

Received July 12, 2020, accepted August 2, 2020, date of publication August 13, 2020, date of current version August 26, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3016472

# A Robust Autonomous Following Method for Mobile Robots in Dynamic Environments

**DAPING JIN**<sup>ID</sup>, **ZHENG FANG**, (Member, IEEE), AND **JIEXIN ZENG**

Faculty of Robot Science and Engineering, Northeastern University, Shenyang 110819, China

Corresponding author: Zheng Fang (fangzheng@mail.neu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61573091, in part by the Fundamental Research Funds for the Central Universities under Grant N182608003 and Grant N172608005, in part by the National Defense Basic Scientific Research Program of China under Grant 6142414200208, in part by the Natural Science Foundation of Liaoning Province under Grant 20180520006, and in part by the Major Special Science and Technology Project of Liaoning Province under Grant 2019JH1/10100026.

**ABSTRACT** Autonomous following is one of the key issues in mobile robotics with a wide range of applications in commercial, industrial and military areas. To achieve autonomous following, the robot needs not only to robustly detect and track the target, but also efficiently follow the target while avoiding obstacles. In this paper, we propose a method that combines a Line-of-sight sensor (a 2D laser range finder) and a Non-line-of-sight sensor (a low-cost Angle-of-Arrival (AOA) tag) to identify, track and follow the target person in dynamic environments. First, in order to get smooth and continuous moving trajectory of the target person, a Kalman Filter is used to fuse person tracking information from AOA data and laser data. Then, a real-time robot-centric rolling grid map is constructed using the laser data. On top of the rolling grid map, a target potential field is generated by using the fast marching method, and then a direction gradient field is created based on the target potential field. With the rolling grid map, the target potential field and the direction gradient field, an improved dynamic window algorithm, FMM-DWA, is proposed to control the robot to move towards the target. The proposed method considers not only the distance between the robot and the target, but also the difference between the heading of the robot and the reference direction provided by the direction gradient field, to avoid falling into the local optimum. To validate the performance of the proposed method, a series of person following experiments are carried out in complex dynamic environments. The experimental results show that the proposed following algorithm can effectively deal with the occlusion problem and robustly follow the target person while quickly avoiding the static and dynamic obstacles.

**INDEX TERMS** Autonomous following, person tracking, FMM-DWA, local path planning, dynamic environments.

## I. INTRODUCTION

With the continuous development of robotic technologies, robots have gradually come into our daily life. Nowadays, robots not only need the ability to accomplish tasks themselves, but also the ability to collaborate with humans. Autonomous person following is one of such kind of ability with a wide range of applications in commercial, industrial and military areas. For example, an autonomous following robot can carry heavy goods for customers in big shopping malls to enhance the shopping experience [1]. Another example is that an autonomous following robot can lead or follow a worker to a specific zone in big logistic

warehouses to pick goods and return to the packaging area autonomously [2]. Notably, Boston Dynamics' LS3 legged robots have a well-developed person-following capability to act as load carrying mules [3]. For such kind of applications, one key step is human tracking and following, where the robot need not only to robustly track the person but also smoothly follow the person while avoiding both static and dynamic obstacles.

Currently, cameras are widely used for autonomous following tasks [4], [5]. Cameras have many advantages for this task, such as low cost, compact and having abundant information. However, cameras are very sensitive to illumination changes which following robots always suffer in practice. Besides, it is difficult for cameras to determine accurate distance to the target and surrounding obstacles, which pose

The associate editor coordinating the review of this manuscript and approving it for publication was Malik Jahan Khan<sup>ID</sup>.

great challenges for efficient following in complex environments. In addition to cameras, laser scanners can also be used for person tracking and following [6], [7]. Compared to cameras, laser scanners are capable of providing accurate distance to the target person and surrounding obstacles, which will make following easier. There are also some works combine cameras and laser sensors to enhance the tracking ability [8], [9]. However, both laser scanner and camera are visual line-of-sight (LOS) sensors, which means they can not track target when severe occlusion happens. Unfortunately, service robots working in markets or warehouses always suffer dynamic obstacles blocking most part of the field-of-view or sudden losing target due to quickly turning around a corner, which will make the robot lose its target and then fail to plan a path to follow the moving target. Besides, those robots are also always suffering from unable to identify the staff with similar body figure and same uniform working in the market or warehouse. Besides LOS sensors, Non-LOS sensors can also be used for person tracking and following. For example, [10] developed a human-tracking ground robot and [11] proposed an omni-directional person tracking drone based on ultra-wideband (UWB) radio transceivers mounted on a robot and a target. However, though those methods can partially deal with tracking problem in occlusion environments. The tracking performance decreases dramatically if there are obstacles existing between the robot and person, where the tracking pose often jumps sharply. Furthermore, it is difficult to follow the person smoothly and avoid obstacles by only using UWB sensors.

In order to achieve robust person tracking and smooth following in dynamic environments, in this paper, we propose a method that combines a Line-of-sight sensor (a 2D laser range finder) and a Non-line-of-sight sensor (a low-cost Angle-of-Arrival (AOA) tag) to identify, track and follow the target person. First, the laser range finder is used to detect the human legs to obtain the location of pedestrians. At the same time, the AOA base-station equipped on the robot receives the data from the hand-held AOA tag, and the position of the tag is calculated using triangulation. In order to get smooth and continuous moving trajectory of the target person, a Kalman Filter is used to fuse person tracking information from AOA data and laser data. Then, a real-time robot-centric rolling grid map is constructed using the laser data. After that, a target potential field is generated by using the fast marching method [12], and a direction gradient field is created based on the target potential field. Finally, an improved dynamic window algorithm (FMM-DWA) with evaluation of the robot's heading is proposed to control the robot to move towards the target while avoiding falling into local optimum that traditional DWA [13] often encountered. By using AOA tag together with laser scanner, we not only solve the difficulty of identifying people with same figure or uniform, but also solve the occlusion problem widely encountered in the real world. In addition, laser sensors are very good for path planning and obstacle avoidance. The experimental results in various scenarios show that the proposed

tracking and following algorithms can effectively deal with the occlusion problem and robustly follow the target person while quickly avoiding the static and dynamic obstacles. The main contributions of the paper are as follows:

- A robust and accurate person tracking method by fusing information from a LOS sensor (a 2D laser range finder) and a Non-LOS sensor (an Angle-of-Arrival tag) is proposed, which can solve the occlusion issue caused by static and dynamic obstacles.
- A real-time target potential field is generated and added to the robot-centric rolling grid map, which can further calculate the direction gradient field to provide reference moving direction for the robot.
- The heading of the robot is introduced to evaluate the sampling trajectory, which not only evaluates the distance between the trajectory endpoint and the target point, but also evaluates the validity of the azimuth angle of the trajectory endpoint to avoid falling into local optimum.

The rest of the paper is organized as follows. Section II briefly describes the related work. Section III introduces the overall framework of the autonomous following method. Section IV details the target person identifying and tracking algorithm based on 2D laser scanner and AOA sensor. Section V describes the robot following and navigation algorithm. Section VI validates the performances of our navigation system in both simulated and real environments. Finally, Section VII concludes the paper.

## II. RELATED WORK

The autonomous following function is mainly realized by two steps. First, the robot identifies and tracks the target person. Second, the robot follows the moving target person in environments while avoiding both static and dynamic obstacles. We briefly describe some works related to those two steps.

### A. PERSON TRACKING

For efficient person tracking, various methods were proposed based on different sensors. Utilizing RGB-D cameras is a popular approach to achieve person-following [14], [15]. For example, Sun *et al.* used RGB-D cameras to detect and identify the target person and developed an indoor person following and companion robot [14]. Chi *et al.* proposed an RGB-D sensor-based human gait detection and recognition method to obtain the target person location [15]. However, RGB-D cameras usually have narrow horizontal field of view (FOV), and it is easy for target moving out of the FOV. Moreover, since cameras perform poorly in the presence of lighting variation, they are not good choices for outdoor environments. Besides cameras, laser scanners are also widely used by person-following robots on account of their stability and large field of view. For example, a 2D laser sensor-based person detection and tracking method usually determines the human location by geometric information of human legs [16], [17] or torso [18]. The person detection and tracking methods based on the 2D laser scanner are scarcely affected

by the illumination change, but the information obtained by the 2D laser scanner is relatively limited. For problems like this, Bohlmann *et al.* presented a system based on 3D laser sensors for autonomous following of a walking person in outdoor environments [7], which can achieve tracking more effectively. 3D laser sensors also have large field of view, so that they are not easy to lose the target. Nevertheless, if only the contour information of the individuals is used, it is difficult to distinguish the target person from other moving objects and misidentification will easily occur when there are many people in the environment. As aforementioned, there are many shortcomings of using only single sensor to detect and track the target. Therefore, researchers also developed object detection and tracking methods that incorporate multiple sensors. The general way is the combination of a camera and a 2D laser scanner [8], [19]–[21]. To improve the reliability of target recognition, [8] and [19] proposed a detection and identification method that combines an omnidirectional camera with a 2D laser scanner. The omnidirectional camera is used to identify the target, and the laser scanner is used to measure the distance of the person identified by the camera. In addition, [20] and [21] presented a method of combining one or more high-resolution cameras with a 2D laser scanner. Those high-resolution front cameras are used to re-identify the target person, which can solve the problem of short-term target loss when only using a 2D laser scanner to track the target. The 2D laser scanner is used to make up for the defect that the camera has a narrow field of view. However, in dynamic and uncertain environments, due to the complex and unpredictable occlusion situations, the combination of cameras and laser scanners to detect the pedestrian would not be enough. To solve this problem, wearable devices [22], [23], are also used for person tracking. Those devices can cope with the occlusion problem to some extent, but the unstable signal and limited detection range also lead to tracking pose jumping sharply and frequently.

