robotics MDPI

Article

A Pedestrian Avoidance Method Considering Personal Space for a Guide Robot

Yutaka Hiroi [1,*,†] and Akinori Ito [2,†]

[1] Faculty of Robotics and Design, Osaka Institute of Technology, 1-45 Chayamachi, Kita-ku, Osaka 530-8568, Japan

[2] School of Engineering, Tohoku University, 6-6-5 Aramaki aza Aoba, Aoba-ku, Sendai 980-8579, Japan; aito@spcom.ecei.tohoku.ac.jp

* Correspondence: yutaka.hiroi@oit.ac.jp

† These authors contributed equally to this work.

Received: 19 September 2019; Accepted: 6 November 2019; Published: 18 November 2019

Abstract: Many methods have been proposed for avoiding obstacles in robotic systems. However, a robotic system that moves without colliding with obstacles and people, while still being mentally safe to the persons nearby, has not yet been realized. In this paper, we describe the development of a method for a mobile robot to avoid a pedestrian approaching from the front and to pass him/her by while preserving the "public distance" of personal space. We assume a robot that moves along a prerecorded path. When the robot detects a pedestrian using a laser range finder (LRF), it calculates the trajectory to avoid the pedestrian considering their personal space, passes by the pedestrian, and returns to the original trajectory. We introduce a virtual target to control the robot moving along the path, such that it can use the same control strategy as for human-following behavior. We carry out experiments to evaluate the method along three routes, in which the robot functioned without problems. The distance between the robot and the pedestrian was 9.3 m, on average, when the robot started to use avoiding behavior, which is large enough to keep a public distance from a pedestrian. When the robot passed by the pedestrian, the minimum distance between them was 1.19 m, which was large enough for passing safely.

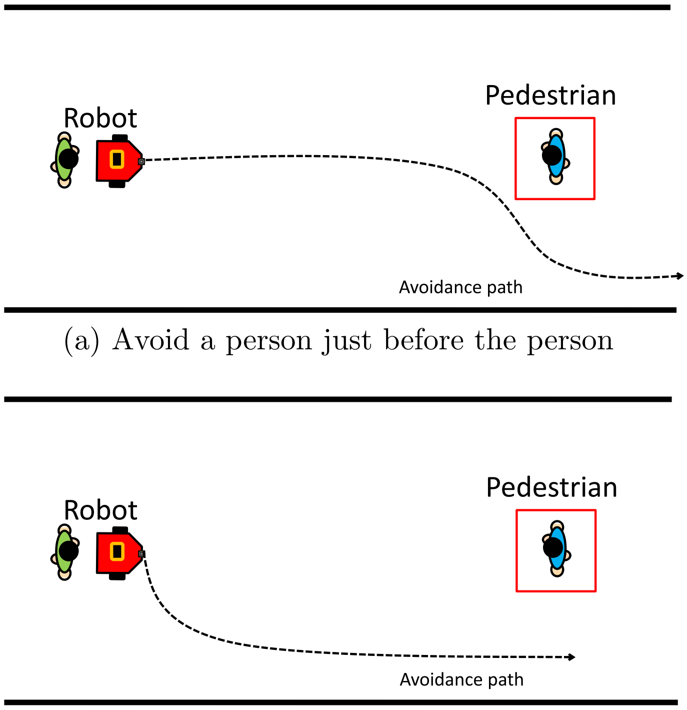
Keywords: mobile robot; pedestrian avoidance; guide robot

1. Introduction

Much effort has been carried out for developing mobile robots that move around humans and support our daily life. These robots are designed with the intention to help people, either in a home environment [1] or in a public space (e.g., a shopping mall) [2]. When a robot moves around humans, the most important requirement is safety for the humans. Thus, there have been a huge number of studies into collision-free robot navigation [3–6]. Not only is avoiding collision with humans essential, but it is also important to avoid frightening people or making them uncomfortable.

In a human–human relationship, it is said that people consider "personal space" [7] when staying a proper distance away from other people. The literature has pointed out that there are four kinds of interpersonal distances: intimate, personal, social, and public distances [8]. The public distance is a distance between people with no social relationship, with a typical length of 3.5–10 m [8]. Thus, several studies have considered how to navigate a robot while considering personal space [9–12].

The robot should keep several meters away from the nearest person, in order to consider the public distance. We believe that it is also crucial, in addition to keeping a public distance from people, for a robot to express that the robot's "intention" is to keep the public distance, such that the pedestrians do not feel anxious about the robot. Figure 1 depicts such a situation. Conventional obstacle- and pedestrian-avoiding methods initiate the avoidance of a pedestrian when the robot is about to collide with the person, as shown in Figure 1a. However, it is desirable for the robot to start avoiding the pedestrian from a distance, to demonstrate to the person nearby that the robot recognizes the oncoming pedestrian and is ready to avoid him/her, as shown in Figure 1b.



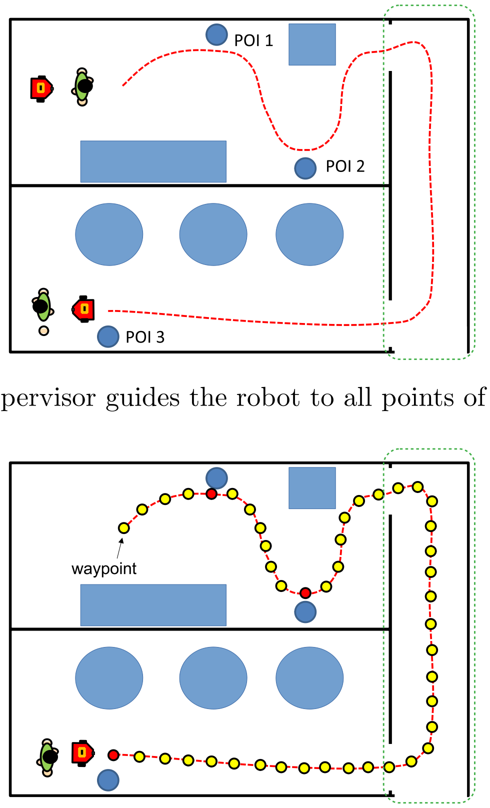
(a) Avoid a person just before the person

(b) Avoid a person keeping public distance

Figure 1. Comparison of pedestrian avoidance methods.

We have been developing a mobile robot, ASAHI [13], which is designed to move around humans and support their daily lives. As it is designed to coexist with humans, it should be not only physically but also mentally safe to the surrounding persons. Thus, we developed the navigation methods of ASAHI while considering personal space.

We consider a situation where the user teaches the robot the path to follow beforehand, after which the robot moves along the learned path [14–17]. The robot is designed to move around an indoor environment, such as rooms and corridors, as a service robot, in order to fulfill several navigation tasks, such as those presented in RoboCup@Home [18,19]. Various environments could be considered for indoor service robots. In this paper, we considered an environment that is typically assumed in RoboCup@Home [18], where there are two or three rooms with furniture and a corridor, along with several places of interest (POIs) for a robot to visit. First, a human supervisor brings the robot through the environment and teaches the robot where the POIs are. Figure 2 depicts the scenario under consideration. As shown in Figure 2a, the robot follows the supervisor and remembers the POIs [15]. In this process, the robot remembers the path and the POIs as points on the remembered path (the waypoints). Figure 2b shows the memorized path, expressed as a set of waypoints. The robot samples waypoints on the trajectory while following the supervisor. In the experiment, the interwaypoint length was 740 mm, considering the footstep length of a human [15]. After remembering the paths, the robot autonomously guides a person by moving along a memorized path to visit a specified place. While guiding a person, the robot observes the distance to the guided person and, if the distance becomes large, the robot stops to wait for the person.



