $S_r(t+\Delta t) \leftarrow UpdateRobotPosition(V_r^t)$ 
17                  $d^{obs} = \min(d^{obs}, d^{obs}(t))$ 
18             end
19         end
20         if collision = true then
21              $Tr\_Cost_i = -1$ 
22         else
23              $Tr\_Cost_i = \alpha \cdot d^{goal}(v, w) + \beta \cdot d^{path}(v, w) + \gamma \cdot \exp\{-\eta[d^{obs}(v, w) - d^{goal}]\}$ 
24              $+ \kappa \cdot obs\_cost$ 
25         end
26          $Tr\_Set \leftarrow push(Tr_i, Tr\_Cost_i)$ 
27     end
28      $BestTr = MinimumCost(Tr\_Set)$ 
29     Excuse  $BestTr$  for one time step
30 return  $BestTr$ 

```

For a series of trajectories generated by dynamic triangle window, we use trajectory optimization equation  $TrCost$  to evaluate all trajectories:

$$TrCost(v, w) = \alpha \cdot d^{goal}(v, w) + \beta \cdot d^{path}(v, w) + \gamma \cdot e^{-\eta \cdot [d^{obs}(v, w) - d^{goal}]} + \kappa \cdot obs\_cost \tag{26}$$

where  $\alpha$  is the weight of  $d^{goal}$ ,  $\beta$  is the weight of  $d^{path}$ ,  $\gamma$  is the weight of  $d^{obs}$ , and  $\kappa$  is the weight of  $obs\_cost$ .

The specific definitions of each standard are as follows:

- Distance  $d^{goal}$  from the local guidance point: In this paper, the distance  $d^{goal}$  between the grid at the end point of the generated trajectory and the local guide point is obtained. The larger this value is, the further the path is away from the global path.

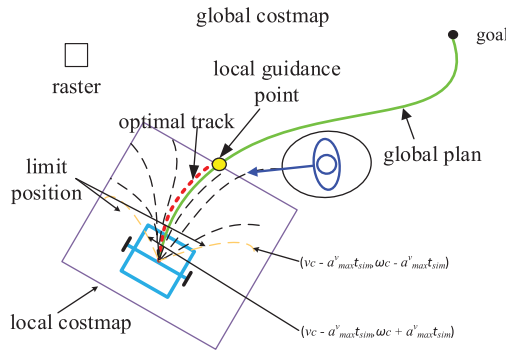


Figure 10. Trajectory preprocessing and local guidance point setting.

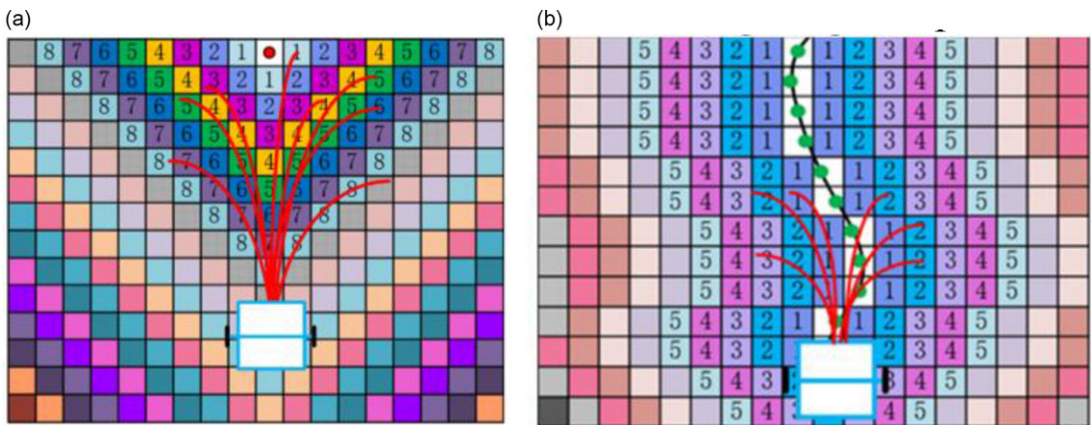
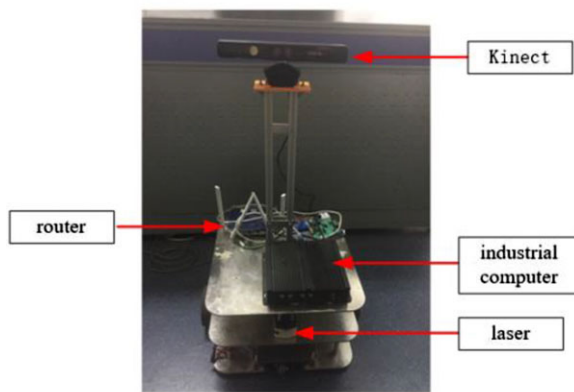


Figure 11. Estimated distance of cost map. (a) Calculation of distance from local guidance point. (b) Calculation of the minimum distance from the global path.

- The deviation  $d^{path}$  from the global path: The optimal trajectory should be oriented toward the local guidance point and close to the global path. Therefore, in this paper, the shortest distance  $d^{path}$  between the last point of trajectory  $Tr$  and the global path is taken as the shortest distance. The smaller the value is, the closer the path is to the global path.