### B. PERSON FOLLOWING

For person-following navigation, there have been some classic control methods, such as PID control [6], [24] and pure pursuit [25]. Those methods are easy to implement and have high execution efficiency, but they only calculate the angular and linear velocity of robot according to the distance and angle between the target and the robot, without considering obstacle avoidance. Analogously, Bohlmann *et al.* proposed a motion control method based on recorded path of the target person [7]. This assumes that the path taken by the person is an obstacle-free path. Obviously, it does not consider obstacle avoidance either. However, in real scenarios, there may be some dynamic obstacles in the environments when the robot is following the target person, so it is not safe to use above control methods in a dynamic environment. To achieve safe and autonomous following in dynamic and uncertain environments, researchers have also presented many person-following navigation methods. For example, Prassler *et al.* proposed an extended velocity

obstacle approach [26]. Firstly, the points of a circular area around the target person are taken as the virtual goal set. Secondly, the velocity obstacle approach is used to control the robot to move towards the nearest virtual goal. Furthermore, many path planning algorithms (RRT [27], Hybrid Astar [28]) are also used to generate a collision-free path to the virtual goal. When the target person stops or moves slowly, these methods can plan a collision-free path very well. However, when the target person frequently moves or there are many dynamic obstacles in environments, if the path planner cannot re-plan the path in real time, the efficiency of the person following will be reduced and even fail. In addition, there are also some following methods that do not specify the target point, but use simple obstacle avoidance algorithms to make the robot approach the target point. For example, Peng *et al.* developed a sonar-based tracking control strategy for person-following robot [29]. This method generates a path set while identifying the target person, and selects an optimal path to control the robot moving. Meanwhile, Zhang *et al.* also proposed a sonar-based control strategy for person-following robot and used virtual field force principle to control the robot [30]. However, neither of the two methods takes into account the motion model of the robot, but only regards the robot as a particle. Thus, the motion trajectory of the robot is not smooth. In addition to the above traditional following methods, [31] proposed an approach in which an agent is trained by hybrid-supervised deep reinforcement learning to perform a person following task in end-to-end manner. However, like some traditional methods, this method does not consider the avoidance of dynamic obstacles, and it's difficult to control the robot to follow people smoothly with action instructions directly generated from discrete images.

### III. SYSTEM OVERVIEW

The person-following algorithm proposed in this paper is tested on a differential-driven robot. As shown in Figure 1, the front of the mobile robot is equipped with an R2000 2D laser range finder from Pepperl+Fuchs and an AOA base-station module. The AOA base-station module consists of two AOA base stations, which receive the data of the hand-held AOA tag. The laptop is used as an execution platform of the algorithm proposed in this paper. And there are encoders at the two wheels of the robot, which can be used to calculate the odometry of the robot.

As shown in Figure 2, our person-following system mainly consists of three parts, namely the target person tracking, the rolling grid map creation and the FMM-DWA planner. To get the target person tracking information, the 2D laser sensor is used to detect and track the target person, while the AOA tag is used to identify the target person and measure his/her relative position, and then a Kalman Filter [32] is used to fuse the information from the two sensors to get a stable target pose. For autonomous following, firstly the rolling grid map is created based on the laser scanning information and the odometry of the robot. Then, based on the rolling grid map and the target pose, the FMM-DWA planner will generate the



FIGURE 1. Our hardware system configuration.

velocity instruction to control the robot moving towards the target and keep a constant distance from the target.

IV. TARGET PERSON TRACKING METHOD

Person identification and tracking is a widely-studied issue in mobile robotics. To achieve robust and accurate person tracking in dynamic or even occluded environments, in this paper, an AOA tag and a 2D laser range finder are used to solve the identification and tracking problem. With the AOA tag, it is also easy to solve the problem of tracking a specific person while there are many people with similar figure and uniform in the environment. However, the AOA localization is a wireless localization technology based on antenna arrays and angle-of-arrival measurements. There will be big noise when there are obstacles existing between the AOA transmitter and receiver, which will result in the pose estimation of the target jumps sharply. Therefore, in this paper, we try to use a Kalman Filter to fuse the tracking results from both laser scanner and AOA sensor to get a more stable and accurate tracking performance of the target person.

A. PERSON TRACKING USING AOA TAG

The AOA person tracking system is composed of two parts, namely a base-station and a tag. The tag is held by the target person, and the base-station is on the robot. The base-station module senses the direction of the tag signal and calculates the relative angle between the base-station and the tag. Thus, the position of target person could be determined by the position information estimated by the AOA localization system. The AOA localization system [33] is a two-base station localization system, which uses the incident angle of the signal to realize localization. The basic principle is that the directional antenna array at the base station is used to obtain the direction of the signal sent by the the tag, so as to determine the position of the tag according to the angle of arrival of the signal and the base station’s position. The position  $X$  of the target person expressed as  $(x, y)$  in two-dimensional coordinate system is then transformed into the odometry coordinate for later information fusion.

B. PERSON TRACKING USING LASER SCANNER

In our system, the 2D laser sensor equipped in the front of the mobile robot is about 40cm height from the ground. The laser scanning angular range is  $-180^\circ \sim 180^\circ$ , and its angular resolution is  $0.35^\circ$ . In order to track a person, we need to detect human legs in environments. First, the laser points from a scan are segmented into several clusters, where each cluster has at least five laser points whose mutual distance is less than a certain threshold. Second, geometric features are generated for each cluster, including the number of laser points, the width and length of laser clusters, the distance and angle relative to the laser sensor. Finally, based on those geometric features, clusters are further classified as human legs or non-human legs. The classification is done using a random forest classifier trained on a set of 1700 positive and 4500 negative examples [34]. The positive example is gathered by setting the scanner up in a populated area where we specify a bounding box, within which only peoples’ legs will appear. And the reverse example is gathered by moving the scanner on a mobile platform around empty rooms.

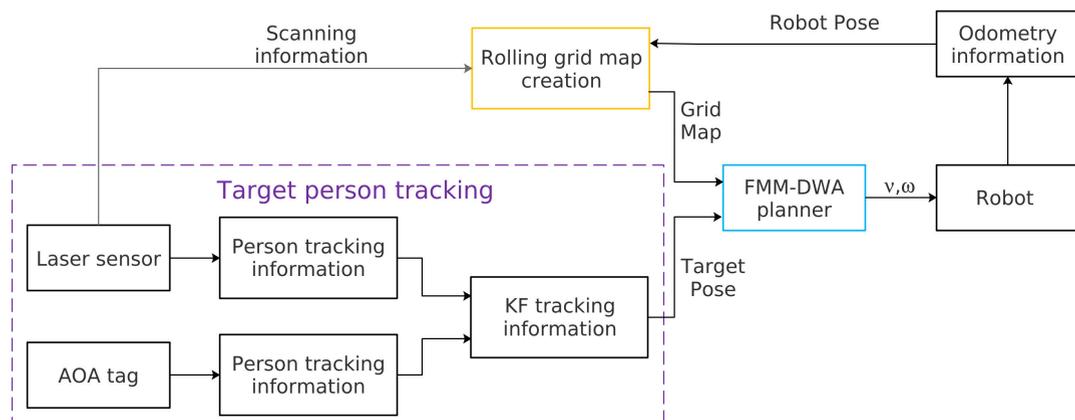
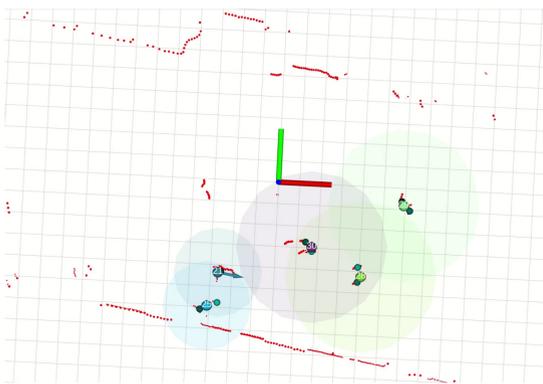


FIGURE 2. The architecture of our person-following system.

As shown in Figure 3, when the classification of clusters is completed, the person will be tracked based on the tracking information of human legs clusters. We assume that there are generally only two leg clusters within the range of 0.4m around a person. When there are other leg clusters within 0.4m of one leg cluster, we will take the average of the leg cluster position as the pedestrian position, and if there are no other leg clusters, the leg cluster position will be directly taken as the pedestrian position. At time  $t$ , the position of the human detected by the laser ranger finder is expressed as  $z_t = \{z_t^1, z_t^2, \dots, z_t^n\}$ , and  $n$  is the number of the human detected by the laser sensor at time  $t$ .



**FIGURE 3.** Human legs detection and human tracking. The red point clouds generated by 2d laser scanning are distributed in the grid diagram with the cell size set to 0.4m. They are classified into human leg clusters and non-human leg clusters. The human leg clusters are represented by circular marks without arrow and number, and marks with arrow or number represent the person detected by laser. The number is the id of the person, and the arrow represents the direction of movement of the person.

### C. KALMAN FILTER-BASED TRACKING

Since the tracking information provided by AOA tag is more robust against occlusion while the tracking information provided by laser scanner is more accurate, in our system we try to fuse them to get a more robust and accurate tracking performance. To achieve this, we first estimate the position of all people by detecting legs from laser points. Then, to identify the target person from other people, the position of the target person returned by the AOA tag is used as the initial value  $X$  of the target. Finally, a Kalman Filter is used to fuse the tracking information from two sensors. The Kalman filter involves two stages: prediction and measurement update. In the prediction stage, there are two important matrix, the state transition matrix  $A_t$  and the motion noise matrix  $R_t$ . The state transition matrix is set according to a constant velocity motion model used in the prediction stage. The matrix  $A_t$  is as follow:

$$A_t = \begin{pmatrix} 1 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 1 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (1)$$

Then, the motion noise  $R_t$  is generally set to be a little bit large. In the update stage, the update phase is divided into two steps: first, the location of the target person is updated by using the position measured by the AOA tag; second, the position detected by the laser range finder is used as an observation to update the first updating result. In the first step, since the position information measured by the AOA tag fluctuates greatly, the data is preprocessed using a sliding window filter to obtain  $Z_t^1$ . The observation noise matrix  $Q_t^1$  is generally set to be small. The second step uses the laser range finder to detect human legs as an observation. Although the position information measured by the AOA tag is not accurate, the value of  $Z_t^1$  does not deviate too much from the true value. The closest value  $Z_t^2$  to  $Z_t^1$  is selected from  $z_t$ , and if the distance between  $Z_t^1$  and  $Z_t^2$  is less than 0.8m, the second updating will be performed. The magnitude of the second measurement noise  $Q_t^2$  is related to the distance between the two position data closest to  $Z_t^1$  in  $z_t$ . The process of using Kalman Filter to fuse the data from the two sensors is described in Algorithm 1.