(a) A supervisor guides the robot to all points of interest

(b) The robot remembers the path as a set of waypoints

Figure 2. Scenario of teaching the points of interest and the waypoints.

When a robot moves along a corridor, the robot can keep a public distance from an oncoming person [20]. Thus, we focus on a scene, such as moving along a corridor, where several pedestrians are walking. If we employ a pedestrian-avoiding method based on local information, the robot avoids an oncoming person just before colliding with the person (see Figure 1a). However, considering a public distance, the robot begins avoiding the person while the robot is far from them (Figure 1b), in order to express to the pedestrian that the robot recognizes them and is trying to keep a public distance from him/her.

We consider the following requirements to achieve the tasks of visiting several POIs by moving through corridors and rooms and keeping a public distance from a pedestrian:

(A) The robot should avoid a pedestrian, considering the public distance for mental safety;
(B) The robot should avoid obstacles, based on local information;
(C) The robot should move along the taught path to visit POIs, while avoiding obstacles and pedestrians.

The method proposed by Shiomi et al. [21] realized (A) and (B); Tsukuba Challange [22] aimed to develop a robot that realizes (B) and (C). However, to our knowledge, there has been no study carried out that fulfills the above three requirements.

In this paper, we develop a pedestrian avoidance method for a mobile robot. The proposed method enables the robot to avoid an oncoming pedestrian from a distance while keeping a public distance from the pedestrian. At the same time, we can combine the proposed method with an obstacle avoidance method based on local information, which enables the robot to keep a distance from the pedestrian while avoiding small obstacles in front of the robot.

The paper is organized as follows. We review the related papers in Section 2. In Section 3, we describe the proposed pedestrian avoidance method. The setup and results of the evaluation experiment are described in Section 4. In Section 5, we conclude the paper.

2. Related Works

2.1. Pedestrian Avoidance for Collision-Free Robot Navigation

There have been a vast number of studies into collision-free robot navigation [3–6,23]. As the survey by Hoy et al. [5] pointed out, most collision-free navigation methods have been formalized as an optimization problem, which finds the best path that minimizes a predefined criterion, such as minimum potential [24]. However, if a robot employs an obstacle avoidance method based on local information, such as the potential field method [24] or the dynamic window approach [25], the robot will only avoid the pedestrian by a small margin, such as the situation depicted in Figure 1a. Therefore, we need to develop another method that enables a robot to avoid an oncoming pedestrian from several meters away, even when he/she is too far to collide.

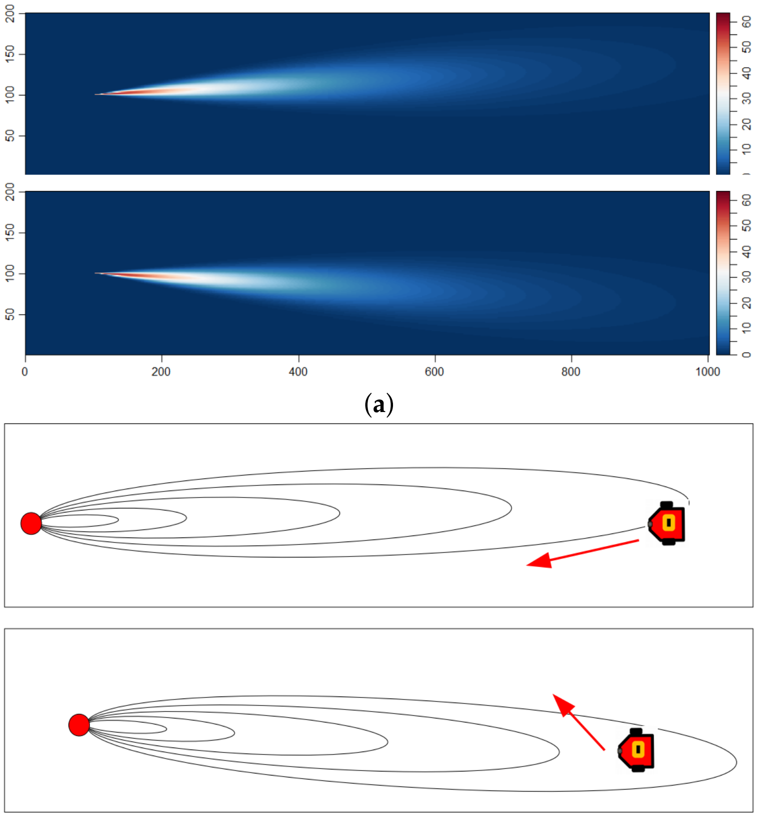
2.2. Robot Navigation Considering the Social Distance

Paccierotti et al. [20] evaluated the proper distance between a robot and a person when the robot passes by. According to their result, it is desirable for the robot to keep more than 3.5 m away from the pedestrian, as shown in Figure 1b. Thus, a number of methods for navigating a robot considering the social distance from humans have been presented [11,21,26–30].

One idea for a robot to keep the public distance is to incorporate the social distance into the optimization constraints [11]. For example, to enable avoiding oncoming pedestrians using the potential field, Hoshino and Maki proposed an anisotropic potential considering the movement direction and speed [31]. They used a potential field based on the von Mises distribution, such that the potential became higher in front of the pedestrian, and the level of the potential changed based on the movement speed of the pedestrian. Similarly, Papadakis et al. [32] proposed a method to express a nonuniform social distance using a potential in a situation where a few people are talking to each other. In principle, it is possible to use such an anisotropic potential where the social force affects a few meters. However, in our situation—where a robot and a pedestrian are almost ten meters away—it is not adequate to use such a potential, because the potential at the robot's position fluctuates significantly according to a slight estimation error of the movement direction of the pedestrian. Figure 3 shows this situation. Figure 3a shows the potential field generated by the von Mises distribution [31], such that the potential reaches 9 m in distance and does not block the side of the pedestrian. From this figure, we can see that a small fluctuation of the pedestrian's movement direction (±2.5 deg, in this case) causes a large change of the potential at a distant point. Figure 3b shows a possible case of fluctuation in the robot's path. Even a slight change or estimation error of the pedestrian's moving path causes the robot to make a zig-zag motion.

This example shows that we want to move a robot based on two different motivations: first, to avoid people or obstacles, such that the obstacles and the robot do not collide; and, second, to avoid a pedestrian from a distance, such that the robot shows the person that the robot intends to keep a social distance from him/her. These two motivations are different, which is why it is difficult to treat them using a single kind of potential.

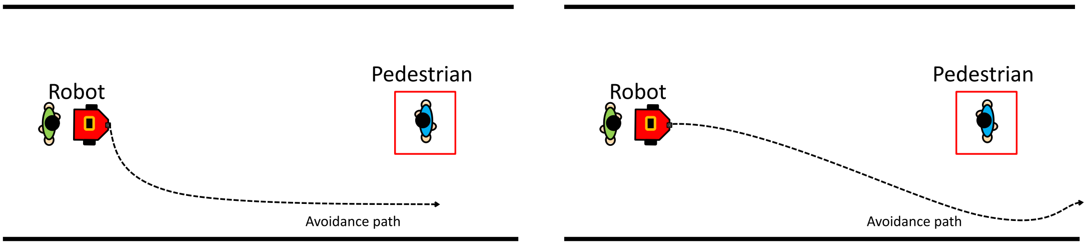
The social force model, proposed in [21], can avoid a pedestrian from 8 m away. However, since this method uses a model of a human, if the pedestrian behaves differently from the prediction of a model (such as stopping suddenly), the method may fail to avoid the pedestrian. Furthermore, the model in this method assumes that the pedestrian is aware of the robot and will try to keep a maximum distance when the robot passes by the pedestrian. Figure 4 shows the difference between the proposed method and the social force model. Comparing Figure 4a,b, the proposed method (a) is safer than (b), as a pedestrian can be aware of the robot's intention to avoid him/her.