- Minimum distance  $d^{obs}$  from pedestrian: This evaluation standard represents the minimum distance from pedestrians along the whole generated track. Human pose  $S_h(t_c)$  and speed  $V_h(t_c)$  are predicted in each time step length  $t \in [t_c, t_c + t_{sim}]$ . The distance  $d^{obs}(t_c)$  is obtained according to  $S_h(t_c)$  and  $V_h(t_c)$ . Using the constant speed prediction model to obtain the human state at the next moment:  $S_h(t + \Delta_t)$  and  $V_h(t + \Delta_t)$ , and then obtain the pose of the robot  $S_r(t_c + \Delta_t)$ , the distance  $d^{obs}(t_c + \Delta_t)$  can be figured out. By calculating all the  $d^{obs}(t_c)$  in the whole simulation time, the minimum value  $d_{min}^{obs}$  is used to evaluate the trajectory  $Tr$  that is negatively correlated with the cost value  $d^{obs}$ .
- Cost value  $obs\_cost$  of the trajectory in the grid: In this paper, the cost value of trajectory is affected by the cost value of cost map grid points. The cost value acquired at the end point of each trajectory is generated as  $obs\_cost$ . If some fatal obstacle is touched in the generated trajectory, this trajectory is directly eliminated. Trajectory with smaller  $obs\_cost$  has the priority when no fatal obstacle is found.



**Figure 12.** *The mobile robot.*

The optimization equation selects the path with the lowest total cost value, and the corresponding velocity pair  $(v, w)$  for the trajectory represents the speed control instruction for the robot at the next moment.

### 3.3 *The frame of the algorithm*

When a target point is set for the robot, the navigation algorithm combines the dynamic triangle window with human perception and applies the trajectory optimization equation to all the generated trajectories to select the optimal speed. The overall configuration of the navigation algorithm is shown in Table VIII.

## 4. Experiment

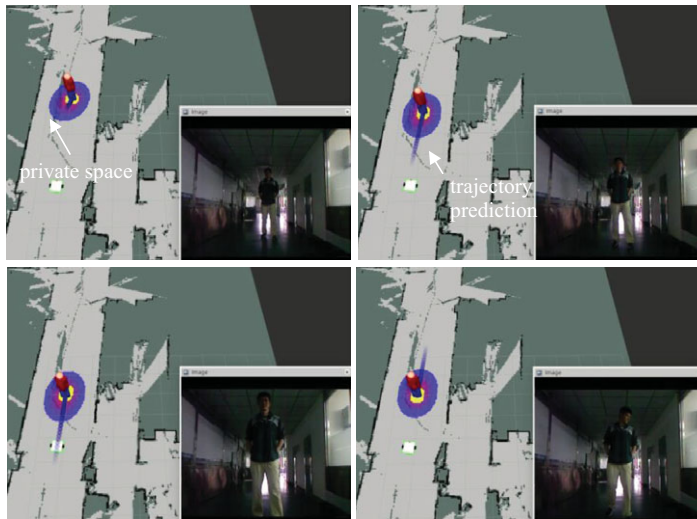
The experimental platform used in this paper is independently built by the laboratory as shown in Fig. 12.

The experimental platform is a two-wheel differential drive platform. The robot body is 500 mm × 500 mm. The platform is a three-layer structure with the top layer for industrial control, router, and Kinect, the middle one for the bottom control module and laser radar, and the bottom layer for the installation of motors and power module. At the core of the robot chassis control system lies the Freescale i. Max 28 ARM chip. The upper control system utilizes a portable PC running Ubuntu 14.04 and ROS Indigo. This configuration allows for the effective collection of sensor data and the execution of autonomous navigation tasks.

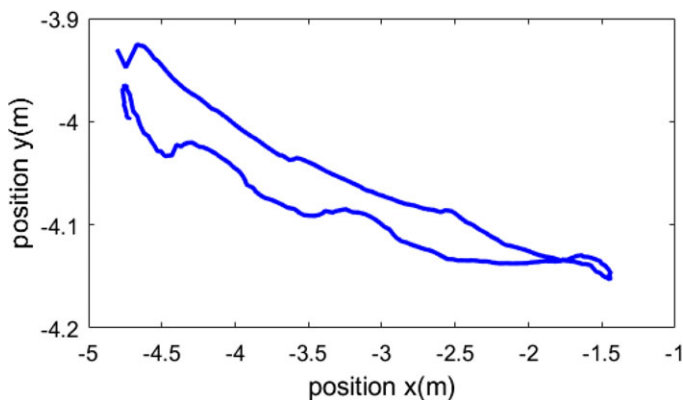
The platform is equipped with two environmental sensors. The first is the Sick TiM 561 laser radar, which can provide 2D point cloud information of the horizontal plane with high precision and strong environmental anti-interference capability. In order to increase the richness of the robot perception, the RGB-D sensor (Kinect) is also added on the platform. It has rich three-dimensional point cloud information, and the depth information can be used to sense the human body, which provides extensive scalability for the function of the robot.

### 4.1. *Behavioral intention cognition experiment*

The experiment combines human body detection, social behavioral model, and trajectory prediction in the cognitive layer of behavioral intention. In this experimental scenario, a single person gradually approaches a stationary robot in a corridor. The social behavioral model defines the private space around the human body in the behavioral intention cognition layer and assigns certain costs value to the grids



**Figure 13.** Experiment of behavioral intention cognition.



**Figure 14.** Trajectory of pedestrian.

around according to the social behavioral model function. In the rviz emulator, these cost values are represented with different colors.