#### Algorithm 1 Kalman Filter

---

```

1 Kalman_Filter( $X_{t-1}, P_{t-1}, Z_t^1, z_t$ )
2 Constant velocity model prediction:
3  $\tilde{X}_t = A_t X_{t-1}$ 
4  $\tilde{P}_t = A_t P_{t-1} A_t^T + R_t$ 
5 AOA tag update:
6  $K_t = \tilde{P}_t H_t^T (H_t \tilde{P}_t H_t^T + Q_t^1)$ 
7  $X_t = \tilde{X}_t + K_t (Z_t^1 - H_t \tilde{X}_t)$ 
8  $P_t = (I - K_t H_t) P_t$ 
9 Laser update:
10  $Z_t^2 = \text{select}(Z_t^1, z_t)$ 
11 if  $\text{dist}(Z_t^1, Z_t^2) < 0.8$  then
12    $K_t = P_t H_t^T (H_t P_t H_t^T + Q_t^2)$ 
13    $X_t = X_t + K_t (Z_t^2 - H_t X_t)$ 
14    $P_t = (I - K_t H_t) P_t$ 
15 end
16 return  $X_t, P_t$ ;
```

---

### V. FOLLOWING AND NAVIGATION

In order to keep the robot at a certain distance from the tracked person while avoiding dynamic obstacles in environments, a FMM-DWA algorithm is proposed to generate trajectory of the robot, so that the robot can move to the target quickly and smoothly. First, a velocity field is constructed based on obstacles information in the rolling grid map created by using laser data. Second, a target potential field is created by the fast marching method based on the grid propagation speed provided by the velocity field, and then a direction gradient field is created based on the target potential field. Third, the trajectory evaluation function constructed based on the rolling grid map, the target potential field and the direction gradient field are used to select the optimal trajectory from

all sampling trajectories generated by the DWA. Finally, the velocity command is generated from the optimal trajectory to control the robot. Different from the traditional DWA algorithm, the FMM-DWA proposed in this paper can avoid falling into local optimum, because the heading of the robot is considered to evaluate the sampling trajectory. A block diagram of our following navigation is shown in Figure 4.

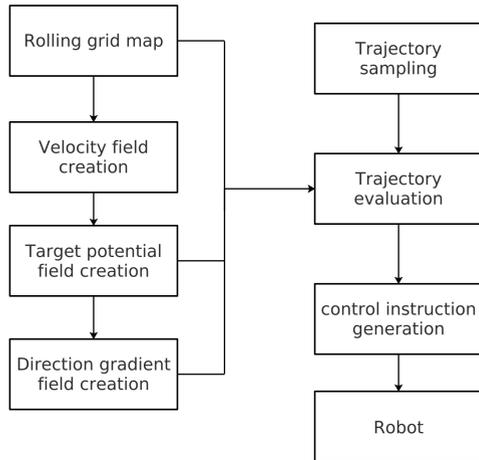


FIGURE 4. A block diagram of our following navigation.

### A. TARGET POTENTIAL FIELD GENERATION

The fast marching method (FMM) was introduced by Sethian in 1996 and is a numerical method that approximates the viscosity solution of the Eikonal equation [35]. The equation is as follows:

$$|\nabla T(x)| * W(x) = 1. \tag{2}$$

where  $x$  represents a point in the search space, and its representation in two-dimensional space is  $x = (x, y)$ .  $T(x)$  is the arrival time from the start point to point  $x$ , and  $W(x)$  is the local propagation speed of the interface at point  $x$ . By discretizing the gradient  $T(x)$ , the Eikonal equation can be solved for each point  $x$  in space, where  $x$  corresponds to the grid of the  $i$ -th row and  $j$ -th column in the planning space represented by the grid. The gradient expression of a point is simplified to Equation 3:

$$\begin{aligned} T_1 &= \min(T(i-1, j), T(i+1, j)), \\ T_2 &= \min(T(i, j-1), T(i, j+1)). \end{aligned} \tag{3}$$

Merging equations 2 and 3, and then squaring the left and right sides of the resulting equation, we can obtain the discrete form of Equation 2:

$$\left(\frac{T(i, j) - T_1}{\Delta x}\right)^2 + \left(\frac{T(i, j) - T_2}{\Delta y}\right)^2 = \frac{1}{W(i, j)^2}. \tag{4}$$

Solving Equation 4 can obtain the solution of the Eikonal equation 2:

$$T(i, j) = \frac{T_1 + T_2}{2} + \frac{1}{2} \sqrt{\frac{2}{W(i, j)^2} - (T_1 - T_2)^2}. \tag{5}$$

The fast marching method classifies the grids in the grid map into three categories: (1) Frozen: the  $T$  values in such grids are no longer changed; (2) Unknown: the  $T$  values in such grids have not been calculated yet; (3) Narrow: This type of the grid is between Frozen and Unknown. The flow chart of the fast marching method is described in Algorithm 2. Firstly, all the grids will be classified as the Unknown class and the  $T$  value will be set to infinity. Secondly, the  $T$  value of the algorithm's initial source point is set to 0 and the grids are classified as the Narrow class. After that, the looping part of the program begins to be performed. The grid whose  $T$  value is minimum in the Narrow class is classified as the Frozen class, and the  $T$  value of its adjacent grids that are not in the Frozen class are solved by using Equation 5. If the adjacent grid belongs to the Unknown class, it is classified as the Narrow class. The execution of the looping part will not stop until the spread of the grid map is completed.

---

### Algorithm 2 Fast Marching Algorithm

---

```

1 FMM( $x_s, X, T, W$ )
2 Initialization:
3 Unknown =  $X$ , Narrow =  $\emptyset$ , Frozen =  $\emptyset$ 
4  $T_i = \infty \quad \forall X_i \in X$ 
5 for  $X_i \in X_s$  do
6    $T_i = 0$ 
7   Unknown = Unknown  $\setminus X_i$ 
8   Narrow = Narrow  $\cup X_i$ 
9 end
10 propagation:
11 while Narrow  $\neq \emptyset$  do
12    $X_{min} = \arg \min_{X_i \in \text{Narrow}} \{T_i\}$ 
13 end
14 for  $X_i \in (\text{Near}(X_{min}) \cap X_i \notin \text{Frozen})$  do
15    $T = \text{SolveEikonal}(X_i, T, W)$ 
16   if  $T < T_i$  then
17      $T_i = \tilde{T}$ 
18   end
19   if  $X_i \in \text{Unknown}$  then
20     Narrow = Narrow  $\cup X_i$ 
21     Unknown = Unknown  $\setminus X_i$ 
22   end
23 end
24 Narrow = Narrow  $\setminus X_{min}$ 
25 Frozen = Frozen  $\cup X_{min}$ 
26 return  $T$ ;

```

---

The FMM algorithm can be thought of as a Dijkstra algorithm. But different from the traditional Dijkstra algorithm, the FMM algorithm can proceed along the diagonal direction instead of along the grids. Figure 5(a) shows the four-neighbor propagation results of the Dijkstra in an unobstructed environment, and Figure 5(b) shows the four-neighbor propagation results of the FMM algorithm, where the velocity  $W$  is constant. It can be seen from Figure 5 that the distance value of each grid to the start grid

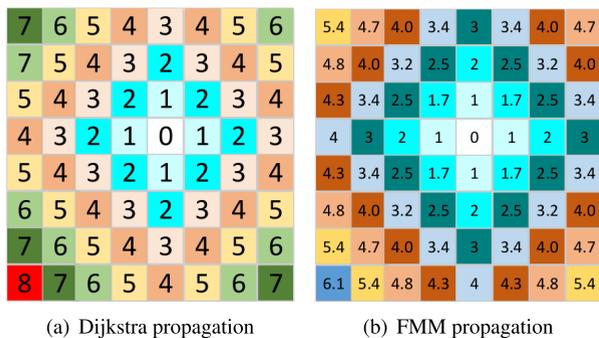


FIGURE 5. Dijkstra vs FMM.

calculated by Dijkstra is the same as Manhattan distance, and the distance calculated by the FMM algorithm is closer to the Euclidean distance. Therefore, the direction of the gradient vector calculated based on the target potential field created by using the FMM algorithm is closer to the direction pointing to the initial source point whose  $T$  value is set to 0.

In the person-following navigation system, the laser sensor is used to detect obstacles in environments, and a robot-centric rolling grid map is constructed based on laser scanning data, as shown in Figure 6(a). To evaluate the arrive time  $T$  of each point in grid map, the FMM algorithm is used to create a target potential field on the rolling grid map. When calculating the arrive time  $T$ , the grid propagation velocity  $W$  in the area where obstacle is located is set to 0, so that the interface propagation cannot pass through the obstacle. Meanwhile, the propagation velocity  $W$  in areas closer to the obstacle are set according to the piecewise function 6. The velocity field containing the speed of each point in the grid map is shown in Figure 6(b).

$$W(x) = \begin{cases} e^{-\alpha/d(x,y)^2} & d(x,y) < d \\ 1 & d(x,y) \geq d, \end{cases} \quad (6)$$

where  $\alpha$  is the obstacle influence factor,  $d(x,y)$  is the distance of the point  $(x,y)$  to the nearest obstacle on the map,

and  $d$  is the threshold. In the grid map, the FMM algorithm is used to propagate the interface by the setting propagation speed, and finally obtains the target potential field, as shown in Figure 6(c). The target potential field is differentiated to generate a direction gradient field, as shown in Figure 7, which provides the reference direction for robot moving.