(**a**)

(**b**)

Figure 3. Fluctuation of the potential and its effect on the robot's movement path. (**a**) The fluctuation of the potential field caused by direction estimation error. Top: +2.5 [deg], Bottom: −2.5 [deg]. (**b**) Path of a robot distant from a pedestrian fluctuates with a small change of the pedestrian's movement direction.



(**a**) The proposed method

(**b**) Social force model

Figure 4. Difference between the proposed method and the social force model.

A robot needs to discriminate between a human and a nonhuman object, to avoid pedestrians and other obstacles differently. The method proposed by Shiomi et al. [21] does not consider a situation where there are pedestrians and other objects. The methods that consider the tasks (B) and (C) described in the introduction focus on global path planning and obstacle avoidance based on local information. For example, in the method presented by Aotani et al. [33], path planning is formulated as an optimization problem which only considers how to avoid collisions.

Our method uses a human detection and tracking method [15] to locate the pedestrians which are to be avoided. Objects other than the detected pedestrian (including both humans and nonhuman objects) are avoided using local information.

Some studies have used deep learning to design the social behavior of a robot when avoiding pedestrians [27,28,30]. The purpose of those works was to imitate the natural human behavior of avoiding other people. For example, Kim et al. [27] developed a system based on machine learning, which was trained using the human behavior of when a human noticed a robot and avoided it.

However, a human near a robot is not necessarily aware of the robot's avoidance behavior, as the robot does not express the intention of the behavior like humans do. On the other hand, the purpose of our work is to design the robot's behavior, such that the robot does not threaten other pedestrians, which is the same motivation as the "preliminary announcement" of the behavior [34].

To realize a robot that achieves the three requirements (A), (B), and (C), described in Section 1, we combined (A) a human detection and tracking method [15] for detecting the pedestrian to avoid from a distance; (B) an obstacle-avoidance method based on local information [23]; and (C) waypoint-following using the same algorithm as the human-following method [15].

3. Proposed Method

3.1. Overview

As shown in Figure 2, the robot remembers a path as a set of waypoints and visits the waypoints to move along the path. Thus, if the robot needs to avoid a pedestrian or an obstacle, the robot moves out of the path and returns to the path after passing the pedestrian.

Figure 5 shows the situation of moving along waypoints while avoiding a pedestrian and an obstacle. (1) The robot first follows the memorized waypoint using a similar method to the human-following method. We set a "virtual target" in front of the robot, and the robot follows the virtual target as if the virtual target were a human. (2) If the robot finds an oncoming pedestrian in the human detection area (the yellow part, 3.5–10 m front of the robot), the robot moves the waypoints aside to avoid the pedestrian (the red dashed line). After passing by the pedestrian, the robot returns the waypoints to the original path. (3) When a robot finds an obstacle (a nonhuman object or a human nearer than the human detection area), the robot immediately moves to avoid the obstacle, and then returns to the original path.

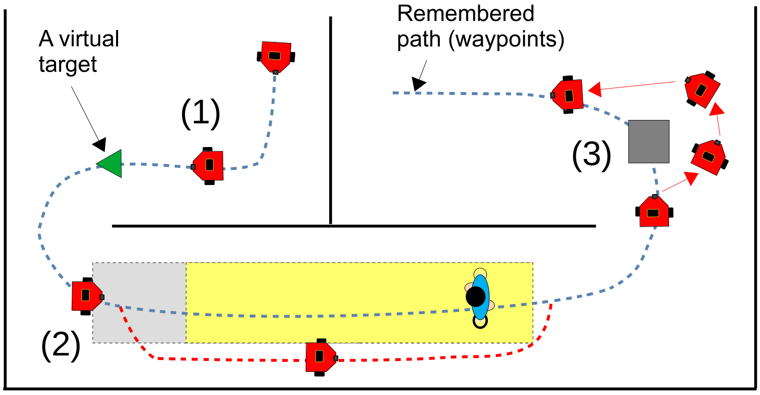


Figure 5. Brief overview of the proposed method.

Figure 6 shows a flowchart of the pedestrian avoidance method. When the robot detects an oncoming pedestrian, the robot determines which direction to move to avoid the pedestrian, based on its distance to the walls. After determining the direction, the robot virtually moves the waypoints in the avoidance direction, and the robot follows the moved waypoints until the pedestrian passes the robot.

Figure 7 presents a detailed explanation of the pedestrian avoidance method, showing the possible situations. First, when the robot detects a pedestrian to avoid (Figure 7a), the robot starts the avoidance motion. When avoiding a pedestrian, the robot measures the breadth of the areas to the left and right of the pedestrian using a laser range finder (LRF) (Figure 7b) and chooses the broader area as the area to move into. Then, the robot calculates the minimum distance to keep from the pedestrian (Figure 7c) and assumes another path parallel to its original path. After the robot recognizes that it has passed by the pedestrian (Figure 7d), it returns to the original path (Figure 7e).

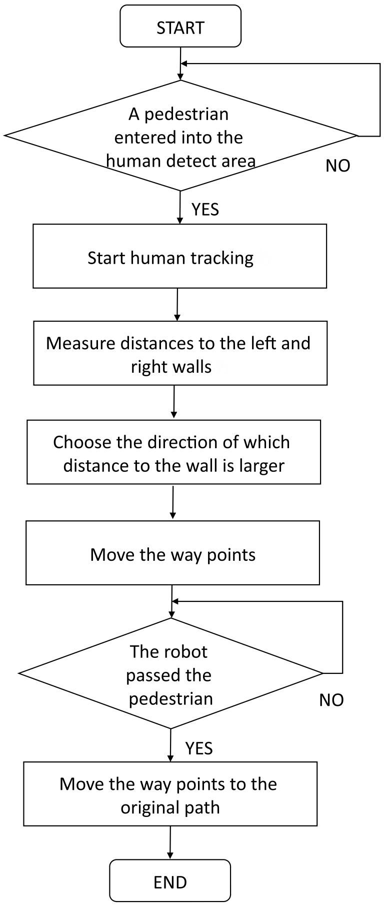
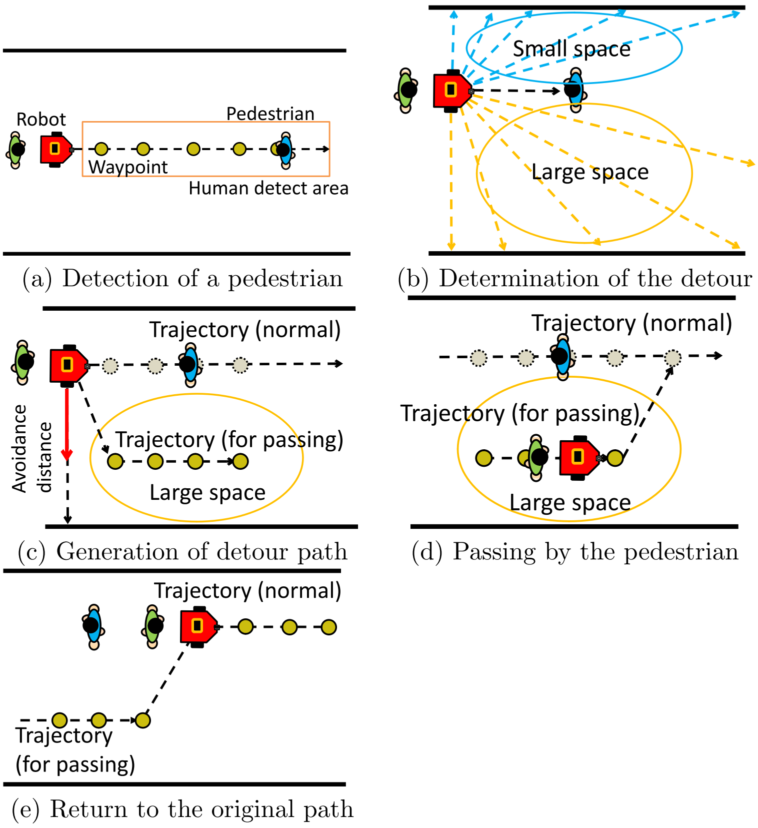