Figure 13 depicts the evolving scene graph over time. Notably, the pedestrian's private space model transitions from yellow to blue in the cost map, signifying a progressive reduction in the cost value. Experimental findings substantiate the robot's ability to extract human information from the behavioral intention cognition layer, allowing it to perceive human behavioral intention. In forthcoming experiments, these social constraints can be effectively integrated into the navigation system.

Figure 14 depicts the pedestrian trajectory map, while Fig. 15 displays information related to pedestrian speed, orientation, and trajectory prediction length. Figure 16 demonstrates that the magnitude of collision probability directly influences the variance of the two-dimensional Gaussian distribution, subsequently shaping the social behavioral model. Specifically, when the relative distance decreases and the speed increases, collision probability grows, leading to an increase in variance. Consequently, the social behavioral model allows the robot to dynamically adjust its choice of human private space based on the risk of collision.

From the pedestrian data detected, this paper extracts information pertaining to the identified pedestrian within the behavioral intention cognition layer. Figure 17 illustrates the cost value curve within the multilevel cost map aligned with the vector representing the pedestrian's trajectory. The cost value curve

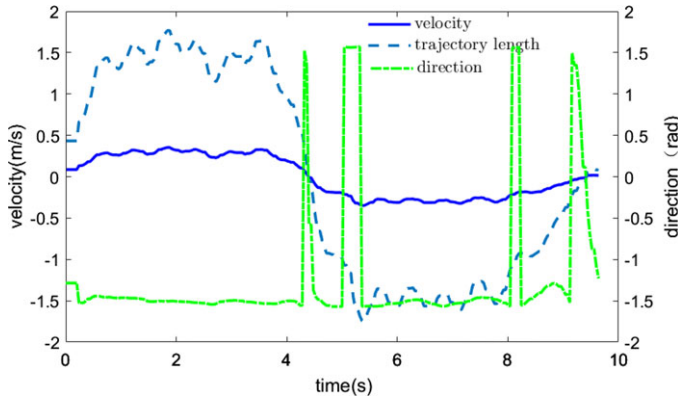


Figure 15. Pedestrian velocity, trajectory prediction length, and direction.

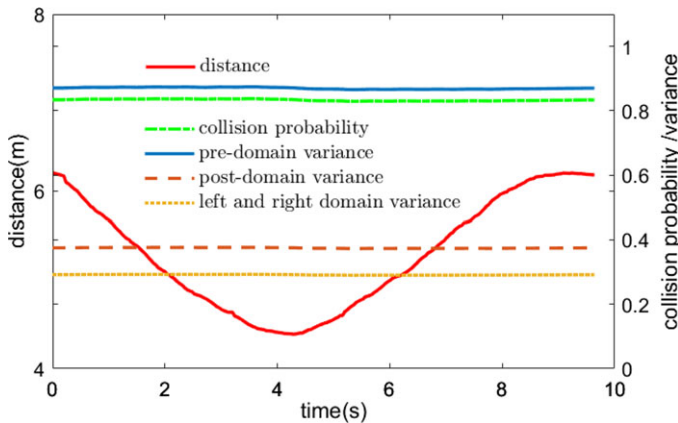


Figure 16. Time–distance, collision probability, and variance curve.

indicates that at the central position of the detected pedestrian, a cost value of 255 is recorded, signifying a critical obstacle. This value gradually decreases as the distance from the pedestrian’s center increases. Due to variations in the variances between the pre-domain and post-domain, distinct attenuation trends are observed in their respective values.

## 4.2. The Human-Aware Navigation experiments

### 4.2.1. Experimental scene 1

First, static experiments are designed to compare the results of traditional DWA algorithm and the proposed algorithm in path planning in a static pedestrian environment. The experimental setting was chosen to be a laboratory corridor, with the pedestrian positioned between the robot’s starting point and the target point.

The use of the traditional DWA algorithm is illustrated in Fig. 18. To facilitate a clear observation of the spatial relationship between the path planned by the robot and pedestrians, pedestrian detection is carried out during the experiment. Over time, it becomes evident that the robot only regards humans as ordinary obstacles, often approaching them too closely, which may lead to discomfort for people. The concept of private space for humans can not be incorporated into the path planning algorithm.

The HAN navigation algorithm is also employed in the same experimental scenario. As depicted in Fig. 19, it represents a temporal scene graph and records the distance between the pedestrian and

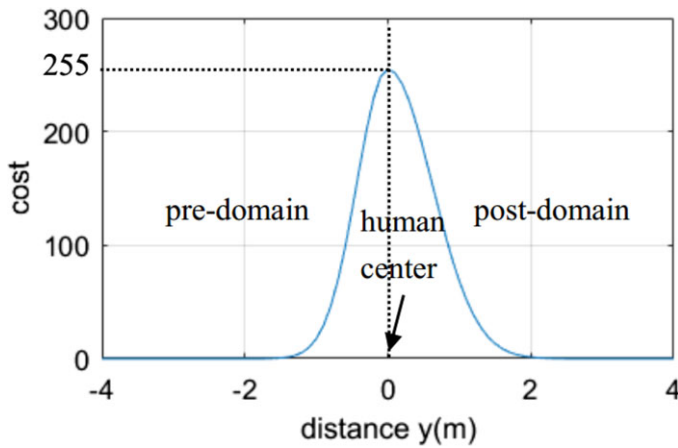


Figure 17. Cost curve of pedestrian direction.