**B. IMPROVED DWA ALGORITHM**

There are many local path planning methods used in the field of mobile robots. The DWA algorithm is a classic online local path planning method that works well in dynamic and uncertain environments. The method mainly samples multiple sets of velocities in the velocity space  $(v, w)$  and simulates the trajectory of the robot within a certain period of time. After obtaining multiple sets of trajectories, these trajectories will be evaluated, and the velocity of the optimal trajectory is selected to drive the robot to move. The highlight of this algorithm reflects in the dynamic, which means that the velocity sampling space is limited to a feasible dynamic range according to the acceleration and deceleration performance of the mobile robot. Two-wheel differential-driven mobile robots can only move forward and backward or rotate. Considering two consecutive moments, the robot moves a very short distance, so the trajectory between the adjacent moments can be regarded as a straight line. Assuming that the robot's pose at time  $t$  is  $(x_t, y_t, \theta_t)$ , then the pose at time  $t + 1$  can be calculated according to Equation 7. By sampling multiple sets of velocities in the velocity space, the expected poses of the robot at different velocities are calculated, and then the simulated trajectories of the robot could be generated, as shown Figure 8.

$$\begin{aligned} x_{t+1} &= x_t + v_t * \Delta t * \cos \theta_t \\ y_{t+1} &= y_t + v_t * \Delta t * \sin \theta_t \\ \theta_{t+1} &= \theta_t + w_t * \Delta t \end{aligned} \quad (7)$$

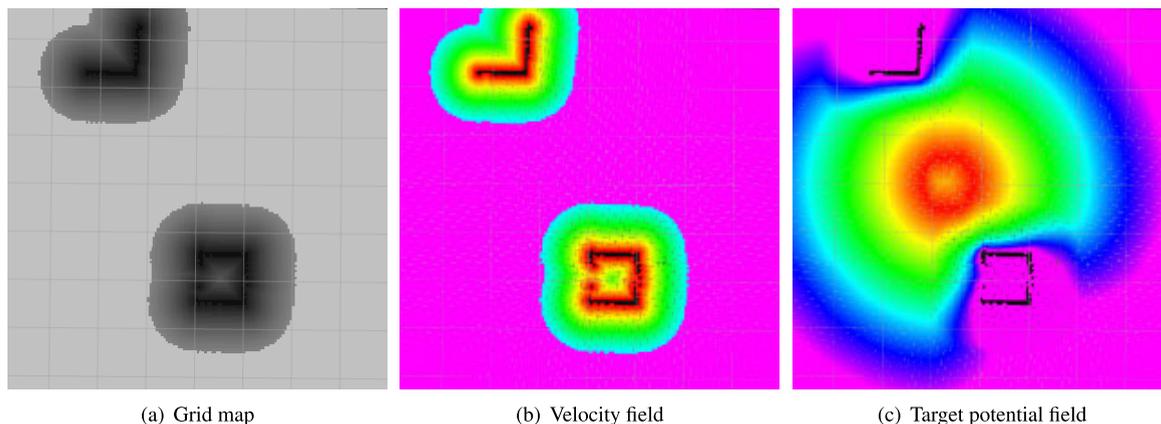
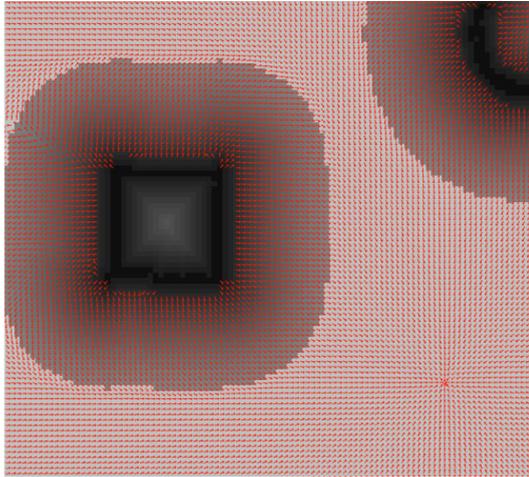
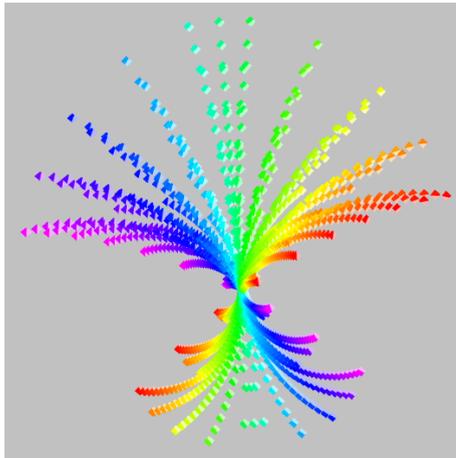


FIGURE 6. Target potential field generation. (a) Grid map. (b) Velocity field. The velocity field is constructed by using velocity equation based on the grid map. (c) Target potential field. The target potential field is created by using the fast marching method based on the grid propagation speed provided by the velocity field.



**FIGURE 7.** Direction gradient field. The direction gradient field is created based on the target potential field. The red arrows are the reference direction provided by the direction gradient field for the robot to adjust the heading.



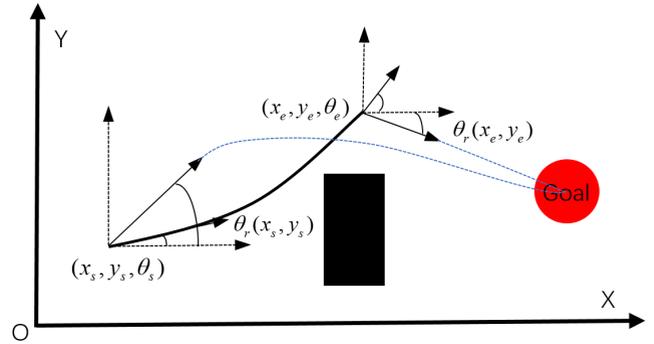
**FIGURE 8.** Sampling trajectory in collision-free environments.

In order to find the optimal trajectory, a series of evaluation functions are used to select the most suitable trajectory from all sampling trajectories. To evaluate each candidate trajectory, we introduce the target potential field and direction gradient field into our evaluation function. In environments with obstacles, FMM algorithm is used to establish a target potential field, where  $T(x, y)$  represents the time needed to arrive target position from current position  $(x, y)$ . By differentiating the target potential field, we can get a direction gradient field, and therefore get the reference direction  $\theta(x, y)$  for the robot moving from current position to the target position.

As shown in Figure 9, the start point of the trajectory is  $(x_s, y_s, \theta_s)$  and the end point of the trajectory is  $(x_e, y_e, \theta_e)$ . To select the best trajectory, following evaluation steps are considered.

### 1) MOTION VALIDITY EVALUATION

In order to evaluate the effectiveness of moving from current position to the target, a motion validity function as



**FIGURE 9.** Motion model.

Equation 8 is constructed.

$$goal\_cost = \left(2 - e^{-\beta|\theta_e - \theta_r(x_e, y_e)|}\right) * \frac{T(x_e, y_e)}{\sum T(x_e, y_e)} \quad (8)$$

where  $goal\_cost$  is the trajectory validity cost, which is used to evaluate whether the trajectory moves to a position where the  $T$  value is low.  $\sum T(x_e, y_e)$  is the sum of the end point arrival time  $T(x_e, y_e)$  of all trajectories.  $\beta$  is the influence factor of the robot's heading. When the difference between the end point orientation  $\theta_e$  of the trajectory and the reference direction  $\theta_r(x_e, y_e)$  provided by the direction gradient field increases, the  $goal\_cost$  will become larger, so that the trajectory closer to the reference direction will be selected.

### 2) HEADING VALIDITY EVALUATION

As you can see, the motion validity evaluation function Equation 8, has the same effect as the trajectory evaluation function of the traditional DWA algorithm. But only considering the use of an evaluation function similar to Equation 8 to select the optimal trajectory, the robot is easy to get into trouble and eventually cannot escape from obstacles. Therefore, in this paper we also introduce the evaluation of the robot's heading into the evaluation function so that the robot can quickly adjust its heading to avoid obstacles ahead when approaching the obstacles, especially enable to move back when it is too close to an obstacle. The evaluation of the heading of the robot is expressed by Equation 9.

$$angel\_cost = \left(\frac{1}{T(x_s, y_s)} + e^{-\alpha d(x_s, y_s)^2}\right) * \frac{|\theta_e - \theta_r(x_e, y_e)|}{\sum |\theta_e - \theta_r(x_e, y_e)|} \quad (9)$$

where  $T(x_s, y_s)$  is the arrival time of the start point of a trajectory. When  $T(x_s, y_s)$  is small, the  $angel\_cost$  increases rapidly, which makes the robot more likely to choose a trajectory that is similar to the reference direction provided by the direction gradient field.  $d(x_s, y_s)$  is the distance between the start point of the trajectory and the nearest obstacle. When  $d(x_s, y_s)$  is small, the robot quickly adjusts the heading to the reference direction to avoid getting too close to the obstacles.

Finally, to select an optimal trajectory from all sampling trajectories, the sum of the weighted  $goal\_cost$  and the weighted  $angel\_cost$ , as shown in Equation 10, is used to

evaluate the merits and demerits of each trajectory.

$$total\_cost = \sigma goal\_cost + \lambda angle\_cost \quad (10)$$

where  $\sigma$  and  $\lambda$  are the weight of motion validity evaluation and heading validity evaluation respectively.