START
A pedestrian entered into the human detect area
NO
YES
Start human tracking
Measure distances to the left and right walls
Choose the direction of which distance to the wall is larger
Move the way points
The robot passed the pedestrian
NO
YES
Move the way points to the original path
END

Figure 6. Flowchart of the proposed method.



(a) Detection of a pedestrian

(b) Determination of the detour

(c) Generation of detour path

(d) Passing by the pedestrian

(e) Return to the original path

Figure 7. Detailed overview of the proposed method.

We need to develop the following six functions to realize the proposed method: (1) Detection of the pedestrian, (2) determination of the direction to move, (3) calculation of the avoidance distance, (4) generation of the avoidance path, (5) detection of passing by the pedestrian, and (6) returning to the original path.

3.2. Avoiding an Oncoming Pedestrian

3.2.1. Detection of a Pedestrian

Let the distance to be kept (i.e., the public distance) be D_{pub}. When the walking speed of the pedestrian is v_p, the robot's moving speed is v_r, and the delay of starting avoidance after recognizing the pedestrian is τ, the human–robot distance to start avoidance is

$$D_{av} = D_{pub} + (v_p + v_r)\tau. \tag{1}$$

For example, when $D_{pub} = 8$ m, $v_p = 1.4$ m/s, $v_r = 0.4$ m/s, and $\tau = 1$ s, the robot needs to start avoiding when the human–robot distance is 9.8 m. If the pedestrian moves to block the robot, the robot either avoids the pedestrian just in front of them or stops.

The method to detect and track the pedestrian is the same as that proposed by Nakamori et al. [35], which uses an LRF. We briefly describe the human detection and tracking method. Using an LRF, we obtain distances from the LRF to the other end in all directions. First, we detect the edge of an object by observing the difference of distances of two contiguous measurement points obtained from the LRF. Then, we determine whether an object (the region between two edges) is a human or not by comparing the width of the object and the typical human body size. If the object is determined to be a human, the center point of the object is regarded as the location of a human. We need to determine the human detection area to exploit this detection method. We set the detection area as a rectangle 900 mm wide and 6500 mm deep, 3500 mm away from the front of the robot, as depicted in Figure 8.

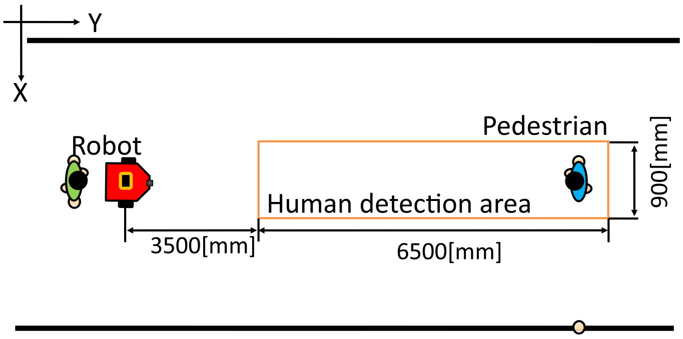


Figure 8. The human detection area.

When the robot detects a pedestrian in the human detection area, the robot starts to track the pedestrian. Let $(x_p(t), y_p(t))$ be the center position of the pedestrian at time t, where the unit of time is based on the scan of the LRF. In the experiment, we used a Hokuyo UTM-30LX, which scans every 25 ms. Thus, in this description, the unit of time was 25 ms. The view angle of the LRF was 270 deg, and the maximum distance of measurement was 30 m.

When the positions of observed objects in two contiguous observations are near enough, those two observations are regarded as an identical object. The threshold for determination was 500 mm; that is, we regarded two observations as belonging to the same object when $|x_p(t) - x_p(t+1)| \leq 500$ mm and $|y_p(t) - y_p(t+1)| \leq 500$ mm.

3.3. Determination of the Direction to Move

After detecting the pedestrian, we need to determine the direction (left or right) to move in order to avoid the pedestrian. To do that, as shown in Figure 7b, we take the average distances to the right and left of the robot and choose the larger direction.

3.3.1. Calculation of the Avoiding Distance

Next, we determine the distance between the original path and the new path for avoiding the pedestrian. Several works have measured the socially acceptable distance between a robot and a pedestrian passing by. Yoda et al. [36] considered 1200 mm to be an appropriate distance. Pacciarini et al. [20] examined three distances (200, 300, and 400 mm), and concluded that 400 mm was the best among those three conditions. In this work, we consider a situation where a robot and a pedestrian pass each other in a corridor. According to the Japanese law on architecture (*Kenchiku Kijun Hou*—Architectural Standard Law), article 119—the width of a corridor of a school (i.e., elementary, junior high, and high school) should be no less than 2.3 m when it has rooms on both sides. In this case, even when the pedestrian walks in the middle of the corridor, the robot can pass by the pedestrian by choosing a path in the middle of the left or right side. Figure 9 shows the size and available path of the robot. We considered the robot to be 400 mm wide [13] and the maximum body width of the pedestrian to be 556 mm, according to the AIST anthropometric database [37]. Considering this situation, we decided that the robot's path should be in the middle of the broader side.

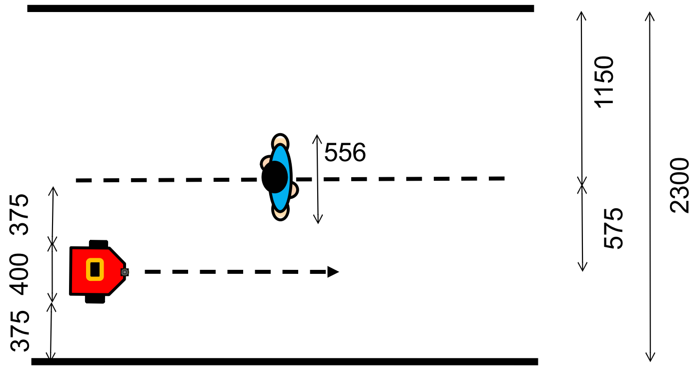


Figure 9. Size of the corridor and available path of the robot.

3.3.2. Generation of the Avoidance Path

The robot follows the original path by moving from a waypoint to the next waypoint. Thus, when avoiding a pedestrian, the robot dynamically generates an avoidance path and follows the generated path. The avoidance path can be generated by shifting the waypoints in front of the robot towards the avoiding side, as shown in Figure 7c.

3.3.3. Detection of Passing by the Pedestrian

After the robot passes by the pedestrian, the robot returns to the original path. Let the coordinates of the robot and the human be (x_r, y_r) and (x_h, y_h), respectively. Then, the distance between the robot and the pedestrian is

$$D_{\mathrm{hr}} = \sqrt{(x_r - x_h)^2 + (y_r - y_h)^2}. \tag{2}$$

The coordinates are shown in Figure 10. The robot determines that it has passed by the pedestrian when $D_{\mathrm{hr}} \geq 500$ mm and $y_r > y_h$.