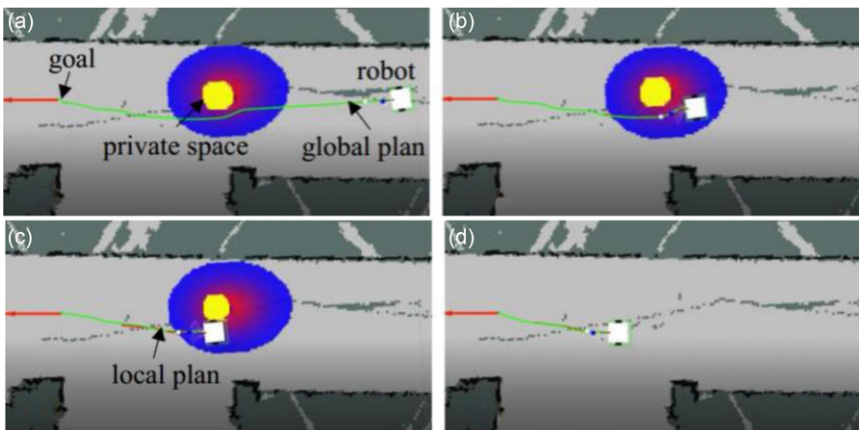


Figure 18. DWA algorithm.

the robot throughout the entire path planning process. This verification demonstrates the algorithm’s capability to enable the robot to distinguish pedestrians from other common obstacles.

From Fig. 19, it can be seen that the trajectory optimization equation not only allows the robot to plan a reasonable path but also respects the private space. Compared with Fig. 18, it can be concluded that the proposed algorithm can make the robot maintain a more reasonable distance with the pedestrian during the movement, thereby increasing the comfort of human.

Similarly, according to the comparison presented in Fig. 20 and Table IX, the social behavioral model can dynamically describe human private space. In nearly identical collision probability scenarios, where the experimental environment remains largely consistent, the proposed algorithm in this study results in a greater distance between the machine and the human. This approach, in turn, better respects the individual’s private space.

4.2.2. Experimental scene 2

The second experiment is also conducted within the corridor. In this experiment, both the robot and the person initiate their motion in opposite directions along the same straight path. The trajectory of the robot’s planned path and the relative positional relationship between the human and the robot are documented, as shown in Fig. 21.



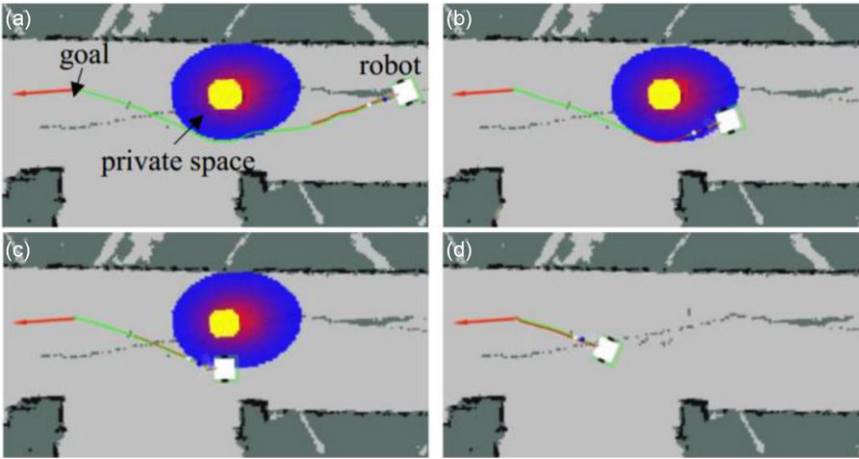


Figure 19. HAN algorithm.

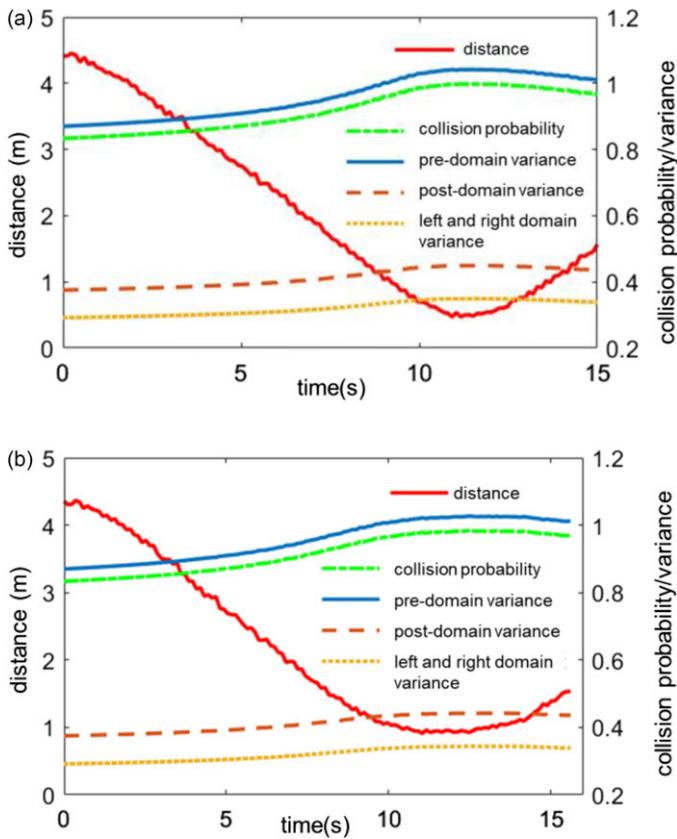
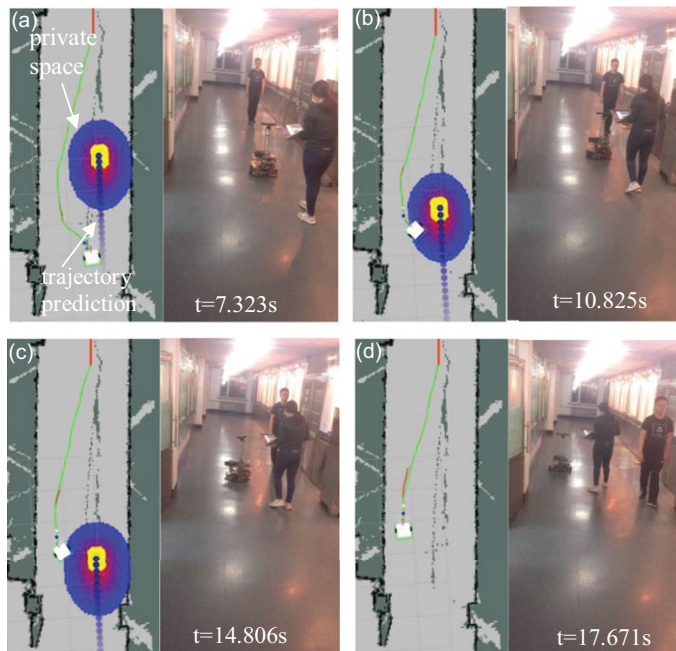


Figure 20. Time–distance, collision probability, and variance curve. (a) DWA algorithm. (b) HAN algorithm.