## VI. EXPERIMENTS AND RESULTS

As you know, the person-following method proposed in this paper is mainly composed of two parts: the first part is the target person tracking based on AOA tag and 2D laser; the second part is the target following navigation based on the FMM-DWA algorithm. In order to verify the feasibility and robustness of the person following system, the experiment is also divided into the two parts in real and/or simulated environments. The real experiment mainly relies on the laboratory devices: a laptop (cpu: i5, memory 8G), a two-wheel differential-drive mobile robot, an R2000 2D laser range finder from Pepperl+Fuchs, and an AOA sensor from Yaoguang Company. All our algorithms are implemented using C++ and ROS framework on Ubuntu LST16.04. The simulation experiment of obstacle avoidance was carried out using the Gazebo simulator.

### A. TARGET TRACKING EXPERIMENT

In the person tracking experiment, we carried out two tests to validate the filtering performance of our proposed tracking method. In the first test, there is no obstacles between the robot and target person. In the second test, there are some obstacles moving between the robot and the target.

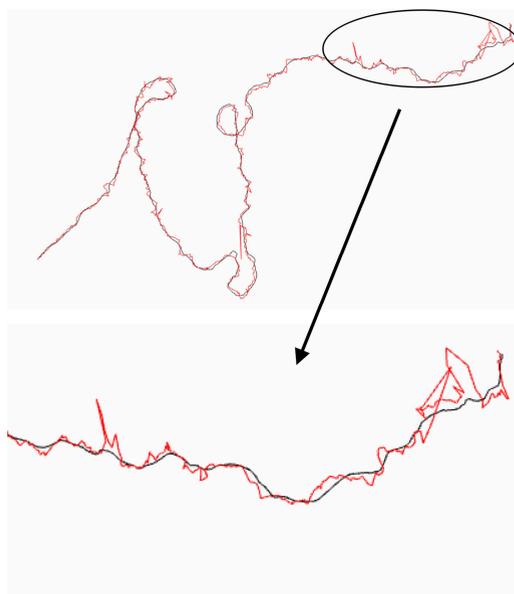
#### 1) NON-OCCLUDED ENVIRONMENT

First, we compare the tracking trajectory from the Kalman Filter and the AOA tracking system. The results are shown in Figure 10. As can be seen from the figure, when only the AOA equipment is used, the position estimation of the target person is inaccurate, and sometimes there is a sharp jump. However, the trajectory obtained by Kalman Filter is much smoother, which eliminates some jumps in the AOA measurement.

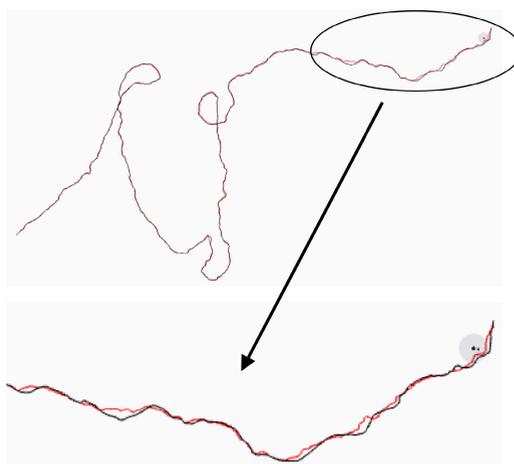
Then, the laser tracking results are also compared to the results of Kalman Filter. It can be seen from Figure 11 that the trajectory obtained by laser tracking is similar to the results of Kalman Filter, but the trajectory obtained using Kalman Filter is also smoother.

#### 2) OCCLUDED ENVIRONMENT

In the case of no occlusion, the position of the person detected by the laser sensor is relatively reliable. However, in the case of occlusion, relying solely on the laser sensor may cause tracking failure. For example, in the case of Figure 12(a), the id of the person tracked by a laser scanner is 4. However, as shown in Figure 12(b), since the target person were blocked by obstacles for a while, the person tracking failed. In Figure 12(c), when the tracked person disappears for a while and is tracked again by the laser scanner, the id of the tracked person is reassigned to 51 at this time. This is

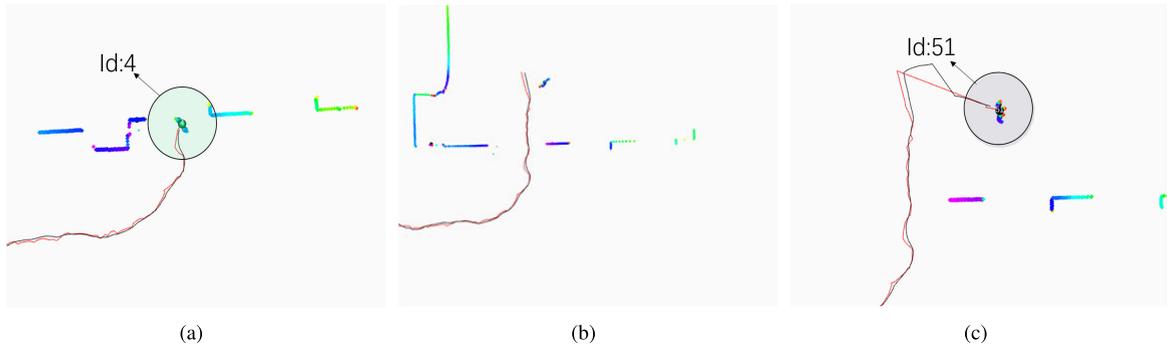


**FIGURE 10.** AOA tracking trajectory and Kalman filter trajectory. The red trajectory is obtained by using the AOA sensor, and the black trajectory is obtained by using the Kalman Filter.



**FIGURE 11.** Laser tracking trajectory and Kalman filter trajectory. The red trajectory is obtained by using the laser scanner and the black trajectory is obtained by using the Kalman Filter.

because it is very difficult to re-identify a person using 2D laser scanner. If only a laser range finder, the robot will not be able to effectively track the target person when the target is occluded. Even if the target person is tracked again by the laser, it will be assigned a new target id. Similarly, If only a camera is used, it will also fail when occlusion occurs, especially in environments with many people wearing same uniform. In Figure 13(a), the camera recognizes that the target id is 0. When the half body of the target person is blocked by the wall, the recognition will fail, as shown in Figure 13(b). At this time, neither the laser sensor nor the camera sensor can effectively track the target person, and the target will lose. In Figure 13(c), when the target person re-enters the camera field of view, the id of the tracked person is also reassigned. This is because the camera-based



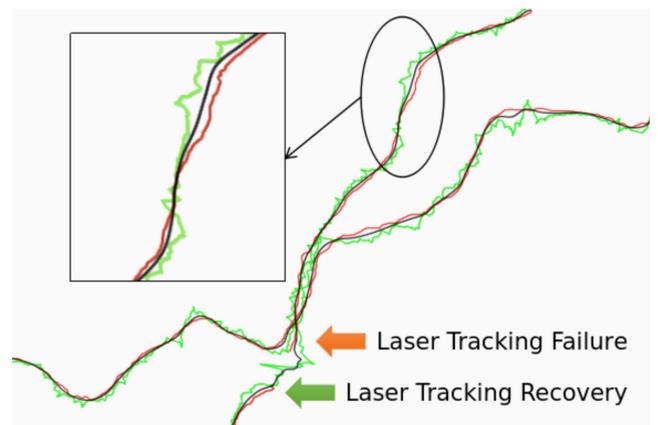
**FIGURE 12.** These three picture show the situation of using only the laser scanner to detect and track a person. (a) The id of the first detected person is assigned to 4. (b) The person tracking failed. (c) After the target disappears for a while and is tracked again, the target person's id is reassigned to 51.



**FIGURE 13.** These three pictures show the situation of using only the camera to identify and track a person. A labeled yellow box is used to mark the recognized person. (a) The id of the first recognized person is assigned as 0. (b) The person tracking failed. (c) After the target disappears for a while and is recognized again, the target person's id is reassigned to 1.

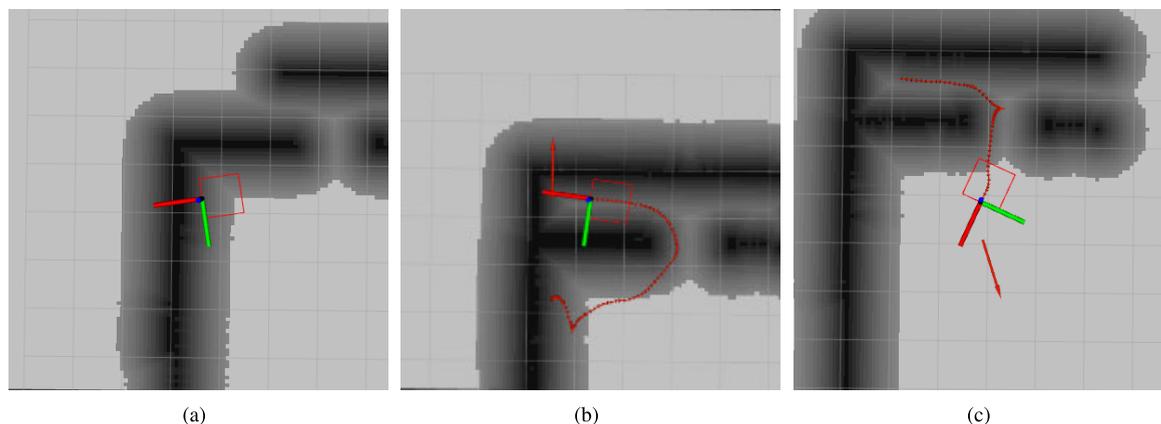
pedestrian re-identification method is also not robust enough. If the camera is mounted on a moving robot, it will have a lower probability of successfully re-identifying the target person. Even we combine cameras and laser scanners, it is also very difficult to solve occlusion problems always suffering in practical environments since they are both line-of-sight sensors.