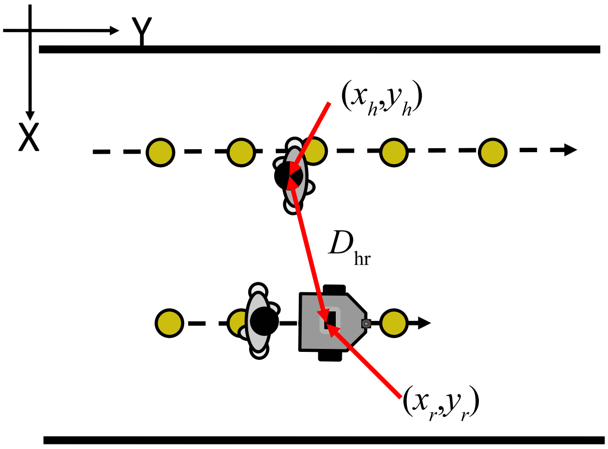


Figure 10. Coordinates of the robot and the pedestrian.

3.3.4. Returning to the Original Path

After the robot has passed by the pedestrian, the robot returns to the original path by changing the next waypoint from that of the avoidance path to the corresponding point of the original path, as shown in Figure 7e.

3.4. Following the Waypoints

3.4.1. The Human Following Method

In this section, we explain how the robot follows the waypoints. As explained in Section 1, the robot first follows a person (the supervisor) to learn the POIs, as well as the path to visit the POIs. The method to follow the person is based on that proposed by Sakai et al. [15]. We used the same method to follow the waypoints. Figure 11 shows the parameters for controlling the robot to follow the target person, where R is the distance between the center points of the robot and the human, θ_h is the angle between the frontal direction of the robot and the human, and D_stop is the distance from the person at which the robot should stop. Then, the speed of the robot (V m/s) is determined as follows:

$$V = \min(V_0, V_\mathrm{min}), \tag{3}$$

$$V_0 = \begin{cases} -V_\mathrm{back} & \text{if } R < D_\mathrm{stop} - D_\mathrm{back} \\ 0 & \text{if } D_\mathrm{stop} - D_\mathrm{back} \leq R < D_\mathrm{stop} \\ K_V(R - D_\mathrm{stop}) & \text{if } D_\mathrm{stop} \leq R. \end{cases} \tag{4}$$

Here, we employed the constants $V_\mathrm{min} = 0.4$ m/s, $V_\mathrm{back} = 0.2$ m/s, and $D_\mathrm{back} = 0.1$ m.

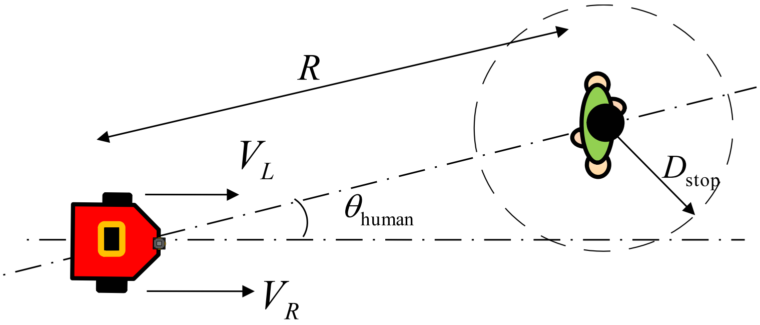


Figure 11. Parameters for human following.

The velocities of the left wheel V_L and of the right wheel V_R are determined as follows:

$$V_R = V + \Delta V, \quad (5)$$

$$V_L = V - \Delta V, \quad (6)$$

$$\Delta V = K_t \theta_{\text{human}} + K_{tD} \dot{\theta}_{\text{human}}. \quad (7)$$

The parameters $D_{\text{stop}}, K_V, K_t,$ and K_{tD} were empirically determined, following [15], as $D_{\text{stop}} = 0.7$ m, $K_V = 0.45\ \text{s}^{-1}$, $K_t = 0.1$ m/s·rad, and $K_{tD} = 0.015$ m/rad.

3.4.2. Following the Waypoints

Following the waypoints can be realized using the same control strategy as explained above. To do this, we need to assume a virtual target, instead of a target person. Figure 12 shows the concept of the virtual target. a virtual target is set on a line beyond the next waypoint, such that the robot can arrive at the next waypoint quickly. When the robot is near enough to the next waypoint, the robot judges that it has arrived at the waypoint and changes the target to the next waypoint.

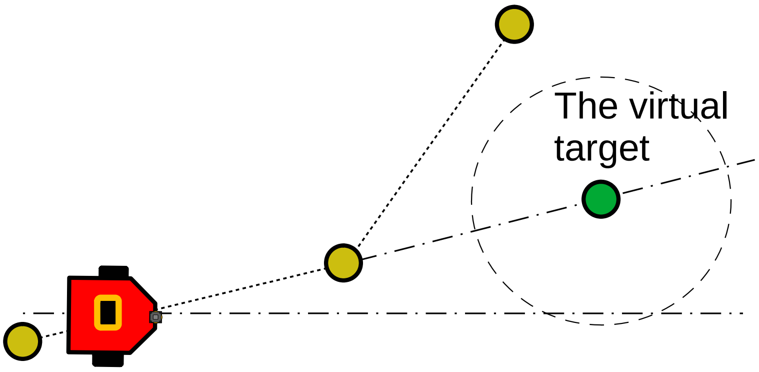


Figure 12. The virtual target.

Figure 13 shows the robot's waypoint-following behavior, where $(WP_{x(k)}, WP_{y(k)})$ is the coordinate of the k-th waypoint, (x_{vr}, y_{vr}) is the coordinate of the robot projected onto the line connecting the waypoints, and (x_{vt}, y_{vt}) is the coordinate of the virtual target. The virtual target is always set D_{vt} ahead from the projected coordinate of the robot along the line connecting the previous and the next waypoint. The robot moves toward the virtual target and, when the robot arrives at a point within D_{ar} of the next waypoint, the robot moves toward the new virtual target. In the experiments, we used $D_{\text{vt}} = 1.5$ m and $D_{\text{ar}} = 0.2$ m.

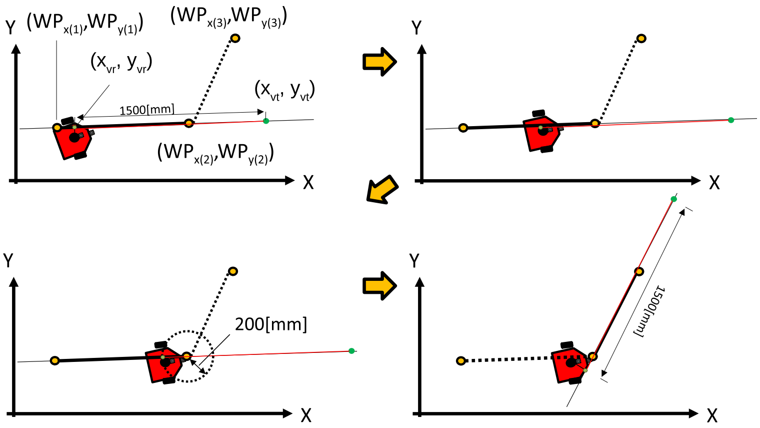

Y
(WP$_{x(1)}$,WP$_{y(1)}$)
(WP$_{x(3)}$,WP$_{y(3)}$)
(x$_{vr}$, y$_{vr}$)
(x$_{vt}$, y$_{vt}$)
1500[mm]
(WP$_{x(2)}$,WP$_{y(2)}$)
X
Y
X
Y
200[mm]
X
Y
1500[mm]
X

Figure 13. Following the waypoints.