**Table IX.** Navigation results in a static environment.

	HAN algorithm	DWA algorithm
Average pedestrian speed [m/s]	0.0	0.0
Average pedestrian position [rad]	5.918	5.934
Planning times	62	58
Minimum distance [m]	0.849	0.5022



**Figure 21.** The result of HAN algorithm in real scenes. (a)  $t = 7.323$  s. (b)  $t = 10.825$  s. (c)  $t = 14.806$  s. (d)  $t = 17.671$  s.

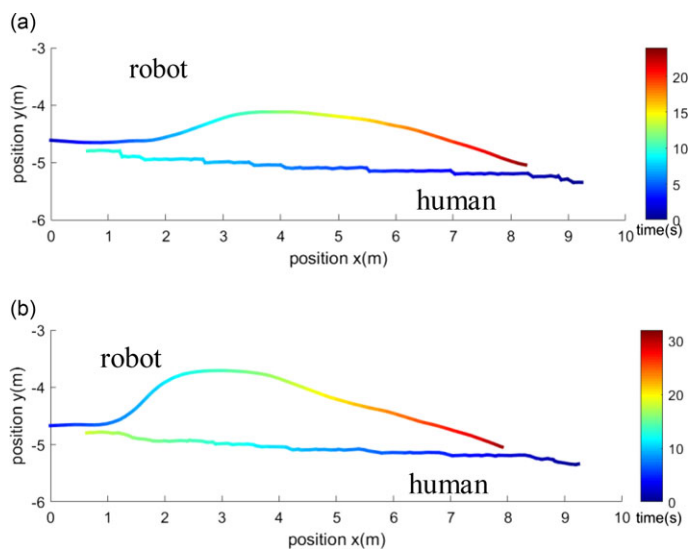
As shown in Fig. 21(a), the predicted human trajectory does not intersect with the robot's path planning scope, resulting in the robot remaining unresponsive to the detected pedestrian. However, as the robot enters the pedestrian's vicinity, as shown in Fig. 21(b) and (c), it promptly shifts to the opposite side of the corridor, thereby ensuring ample space for the individual and avoiding any obstruction to their path. As shown in Fig. 21(d), when the pedestrian exits the range of the laser radar detection, the robot no longer considers their presence and continues to employ trajectory optimization equations to determine the most optimal path.

For the same experimental scene, this paper compares the DWA algorithm with the proposed algorithm in their dynamic performance. Figure 22 illustrates the trajectory of both the robot and the human's actual movement, with RGB data employed to represent the time axis.

From the comparison of Fig. 22(a) and (b), it is evident that the HAN algorithm initiates the robot's avoidance maneuver at  $t = 7.323$  s, while the DWA algorithm begins avoidance at  $t = 8.756$  s. Therefore, the navigation algorithm studied in this paper demonstrates an ability to make decisions earlier based on the predicted trajectory of pedestrians in the scene, without significantly impacting the pedestrians' paths. Examining the distance data between the robot and the pedestrian in Fig. 22, it becomes apparent that the HAN algorithm effectively maintains a reasonable separation between the robot and pedestrian, thereby achieving the objective of respecting human private space.

**Table X.** *Navigation results in a corridor scene.*

	HAN algorithm	DWA algorithm
Average pedestrian speed [m/s]	0.957	0.962
Average pedestrian position [rad]	5.936	5.917
Planning times	128	102
Minimum distance [m]	2.06	0.849

**Figure 22.** *Trajectory comparison in corridor scenes. (a) DWA algorithm. (b) HAN algorithm.*

The curve of the variance parameters of pedestrian's private space, specifically social behavioral model, is shown in Fig. 23. A comparison between the data presented in Table X and the contrast between Fig. 23(a) and (b) reveal that, under nearly identical collision probabilities, the distance between the robot and the human is greater when the HAN algorithm is employed. This effect demonstrates the algorithm's ability to better respect the private space of individuals in a social context.