Fortunately, the AOA tag is very helpful to solve the problem that the laser scanner cannot re-identify the target person. This is because the AOA device can provide a unique id to help the laser re-identify the target. Besides, when occlusion occurs, the AOA sensor can still provide a target to lead the robot toward the target person. However, the AOA cannot be used alone, because our low-cost AOA sensor has two problems, namely the signal strength is relatively weak and the signal value fluctuates greatly. If only the AOA sensor is used to provide a target, our robot motion will be very unstable. Therefore, it is necessary to fuse the two measurements from AOA sensor and laser scanner. As shown in Figure 14, the light green and red trajectories are the AOA tracking trajectory and the laser tracking trajectory respectively. There are many jumps in the AOA tracking trajectory. The laser tracking trajectory is much more stable than that of the AOA. However, laser scanner is a line-of-sight sensor, which cannot track the target when severe occlusion happens. The black trajectory is the target person's trajectory obtained by using



**FIGURE 14.** Kalman filter tracking in occluded environments. The light green trajectory is the AOA tracking trajectory, and sometimes there is a sharp jump. The red trajectory is the laser tracking trajectory, and the black trajectory is the target person's trajectory obtained by using the Kalman Filter. The laser scanner failed to track the target person in the orange arrow area, and the person disappeared for a while. The target person was tracked again by the laser scanner in the green arrow area. Unfortunately, the id of the tracked person changed. With the AOA re-initialization, our method could track the person again.

Kalman Filter to fuse the person tracking information from both AOA and laser scanner. It can be seen from the result that this fusion can achieve a stable and smooth tracking trajectory. In addition, since the field-of-view of the laser scanner for human legs detection is blocked in the orange



**FIGURE 15.** These three pictures show the situation of obstacle avoidance in simulated environments. The red arrow represents the target pose, the red box represents the robot's body, the red trajectory is the robot's track, the black part is the obstacle, and the gray areas are generated by the obstacle inflating with a given radius. (a) The robot faces the obstacle to create a situation of the robot getting into trouble. (b) The first goal was sent to test the obstacle avoidance performance of the long rectangular robot when coming into a narrow corridor and the ability of getting out of trouble. (c) The long rectangular robot will escape from the narrow corridor.

arrow area, the laser scanner failed to track the target person. As you can see, there is a gap in the red trajectory between the orange and green arrow. But our tracking system could still obtain a full tracking trajectory. And after the target person disappeared for a period of time, the laser scanner successfully tracked the person again in green arrow area with AOA re-initialization. We get a smooth person tracking trajectory again. Therefore, our tracking system is also more robust than using only line-of-sight sensors.

## B. AUTONOMOUS FOLLOWING EXPERIMENT

The autonomous following experiment is divided into three parts: obstacle avoidance experiment in simulated environments, person following experiment in collision-free environments and person following experiment in environments with static and dynamic obstacles.

### 1) OBSTACLE AVOIDANCE EXPERIMENT IN SIMULATED ENVIRONMENTS

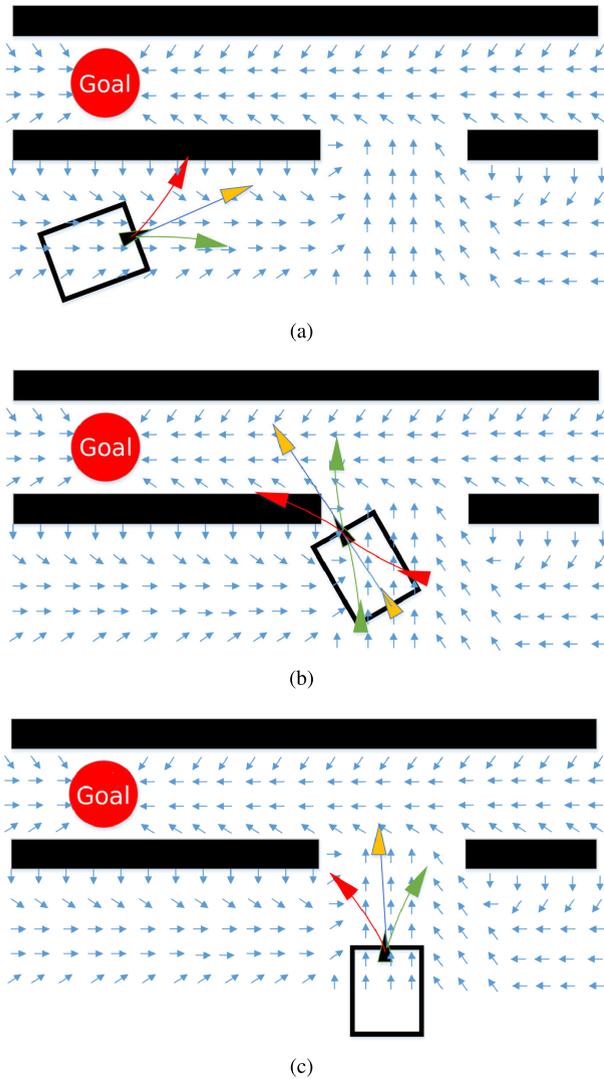
The simulation experiment was carried out on the ASUS K450v laptop running Ubuntu LST16.04 and ROS. The C++ programming language was used to implement all algorithms. The 3D Gazebo simulator was used for robot and environment simulation. In this experiment, we want to test obstacle avoidance ability and the ability of escaping from local minimum of the proposed method. Videos demonstrating the performance of our approach can be found at: <https://youtu.be/czzNazlj0vk>.

At the beginning, the robot faces the obstacle as shown in Figure 15(a). Then, a goal is sent at the end of the corridor, and the robot moves toward the target point and generates a red trajectory as shown in Figure 15(b). In Figure 15(c), another goal is sent. Since the corridor is very narrow and our robot is a long rectangle, the robot cannot turn in the narrow corridor. For such kind of scenario, the traditional DWA will fail. However, for our method, the robot will move backwards to a spacious place and then adjust the heading.

The simulation experiment verified that the navigation algorithm has a better obstacle avoidance ability.

Compared with the traditional DWA algorithm, the improved DWA algorithm proposed in this paper introduces the evaluation of the robot's heading, so as to avoid falling into the local optimum. The traditional DWA calculates the distance from the endpoint of each sampling trajectory to the target point and selects the trajectory closest to the target point. As shown in Figure 16(a), the red trajectory is too close to obstacles, so its cost is the largest of the three trajectories. The yellow trajectory is closer to the target point than the green trajectory, thus the yellow trajectory has a lower cost and will be selected. However, when the robot moves to the end of the yellow trajectory, the three forward sampling trajectories hit obstacles. At this time, the robot get into trouble and cannot escape from obstacles. The method proposed in this paper introduces the robot's heading evaluation to avoid getting into the local optimum. In Figure 16(a), the difference between the endpoint's direction of the yellow trajectory and the reference direction provided by the direction gradient field is large, so the cost of the trajectory is big. The green trajectory is far from the target point, but the endpoint's direction of the trajectory is similar to the reference direction in the gradient field, so the direction angle cost of green trajectory is less than that of the yellow trajectory. Combining the distance cost and the heading cost, the green trajectory will be selected in our method.

When the robot fall into the local optimum, all sampling trajectories of the robot will collide with obstacles. For DWA method, the robot will be get stuck and stop there. However, the method proposed in this paper can also escape from this situation. As shown in Figure 16(b), since the robot is too close to obstacles, the three forward sampling trajectories will hit the obstacle. In this case, the traditional DWA algorithm considers that the current position is closer to the target point and the potential field value is smaller, so the backward trajectory will not be selected and the robot will be get stuck

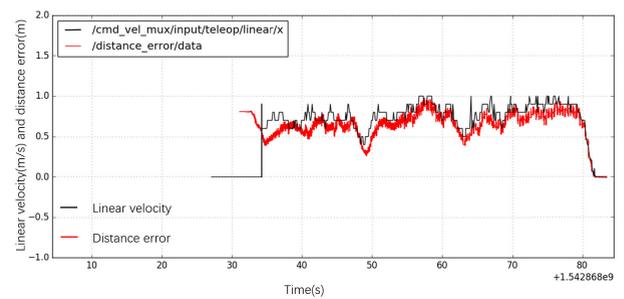


**FIGURE 16.** Schematic diagram of the improved DWA algorithm. The top-left red circle is the target point, the black-filled rectangular blocks represent obstacles, and the blue arrows distributed in the barrier-free area are the reference directions provided by the direction gradient field for the robot to adjust the heading. The long red, yellow and green arrows represent the three sampling trajectories generated by DWA, and the triangle at the end of the trajectory indicates the direction of the end of the sampling trajectory.

with the obstacle. However, in our method, the endpoint's direction of the forward sampling trajectory is very different from the reference direction in the gradient field, so the cost of the forward sampling trajectory will be very large. The direction of the green trajectory in the backward direction is closer to the reference direction in the gradient field, so the cost of the trajectory is smaller, and the robot will choose a backward trajectory to get out of trouble. When the heading of the robot is adjusted to be closer to the reference direction in the gradient field, as shown in Figure 16(c), the robot will move forward again to approach the target point.

## 2) PERSON FOLLOWING EXPERIMENT IN COLLISION-FREE ENVIRONMENTS

Different from traditional navigation algorithms, in addition to the obstacle avoidance ability, the robot's following algorithm proposed in this paper should also be able to respond quickly to the moving target person and maintain a constant distance from the tracked person. The real experiment was conducted in an open and collision-free indoor environment. The person will walk arbitrarily to test the velocity response of the robot, and the velocity response curve is as shown in Figure 17. In this experiment, the expected distance between the robot and the target person is set to 0.8m. However, due to factors such as the robot's motion inertia and the continuous movement of the target person, the actual distance between the person and the robot is usually between 0.7m and 1.6m. Besides, the robot can move at a velocity of 1m/s and quickly follow the target person.



**FIGURE 17.** Linear velocity response curve and distance error curve. The black curve is the linear velocity response curve of robot during the person following, and the red curve is the distance error between the robot and the target person.

It can be seen from Figure 18 that although the trajectory of the moving person still fluctuates, the trajectory of the robot is very smooth. When this experiment was completed, the target person walked 45m, and the robot only moved 36m. Compared with the track length of the target person, the track of the robot is shortened by 20%.