The calculation of the coordinate of the virtual target is depicted in Figure 14. The line connecting the waypoints is denoted by $y = ax + b$, and the line perpendicular to the previous line and crossing the center coordinate of the robot (x_r, y_r) is denoted by $y = cx + d$. Here, the constants a, b, c, and d are calculated as follows:

$$a = \frac{WP_{y(2)} - WP_{y(1)}}{WP_{x(2)} - WP_{x(1)}}, \tag{8}$$

$$b = WP_{y(1)} - a \cdot WP_{x(1)}, \tag{9}$$

$$c = -\frac{1}{a}, \tag{10}$$

$$d = y_r - c \cdot x_r. \tag{11}$$

We calculate the coordinates (x_{vr}, y_{vr}) and (x_{vt}, y_{vt}) using these constants, as follows:

$$x_{vr} = \frac{b - d}{c - a}, \tag{12}$$

$$y_{vr} = a \cdot x_{vr} + b, \tag{13}$$

$$x_{vt} = x_{vr} \pm \frac{R}{\sqrt{a^2 + 1}}, \tag{14}$$

$$y_{vt} = a \cdot x_{vt} + b, \tag{15}$$

where R is the distance between (x_{vr}, y_{vr}) and (x_{vt}, y_{vt}), which is set to 1500 mm. As we have two coordinates of (x_{vt}, y_{vt}), we take the point nearest to the next waypoint $(WP_{x(2)}, WP_{y(2)})$.

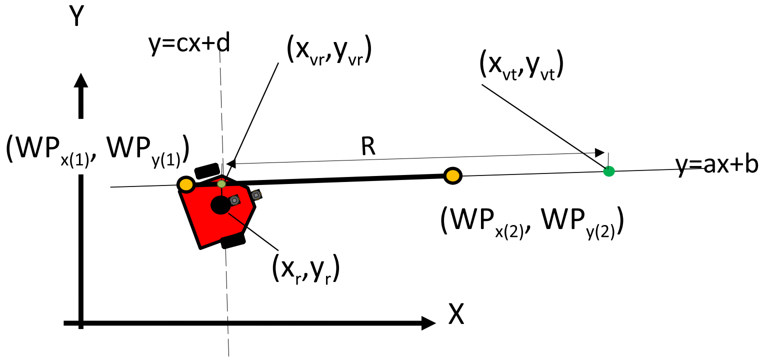


Figure 14. Calculation of the virtual target.

3.4.3. Quick Recovery to the Original Path

When the robot moves far from the path (e.g., when avoiding an obstacle), the robot tries to return to the original path using the algorithm described above. However, when the distance between the path and the robot is large, it takes time for the robot to return to the path. Moreover, if the distance is larger than a certain length, the robot will fail to detect arrival to the next waypoint. Figure 15 depicts this situation, where L is the distance between the center of the robot and the path. Let the distance between the waypoints be D_{iw}. When the robot is at the side of a waypoint, the minimum distance between the robot and the next waypoint is

$$D_{\mathrm{wmin}} = \frac{L(D_{\mathrm{vt}} - D_{\mathrm{iw}})}{D_{\mathrm{vt}}}, \tag{16}$$

as the robot moves straight to the virtual target. The robot recognizes arrival at the next waypoint when $D_{\mathrm{wmin}} \leq D_{\mathrm{ar}}$. As D_{wmin} is proportional to L, the robot cannot arrive at the next waypoint when

$$L > \frac{D_{\mathrm{vt}} D_{\mathrm{ar}}}{D_{\mathrm{vt}} - D_{\mathrm{iw}}}. \tag{17}$$

For example, when $D_{\rm vt} = 1.5$ m, $D_{\rm iw} = 0.74$ m, and $D_{\rm ar} = 0.2$ m, the robot cannot arrive at the next waypoint when $L > 0.395$ m. To solve this problem, we propose to move the virtual target according to L. In Figure 15, (x_{nt}, y_{nt}) is the coordinate of the new virtual target, which is a distance of L away from the line. The coordinate of the new virtual target is calculated as follows:

$$x_{nt} = x_{vt} + x_{vr} - x_r, \quad (18)$$

$$y_{nt} = y_{vt} + y_{vr} - y_r. \quad (19)$$

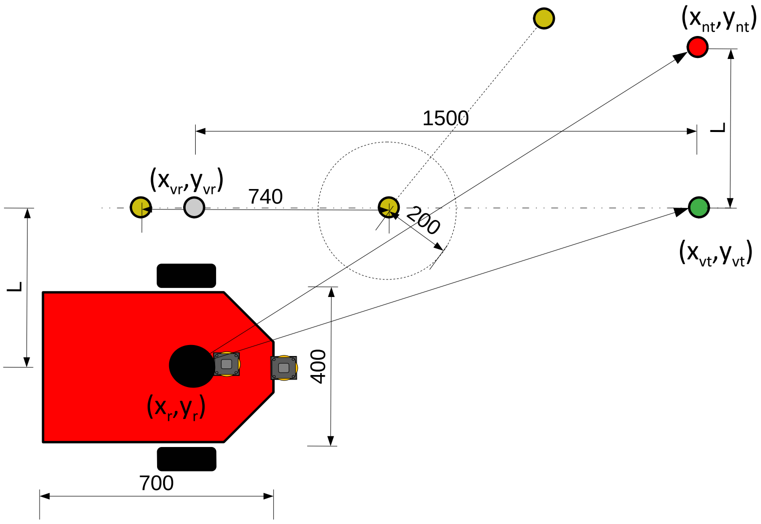
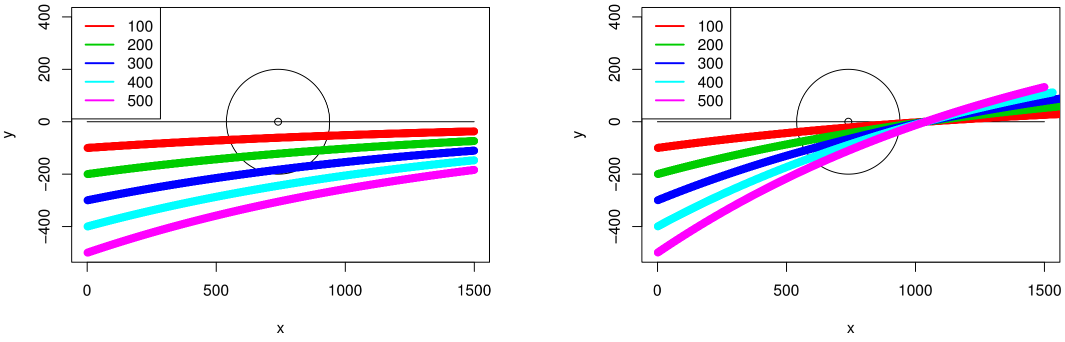

(x_{nt},y_{nt})
1500
L
(x_{vr},y_{vr})
740
200
(x_{vt},y_{vt})
L
400
(x_r,y_r)
700

Figure 15. The new virtual target.

Figure 16 shows a simulation of the robot's trajectory using the old virtual target (x_{vt}, y_{vt}) and the new one (x_{nt}, y_{nt}) from five initial positions: $(0, -100)$, $(0, -200)$, $(0, -300)$, $(0, -400)$, and $(0, -500)$. The small and big circles in the middle of the figure are the next waypoint and the area for judging the arrival at the waypoint, respectively. If we use the old virtual target, it is obvious that we cannot arrive within the judgment circle when the initial position is far from the path. When using the new virtual target, we can arrive at the waypoint, regardless of the initial position.