#### 4.2.3. Experimental scene 3

The third typical scenario involves an intersection with a unique characteristic: the robot's environmental sensing is limited to the area surrounding the door. It is only when the robot approaches the door that the laser radar can collect data from outside. If a pedestrian is suddenly detected at this point, there may not be sufficient time to plan a reasonable path based on predicted movements. Consequently, in the event of unforeseen circumstances resulting from dynamic changes in the environment, the robot may make inconsistent decisions. The algorithm presented in this paper addresses this issue by instructing the robot to come to a halt upon encountering a sudden obstacle, and it will remain stationary until safety is confirmed. Once confirmed, the robot will then proceed along its original path.

The experimental scenario involves both the robot and a person arriving near a door simultaneously. Figure 24(a) shows that the robot detects the pedestrian and determines the pedestrian's private space and movement trend. At this point, the pedestrian is treated as an immediate safety concern, an emergency obstacle. Figure 24(b) represents that when faced with the presence of this emergency obstacle, the robot halts and remains stationary until the pedestrian moves out of the way as shown in Fig. 24(c). Once the robot confirms that it is in a safe state, it then resumes its course toward the intended destination along the original path.

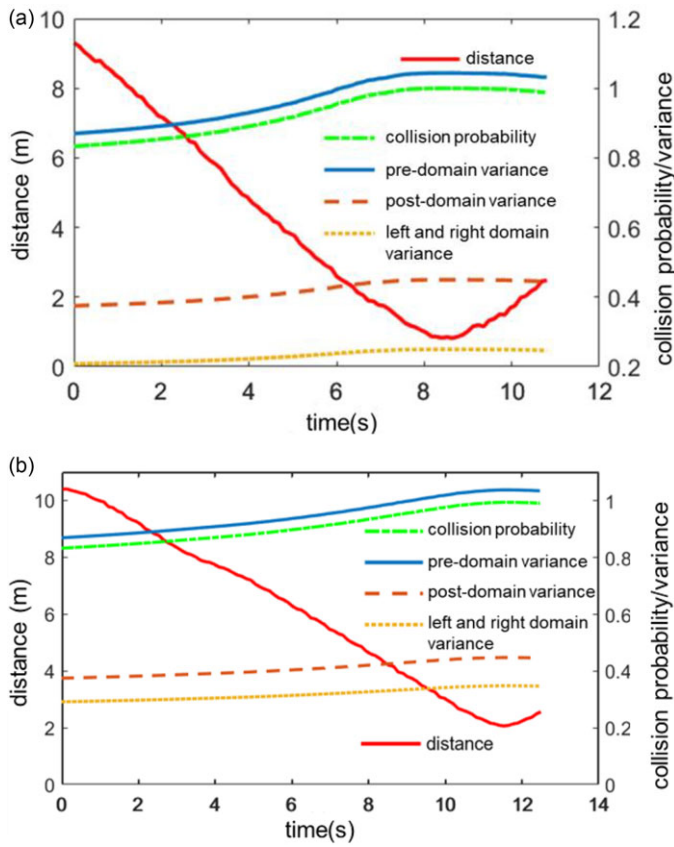


Figure 23. The distance between a robot and pedestrian. (a) DWA algorithm. (b) HAN algorithm.

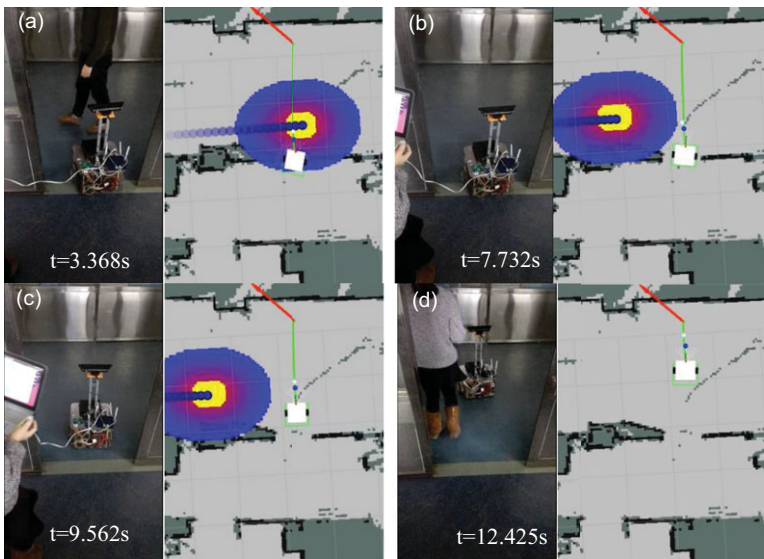
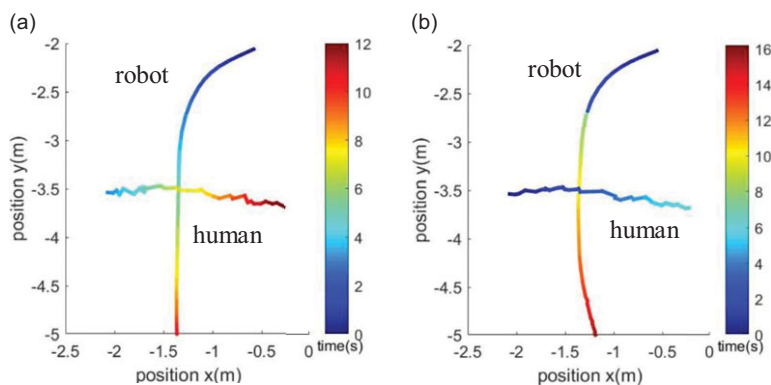


Figure 24. The result of HAN algorithm in special cases. (a)  $t=3.368s$ . (b)  $t=7.732s$ . (c)  $t=9.562s$ . (d)  $t=12.425s$ .