In person-following, the smoothness of the robot trajectory is an important indicator to measure whether it is a good follow movement. In this paper, two indicators are used to measure the quality of the following motion. The first indicator is the acceleration of the robot, and the second indicator is the turning radius of the robot. If the acceleration is too big, it can be considered that the motion is not smooth. If the robot's turning radius is extremely small, the trajectory is also considered as not smooth. During the 36-meter following, a total of 506 motion commands were generated in our method. The distribution of acceleration and turning radius in 506 motion commands is shown in Table 1 and Table 2. It can be seen from the table that the robot's accelerations are mostly smaller than  $0.2m/s^2$ , and only two commands are bigger than  $4m/s^2$ . Besides, most of turning radii are bigger than 1m. Therefore, our method could always generate a smooth following trajectory.

TABLE 1. Acceleration distribution.

Acceleration ( $m^2/s$ )	$a > 4$	$4 > a > 2$	$2 > a > 1$	$1 > a > 0.2$	$a < 0.2$
Number Times	2	36	32	0	436

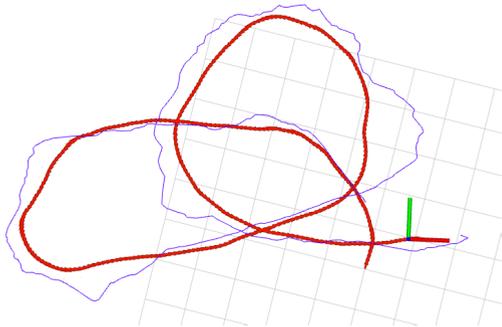


FIGURE 18. Following trajectory in collision-free environments. The grid cell size is set to 1m. The red trajectory is the moving trajectory of the robot, and the purple trajectory is the walking trajectory of the tracked person.

TABLE 2. Turning radius distribution.

Turning Radius ( $m$ )	$\infty$	$r > 5$	$5 > r > 1$	$1 > r > 0.5$	$r < 0.5$
Number Times	143	69	253	40	1

### 3) PERSON FOLLOWING IN ENVIRONMENTS WITH OBSTACLES

For the person following experiment in environments with obstacles, we built an indoor environment with many static and dynamic obstacles as shown in Figure 19(a). The experimental environment is a long corridor. There are many scattered square obstacles in the corridor. The person will pass through all the obstacles in the corridor, then bypass a pillar at the end of the corridor and finally pass through all the square obstacles again and return to the original start point. During the movement, the obstacles and pillar will block the view of the laser scanner, which will make the tracked person disappear for a short while. Besides, the robot also need to avoid all the static and dynamic obstacles (several arbitrarily walking people) while following the tracked person, as shown in Figure 19(b). In this experiment, videos demonstrating the performance of our approach can be found at: <https://youtu.be/0-dSx782yB0>.

In order to validate the performance of the proposed method, our following algorithm is compared to the MPEPC [36] algorithm. In both tests, the walking path of the target person will be kept as consistent as possible. The control frequency of the controller is set as 10Hz in both experiments, and the maximum velocity of the robot is set to 1.5m/s. In Figure 20(a), the trajectory of the robot generated by the FMM-DWA algorithm is shown. As we noted before, in environments with many obstacles, the tracking performance of AOA sensor drops dramatically, which will jump sharply if there is an obstacle between the robot and the target person. Fortunately, we use a Kalman Filter to fuse the tracking data from both AOA and laser scanner, therefore we could get a much smoother tracking trajectory. However,

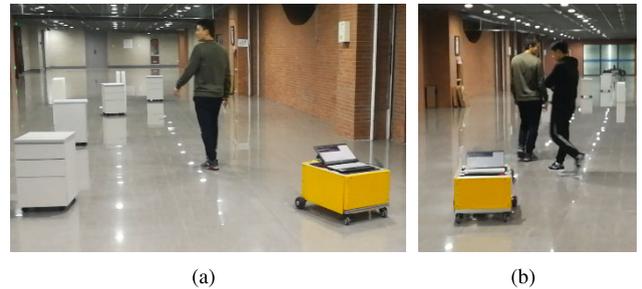


FIGURE 19. Person following in environments with obstacles.

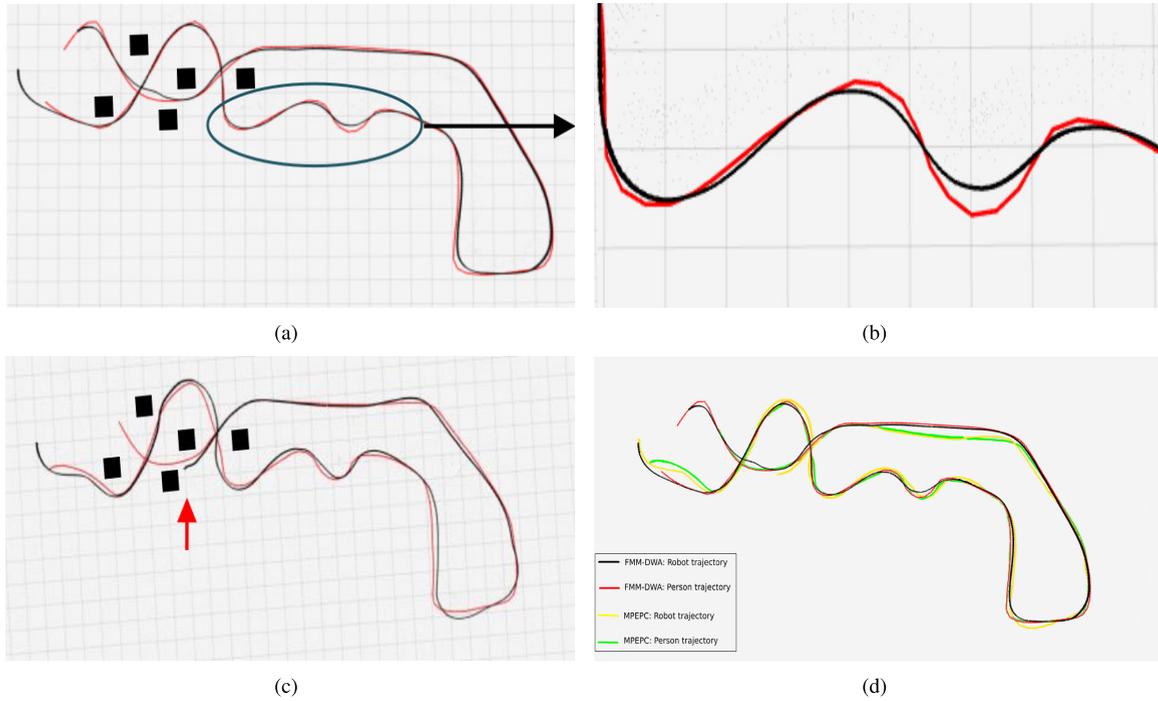
if the AOA data fluctuates frequently, there is still a large error between the position obtained by the Kalman Filter and the position of the real person. For example, in Figure 20(b), when the trajectory of the target person fluctuates frequently for a period of time, the robot may make sharp turns and cannot move very smoothly. In this person-following experiment, our FMM-DWA algorithm successfully avoided static and dynamic obstacles in environments. While for MPEPC algorithm, the robot failed to avoid obstacles that appeared around the red arrow area, as shown in Figure 20(c), where the robot lost its target person.

Figures 21(a) and 21(b) are linear velocity response curves of the FMM-DWA and MPEPC algorithm respectively. It can be seen from the figures that the linear velocity of the FMM-DWA algorithm changes more slowly, and the linear velocity of the MPEPC algorithm has many obvious jumps. In order to quantitatively compare the performance of the two algorithms, the percentage of control commands whose acceleration is greater than  $1 m/s^2$  and the turning radius is less than  $1m$  in the total control command, and the percentage of robot trajectory length and target person trajectory length are calculated as shown in Table 3.

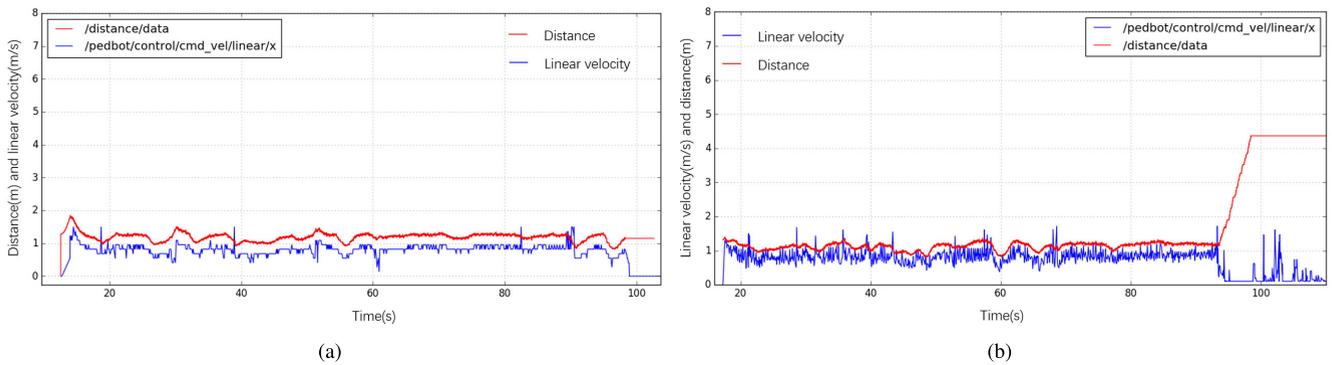
TABLE 3. Comparison of performance between FMM-DWA algorithm and MPEPC algorithm.

Method	$a > 1$	$r < 1$	$S_r/S_h$
FMM-DWA	8%	3%	94%
MPEPC	32%	8%	95%

Table 3 shows that the smoothness of the trajectory generated by FMM-DWA algorithm is obviously better than that generated by MPEPC algorithm, because more than 90% accelerations in FMM-DWA are smaller than  $1m/s^2$  while only 68% in MPEPC. And the control commands of turning radius that are smaller than  $1m$  in MPEPC account for 8% of the total commands, which is 2.6 times of FMM-DWA. Furthermore, the robot's trajectories in FMM-DWA algorithm experiment and MPEPC algorithm experiment are both about 95% of the trajectory of the target person, and there is no obvious difference.