(a) Old virtual target (b) New virtual target

Figure 16. Simulation of the path with two kinds of virtual targets (old and new).

3.5. Combination with Obstacle Avoidance

The proposed pedestrian avoidance method can be combined with an obstacle avoidance method. By combining these methods, the robot can avoid obstacles nearby, as well as avoiding distant pedestrians while considering the public distance.

We employed the obstacle avoidance method by Sakai et al. [23]. Here, we briefly explain the obstacle avoidance method and the combination of the two avoidance methods.

The method of Sakai et al. was designed to avoid obstacles while following a person (the target). The avoidance algorithm is as follows:

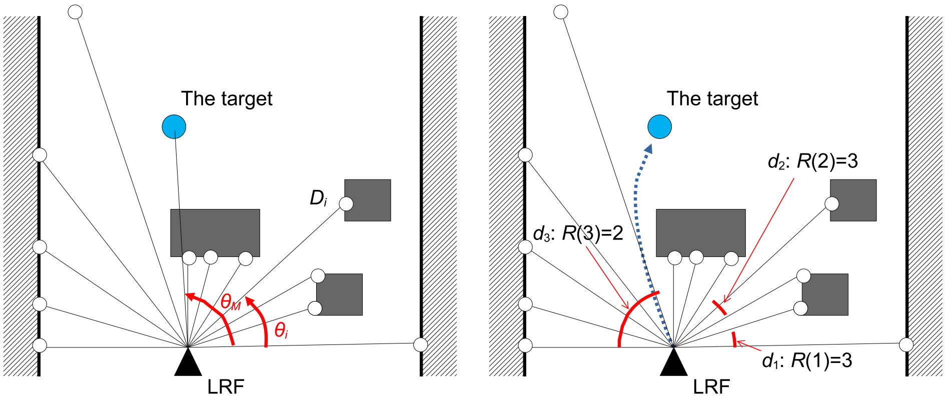
1. The LRF observes the space in front of the robot, and the robot calculates the regions where the robot will collide with an obstacle.
2. Let θ_i be the i-th angle observed from the LRF and D_i be the distance to any object (the target, an obstacle, or the wall) at an angle of θ_i (Figure 17a). List all regions (contiguous angles observed by the LRF) through which the robot can pass. Let $d_1, \ldots, d_n$ be such regions, where d_k covers all angles from θ_{b_k} to θ_{e_k} ($\theta_{b_k} < \theta_{e_k}$).
3. Judge whether the robot needs to avoid any obstacles, considering the positions of the robot and the target. If the straight path from the LRF to the target is included in any of the regions $d_1, \ldots, d_n$, then there is no need to avoid an obstacle.
4. Let the angle from the LRF toward the target be θ_M and

$$\phi_k = \min_{i \in \{b_k, e_k\}} |\theta_M - \theta_i|. \tag{20}$$

   Then, let $R_{\mathrm{deg}}(k)$ be the rank of ϕ_k among $\phi_1, \ldots, \phi_n$, in ascending order (the smallest ϕ_k has the highest rank). Now, $R_{\mathrm{deg}}(k)$ indicates how near the region d_k is to the target.
5. Let $\bar{D}_k$ be the average distance to the object at region d_k, as follows:

$$\bar{D}_k = \frac{1}{e_k - b_k + 1} \sum_{i=b_k}^{e_k} D_i. \tag{21}$$

6. Let $R_{\mathrm{dis}}(k)$ be the rank of $\bar{D}_k$ among $\bar{D}_1, \ldots, \bar{D}_n$, in descending order (the largest $\bar{D}_k$ has the highest rank).
7. Let $R(k) = w_{\mathrm{deg}} R_{\mathrm{deg}}(k) + w_{\mathrm{dis}} R_{\mathrm{dis}}(k)$.
8. Determine the region with the highest rank; $\hat{k} = \arg\min_k R(k)$. Let the robot move toward $d_{\hat{k}}$ (Figure 17b). If there are ties, then we choose the region with the best R_{dis}.



(**a**) Observation of obstacles. (**b**) Determination of the avoidance path.

Figure 17. The obstacle avoidance algorithm.

This algorithm can be easily combined with our pedestrian avoidance method, by regarding the virtual target as the target.

4. Experiment

4.1. Overview and Conditions of the Experiment

We conducted an experiment to validate the proposed method. We checked for the following conditions:

1. The robot detects an oncoming pedestrian walking towards it;
2. The robot tracks the detected pedestrian;
3. The robot determines the area for moving to avoid the pedestrian;
4. The robot returns to the original path after passing by the pedestrian; and
5. The robot moves in front of the person to be guided, keeping a proper distance.

Figure 18 shows the experimental environment. The width of the corridor was 2.38 m. The pedestrian walked toward the robot from a point 15 m away with a speed of 1.4 m/s or 0.7 m/s. We used a metronome to control the pedestrian's walking speed. The pedestrian and the guided person started simultaneously. We prepared three trajectories, as shown in Figure 18.

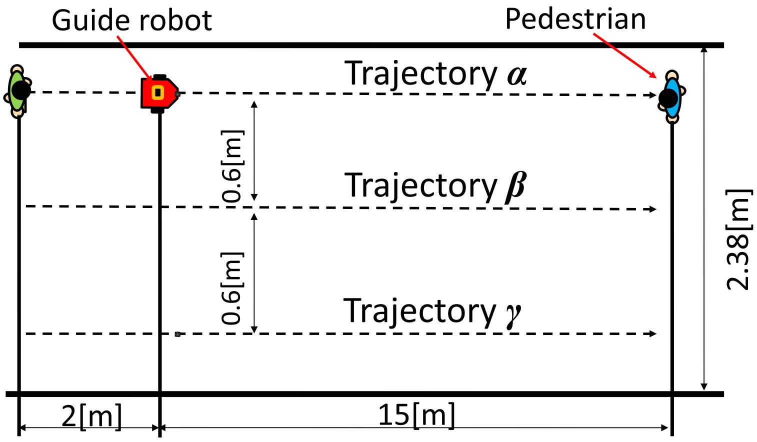


Figure 18. Trajectories used in the experiment.

Figure 19 shows the robot we used for the experiment. The LRF for observing the pedestrian was mounted on the point 1000 mm above floor level. The base of the robot was a Pioneer-3DX by MobileRobots Inc. The maximum speed of the base was 1.6 m/s, but we restricted the maximum speed to be 0.4 m/s, considering the load. One LRF (LRF1), as shown in Figure 19, was used for detecting humans, and another (LRF2) was used for creating a map. We used the ICP scan matching package of the Mobile Robot Programming Toolkit (MRPT) for creating the map. Another LRF (LRF3) was used for detecting obstacles. The RGB-D sensor detected small objects on the floor. Finally, a fourth LRF (LRF4) was used to recognize the person to guide.

4.2. Experimental Results

Figure 20 shows photos of the experiment. The robot could successfully avoid the oncoming pedestrian for all the three trajectories and both walking speeds (Figure 20a). The average distance from which the robot detected the oncoming pedestrian was 9.3 m (Figure 20b). The robot could move to the broader side area for avoidance (Figure 20c). After passing by the pedestrian (Figure 20d), the robot could return to the original path (Figure 20e,f). The first half of Video S1 shows the video recorded at the experiment. In the video, we also recorded the robots map where the pedestrian is recognized.