**Table XI.** Navigation results at intersection.

	HAN algorithm	DWA algorithm
Average pedestrian speed [m/s]	0.493	0.465
Average pedestrian position [rad]	0.538	0.549
Planning times	64	48
Minimum distance [m]	0.771	0.287



**Figure 25.** Comparison of robot and human trajectory at intersection. (a) DWA algorithm. (b) HAN algorithm.

In the same experimental scene, this paper compares the DWA path algorithm with the proposed algorithm and Fig. 25 shows the trajectories of the robot and the human.

Figure 25(a) presents the experimental data obtained using the DWA algorithm. The figure reveals that the robot does not take actions to avoid pedestrian behavior at the intersection. Comparing the positions of the robot and the pedestrian based on RGB data, it becomes evident that the robot crosses in front of the pedestrian, obstructing the pedestrian's intended path. In contrast, the HAN algorithm proposed in this paper addresses this issue. As shown in Fig. 25(b) and Table XI, the robot waits at the intersection at 3.576 s until the robot maintains a reasonable distance from the pedestrian and starts to travel along the original path at 7.431 s. This validates that the path planning algorithm presented in this paper offers a more effective strategy for responding to the sudden presence of pedestrians, thus enhancing human comfort to a certain extent.

## 5. Discussion

Through the implementation of human behavioral intention cognition and human–robot navigation trajectory optimization, we have realized the HAN algorithm that takes into account social factors. Analyses of the experimental results validates the effectiveness of the human behavioral model. In both static and dynamic navigation experiments, our data demonstrates that, in comparison to the traditional DWA method, which treats humans as dynamic obstacles, our HAN algorithm maintains a more appropriate distance from humans. The human model exhibits greater variance, ensuring human comfort, safety, and priority. Our approach has been proven effective, and in a human–robot coexistence environment, the robot successfully navigates socially aware paths, which holds promise for applications in real-life scenario.

However, it is important to note that this study focused solely on individual human interaction. In practical scenarios, more complex situations involving multiple individuals are prevalent. Besides, it is worth noting that the path planning frequency is higher, resulting in longer navigation time. Therefore,

in the future, we aspire to achieve human–robot interaction in a manner similar to human–human interaction, promoting safety, harmony, and comfort in social navigation. Another avenue for improvement is to ensure human positive experience while enhancing the efficiency of robot operations.

## 6. Conclusion

In this paper, a mobile robot navigation algorithm based on human–robot coexistence environment is investigated. The following key aspects are explored:

1. A kind of behavior intention cognition method based on social behavioral model using multilayer cost map is studied. The social behavioral model is proposed to depict the concept of human private space, where the shape of the private space can adapt based on the collision probability between the robot and the human.
2. Based on the improved multilayer cost map containing human information, the HAN algorithm is then studied. The optimization equation is designed to take into account the human private space and motion trend in trajectory optimization. This approach allows the robot to choose the optimal trajectory while considering the comfort of the human.
3. The social behavioral model and the motion trend are combined to generate the cost value, which is converted into a form that the robot can recognize and used by the navigation algorithm. The robot navigation algorithm is realized by utilizing the rich scalability offered by the multilayer cost map.

**Author contributions.** Jiahao Li and Fuhai Zhang conceived and designed the study. Jiahao Li performed experiments. Jiahao Li, Fuhai Zhang, and Yili Fu wrote the article.

**Financial support.** This work was supported in part by the National Natural Science Foundation of China under Grant 62073097, in part by the Natural Science Foundation of Heilongjiang Province of China under Grant LH2021F027, and in part by the State Key Laboratory of Robotics and System, Harbin Institute of Technology (HIT), under Grant SKLRS202107B.

**Competing interests.** The authors declare no competing interest exist.

**Ethical approval.** Not applicable.

## References

- [1] H. S. Hoj, S. Hansen and E. Svanebjerg, “Probabilistic Model-Based Global Localization in an Airport Environment,” *In: 17th IEEE International Conference on Automation Science and Engineering (CASE)* (IEEE, 2021) pp. 1370–1375.
- [2] T. X. Tung and T. D. Ngo, “Socially Aware Robot Navigation Using Deep Reinforcement Learning,” *In: IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)* (IEEE, 2018) pp. 1–5.
- [3] A. Apraiz, G. Lasas and M. Mazmela, “Evaluation of user experience in human-robot interaction: A systematic literature review,” *Int. J. Soc. Robot.* **15**(2), 187–210 (2023).
- [4] T. Smith, Y. Chen, N. Hewitt, B. Hu and Y. Gu, “Socially aware robot obstacle avoidance considering human intention and preferences,” *Int. J. Soc. Robot.* **15**(4), 661–678 (2023).
- [5] R. Yuan, F. Zhang, J. Qu, G. Li and Y. Fu, “A novel obstacle avoidance method based on multi-information inflation map,” *Ind. Robot.* **47**(2), 253–265 (2020).
- [6] S. Kodagoda, S. Sehestedt and G. Dissanayake, “Socially aware path planning for mobile robots,” *Robotica* **34**(3), 513–526 (2016).