**FIGURE 20.** Following trajectory in environments with obstacles. The grid cell size is set to 1m. The red curve is the trajectory of the target person, the black curve is the robot’s trajectory. The black squares are the static obstacles in the environment. (a) The robot successfully completed following a person by using our following algorithm in this crowded environment. (b) Sharp turn. (c) The person following that is implemented by using MPEPC algorithm is easy to fail in this environment. (d) Trajectory of FMM-DWA and MEPEC.



**FIGURE 21.** Linear speed response curve and distance curve in environments with obstacles. The red curve is the distance between the robot and the target person, and the blue curve is the linear velocity of the robot. (a) The linear velocity of using our algorithm to accomplish person following task changes more slowly. (b) The linear velocity of using the MPEPC algorithm has many obvious jumps.

**VII. CONCLUSION AND FUTURE WORK**

In this paper, we proposed a stable and robust person following method. The target person identification and tracking is achieved by using the combination of a 2D range finder and an Angle-of-Arrival tag to solve the occlusion problem. For the following navigation, firstly, a real-time robot-centric rolling grid map is constructed using the laser data. Secondly, a target potential field is generated by using the fast marching method and a direction gradient field is created based on the target potential field. Finally, Based on the rolling grid map, the target potential field and the direction gradient field, an improved dynamic window algorithm, FMM-DWA, is proposed in this paper to control the robot to move towards the

target. In FMM-DWA, the direction gradient field provides a reference of the heading of the robot, which avoids the robot selecting local optimal trajectory. However, the person-following navigation method proposed in this paper does not predict the motion of dynamic obstacles. In the future, we will further improve the navigation performance by considering dynamic obstacle prediction to make our method faster and smoother.

**REFERENCES**

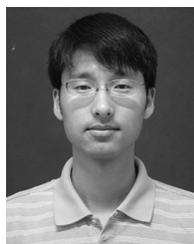
[1] J. Sales, J. V. Martí, R. Marín, E. Cervera, and P. J. Sanz, “CompaRob: The shopping cart assistance robot,” *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 2, Feb. 2016, Art. no. 4781280.

- [2] M. Kenk, M. Hassaballah, and J.-F. Brethé, "Human-aware robot navigation in logistics warehouses," in *Proc. 16th Int. Conf. Inform. Control, Autom. Robot.*, Jul. 2019, vol. 2, pp. 371–378.
- [3] D. Wooden, M. Malchano, K. Blankespoor, A. Howardy, A. A. Rizzi, and M. Raibert, "Autonomous navigation for BigDog," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2010, pp. 4736–4741.
- [4] P. Nikdel, R. Shrestha, and R. Vaughan, "The hands-free push-cart: Autonomous following in front by predicting user trajectory around obstacles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 4548–4554.
- [5] A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *Int. J. Robot. Res.*, vol. 29, no. 14, pp. 1707–1725, Dec. 2010.
- [6] A. Leigh, J. Pineau, N. Olmedo, and H. Zhang, "Person tracking and following with 2D laser scanners," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 726–733.
- [7] A. Beck-Greinwald, A. Zell, K. Bohlmann, S. Buck, and H. Marks, "Autonomous person following with 3D LIDAR in outdoor environment," *J. Automat. Mobile Robot. Intell. Syst.*, vol. 7, no. 2, pp. 24–29, 2013.
- [8] M. Kobilarov, G. Sukhatme, J. Hyams, and P. Batavia, "People tracking and following with mobile robot using an omnidirectional camera and a laser," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2006, pp. 557–562.
- [9] K. Koide and J. Miura, "Identification of a specific person using color, height, and gait features for a person following robot," *Robot. Auto. Syst.*, vol. 84, pp. 76–87, Oct. 2016.
- [10] T. Feng, Y. Yu, L. Wu, Y. Bai, Z. Xiao, and Z. Lu, "A human-tracking robot using ultra wideband technology," *IEEE Access*, vol. 6, pp. 42541–42550, 2018.
- [11] B. Hepp, T. Nageli, and O. Hilliges, "Omni-directional person tracking on a flying robot using occlusion-robust ultra-wideband signals," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 189–194.
- [12] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci. USA*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.
- [13] D. Seror, A. Nissan, R. M. Spira, E. Feigin, R. Udassin, and H. R. Freund, "Comparison of bursting pressure of abdominal wall defects repaired by three conventional techniques," *Amer. Surgeon*, vol. 69, no. 11, pp. 978–980, 2003.
- [14] Y. Sun, L. Sun, and J. Liu, "Human comfort following behavior for service robots," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2016, pp. 649–654.
- [15] W. Chi, J. Wang, and M. Q.-H. Meng, "A gait recognition method for human following in service robots," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 9, pp. 1429–1440, Sep. 2018.
- [16] A. Cosgun, D. A. Florencio, and H. I. Christensen, "Autonomous person following for telepresence robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 4335–4342.
- [17] Y. Sung and W. Chung, "Hierarchical sample-based joint probabilistic data association filter for following human legs using a mobile robot in a cluttered environment," *IEEE Trans. Human-Machine Syst.*, vol. 46, no. 3, pp. 340–349, Jun. 2016.
- [18] E.-J. Jung, B.-J. Yi, and S. Yuta, "Control algorithms for a mobile robot tracking a human in front," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 2411–2416.
- [19] Y. Kobayashi, Y. Kinpara, T. Shibusawa, and Y. Kuno, "Robotic wheelchair based on observations of people using integrated sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 2013–2018.
- [20] C. Granata and P. Bidaud, "A framework for the design of person following behaviors for social mobile robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 4652–4659.
- [21] H.-M. Gross, K. Debes, E. Einhorn, S. Mueller, A. Scheidig, C. Weinrich, A. Bley, and C. Martin, "Mobile robotic rehabilitation assistant for walking and orientation training of stroke patients: A report on work in progress," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2014, pp. 1880–1887.
- [22] J. D. Goyzueta, R. Katz, O. Dumitrescu, H. S. H. Choi, and T. Kahn, "The disappearing kidney: A case of emphysematous pyelonephritis," *Arch. Internal Med.*, vol. 154, no. 22, pp. 2613–2615, 1994.
- [23] R. Liu, G. Huskic, and A. Zell, "Dynamic objects tracking with a mobile robot using passive UHF RFID tags," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2014, pp. 4247–4252.
- [24] R. Gockley, J. Forlizzi, and R. Simmons, "Natural person-following behavior for social robots," in *Proc. ACM/IEEE Int. Conf. Hum.-Robot Interact. (HRI)*, 2007, pp. 17–24.
- [25] S. Hemachandra, T. Kollar, N. Roy, and S. Teller, "Following and interpreting narrated guided tours," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 2574–2579.
- [26] E. Prassler, D. Bank, and B. Kluge, "Motion coordination between a human and a mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 2, Oct. 2002, pp. 1228–1233.
- [27] J. Miura, J. Satake, M. Chiba, Y. Ishikawa, K. Kitajima, and H. Masuzawa, "Development of a person following robot and its experimental evaluation," *Intell. Auton. Syst.*, vol. 11, pp. 89–98, Dec. 2010.
- [28] G. Huskic, S. Buck, L. A. I. Gonzalez, and A. Zell, "Outdoor person following at higher speeds using a skid-steered mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 3433–3438.
- [29] W. Peng, J. Wang, and W. Chen, "Tracking control of human-following robot with sonar sensors," *Adv. Intell. Syst. Comput.*, vol. 531, pp. 301–313, 2017.
- [30] J. Zhang, J. Wang, and W. Chen, "A control system of driver assistance and human following for smart wheelchair," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2014, pp. 1927–1932.
- [31] L. Pang, Y. Zhang, S. Coleman, and H. Cao, "Efficient hybrid-supervised deep reinforcement learning for person following robot," *J. Intell. Robotic Syst.*, vol. 97, no. 2, pp. 299–312, Feb. 2020.
- [32] R. E. Kalman, "A new approach to linear filtering and prediction problems," *J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, Mar. 1960.
- [33] P. Kulaowski, J. Vales-Alonso, E. Egea-López, W. Ludwin, and J. García-Haro, "Angle-of-arrival localization based on antenna arrays for wireless sensor networks," *Comput. Electr. Eng.*, vol. 36, no. 6, pp. 1181–1186, Nov. 2010.
- [34] L. Breiman, "ST4\_method\_random\_forest," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.
- [35] S. Garrido, L. Moreno, and P. U. Lima, "Robot formation motion planning using fast marching," *Robot. Auto. Syst.*, vol. 59, no. 9, pp. 675–683, Sep. 2011.
- [36] J. J. Park and B. Kuipers, "Autonomous person pacing and following with model predictive equilibrium point control," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 1060–1067.



**DAPING JIN** received the B.S. degree in mechanical engineering from Dalian Jiaotong University, Dalian, China, in 2018. He is currently pursuing the M.S. degree with the Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China.

His research interests include autonomous following robot and path planning.



**ZHENG FANG** (Member, IEEE) received the B.S. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from Northeastern University, China, in 2002 and 2006, respectively.

He was a Postdoctoral Research Fellow with Carnegie Mellon University, from 2013 to 2015. He is currently an Associate Professor with the Faculty of Robot Science and Engineering, Northeastern University. His research interests include

visual/laser SLAM, perception, and autonomous navigation of various mobile robots.



**JIEXIN ZENG** received the B.S. degree in autonomous from Northeastern University, Shenyang, China, in 2017, and the M.S. degree from the Faculty of Robot Science and Engineering, Northeastern University, in 2020.

His research interests include optimal control and real-time motion planning for autonomous mobile robot.

...