- [7] M. Predhumeau, A. Spalanzani and J. Dugdale, “Pedestrian behavior in shared spaces with autonomous vehicles: An integrated framework and review,” *IEEE Trans. Intell. Veh.* **8**(1), 438–457 (2021).
- [8] F. Melo and P. Moreno, “Socially Reactive Navigation Models for Mobile Robots,” *In: IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (IEEE, 2022) pp. 91–97.
- [9] E. A. Sisbot, L. F. Marin-Urias, R. Alami and T. Siméon, “A human aware mobile robot motion planner,” *IEEE Trans. Robot.* **23**(5), 874–883 (2007).
- [10] Y. Wen, X. Wu, K. Yamane and S. Iba, “Socially-Aware Mobile Robot Trajectories for Face-to-Face Interactions,” *In: 14th International Conference on Social Robotics (ICSR)* (2022) pp. 3–13.



- [11] Y. Gao and C. Huang, “Evaluation of socially-aware robot navigation,” *Front. Robot. AI* **8**, 721317 (2022). doi: [10.3389/frobt.2021.721317](https://doi.org/10.3389/frobt.2021.721317).
- [12] K. Charalampous, I. Kostavelis and A. Gasteratos, “Recent trends in social aware robot navigation: A survey,” *Robot. Auton. Syst.* **93**, 85–104 (2017).
- [13] H. Miao and Y. Zhu, “Dynamic robot path planning using an enhanced simulated annealing approach,” *Appl. Math. Comput.* **222**(5), 420–437 (2013).
- [14] E. W. Dijkstra, *A Discipline of Programming* (Prentice Hall, Englewood Cliffs, NJ, 2015).
- [15] W. Tang, Y. Zhou, T. Zhang, Y. Liu, J. Liu and Z. Ding, “Cooperative collision avoidance in multirobot systems using fuzzy rules and velocity obstacles,” *Robotica* **41**(2), 668–689 (2023).
- [16] X. Yu, Y. Zhu, L. Lu and L. Ou, “Dynamic window with virtual goal (DW-VG): A new reactive obstacle avoidance approach based on motion prediction,” *Robotica* **37**(8), 1438–1456 (2019).
- [17] B. Guo, N. Guo and Z. Cen, “Obstacle avoidance with dynamic avoidance risk region for mobile robots in dynamic environments,” *IEEE Robot. Autom. Lett.* **7**(3), 5850–5857 (2022).
- [18] L. Liu, X. Wang, X. Yang, H. Liu, J. Li and P. Wang, “Path planning techniques for mobile robots: Review and prospect,” *Expert Syst. Appl.* **227**, 120254 (2023). doi: [10.1016/j.eswa.2023.120254](https://doi.org/10.1016/j.eswa.2023.120254).
- [19] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar and A. García-Cerezo, “Path planning for autonomous mobile robots: A review,” *Sensors* **21**(23), 7898 (2021). doi: [10.3390/s21237898](https://doi.org/10.3390/s21237898).
- [20] A. N. A. Rafai, N. Adzhar and N. I. Jaini, “A review on path planning and obstacle avoidance algorithms for autonomous mobile robots,” *J. Robot.* **2022**, 1–14 (2022). doi: [10.1155/2022/2538220](https://doi.org/10.1155/2022/2538220).
- [21] T. T. Mac, C. Copot, D. T. Tran and R. De Keyser, “Heuristic approaches in robot path planning: A survey,” *Robot. Auton. Syst.* **86**, 13–28 (2016).
- [22] X. Zhang, Y. Zhao, N. Deng and K. Guo, “Dynamic path planning algorithm for a mobile robot based on visible space and an improved genetic algorithm,” *Int. J. Adv. Robot. Syst.* **13**(3), 1–17 (2016).
- [23] H. Qin, S. Shao, T. Wang, X. Yu, Y. Jiang and Z. Cao, “Review of autonomous path planning algorithms for mobile robots,” *Drones* **7**(3), 211 (2023). doi: [10.3390/drones7030211](https://doi.org/10.3390/drones7030211).
- [24] L. Zhou, C. Zhu and X. Su, “SLAM algorithm and Navigation for Indoor Mobile Robot Based on ROS,” **In: IEEE 2nd International Conference on Software Engineering and Artificial Intelligence (SEAI)** (IEEE, 2022) pp. 230–236.
- [25] T. Dang, “Autonomous mobile robot path planning based on enhanced A\* algorithm integrating with time elastic band,” *MM Sci. J.* **2023**, 6717–6722 (2023).
- [26] O. Khatib, “Real-time obstacle avoidance for manipulators and mobile robots,” *Int. J. Robot. Res.* **5**(1), 90–98 (1986).
- [27] L. Zhang, X. Shi, Y. Yi, L. Tang, J. Peng and J. Zou, “Mobile robot path planning algorithm based on RRT\_connect,” *Electronics* **12**(11), 2456 (2023). doi: [10.3390/electronics12112456](https://doi.org/10.3390/electronics12112456).
- [28] S. Bhattacharya, R. Ghrist and V. Kumar, “Persistent homology for path planning in uncertain environments,” *IEEE Trans. Robot.* **31**(3), 578–590 (2017).
- [29] R. Yuan, F. Zhang, Y. Wang, Y. Fu and S. Wang, “A Q-learning approach based on human reasoning for navigation in a dynamic environment,” *Robotica* **37**(3), 445–468 (2019).
- [30] E. T. Hall, *The Hidden Dimension* (Anchor Books, Garden City, NY, 1990).