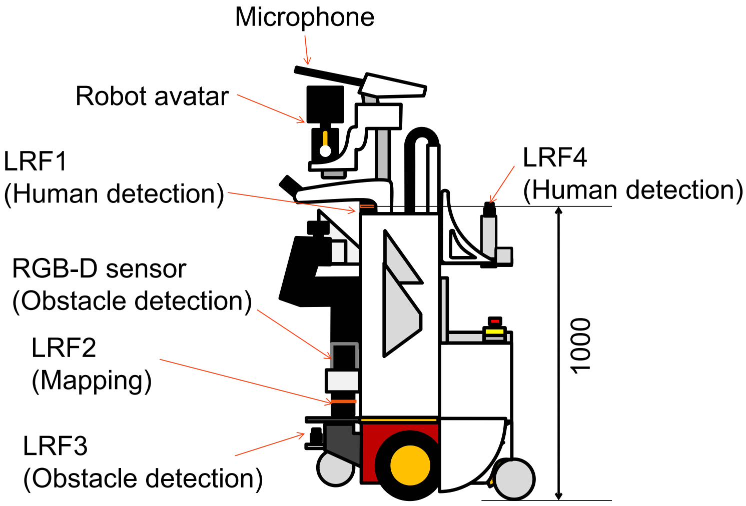
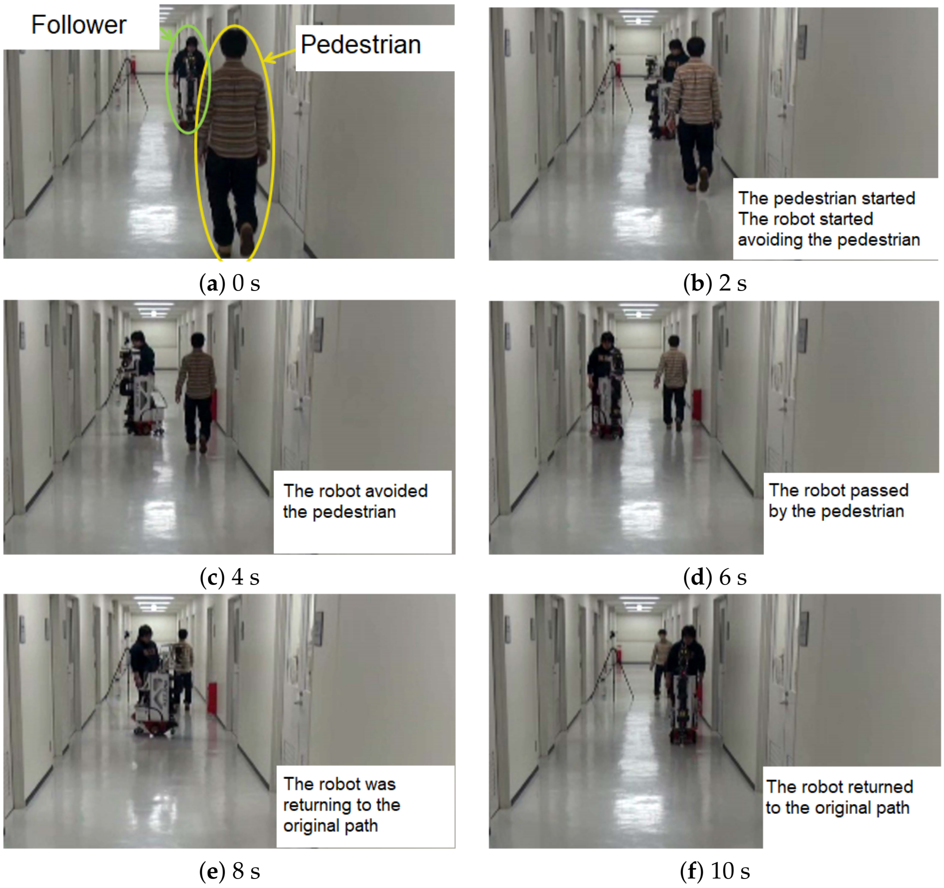


Figure 19. The robot (ASAHI) used in the experiment.



(**a**) 0 s (**b**) 2 s

(**c**) 4 s (**d**) 6 s

(**e**) 8 s (**f**) 10 s

Figure 20. Photos of the experiment (trajectory α, 1.4 m/s).

The average distance between the robot and the pedestrian when passing was 1.19 m. This distance depended on the trajectory and width of the corridor. It is left as future work to determine the best distance to keep between the robot and the pedestrian.

4.3. The Application Experiment

We proved that the robot worked properly with the proposed method. In this section, we describe the results of an experiment conducted in a more realistic environment, where there were obstacles in addition to the oncoming pedestrian. In this experiment, we evaluated whether the robot could keep a distance from the pedestrian while avoiding both the pedestrian and the obstacles.

In the experiment, we implemented that the robot detected obstacles from a distance of 2.0 m and set $w_{\text{dis}} = w_{\text{deg}} = 1.0$.

Figure 21 shows the experimental environment. The distances between the robot and the left and right walls were 2 m and 4 m, respectively. The pedestrian started walking at 12.6 m away from the robot. The walking speed of the pedestrian was 0.70 m/s. We placed two obstacles, a can ($\phi \times$ H, 0.05×0.09 m) and a box (WDH, $0.3 \times 0.3 \times 0.4$ m). The maximum speed of the robot was 0.4 m/s. We set the waypoints before the experiment. The waypoints were aligned every 0.75 m. We conducted three trials for each of the walking speeds.

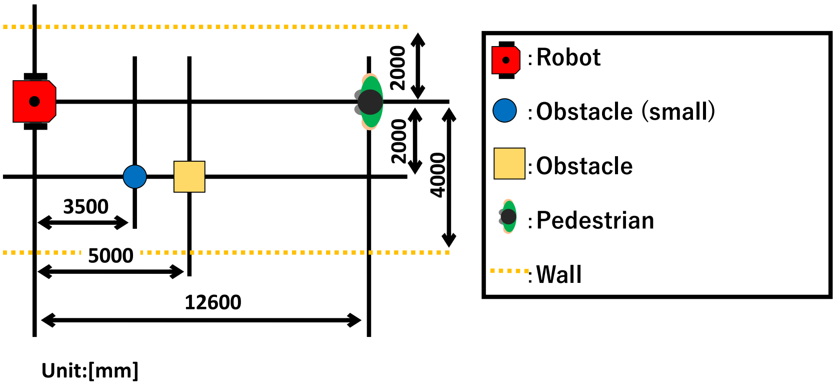


Figure 21. The experimental environment.

Figure 22 shows the example trajectory of the robot. We conducted three trials; the trajectory shown in the figure is the result of the first trial. In this example, the pedestrian passed the robot while the robot was avoiding the obstacle, after which the robot returned to the original waypoints directly.

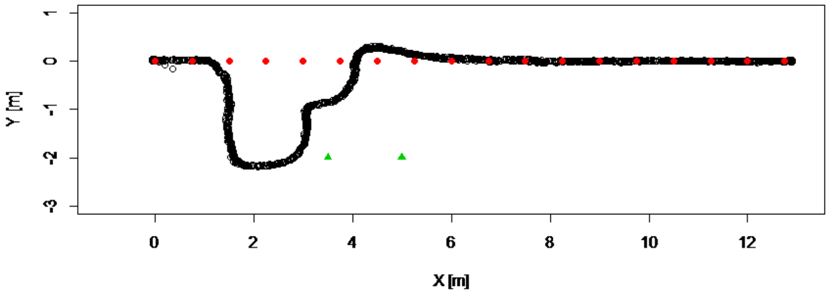
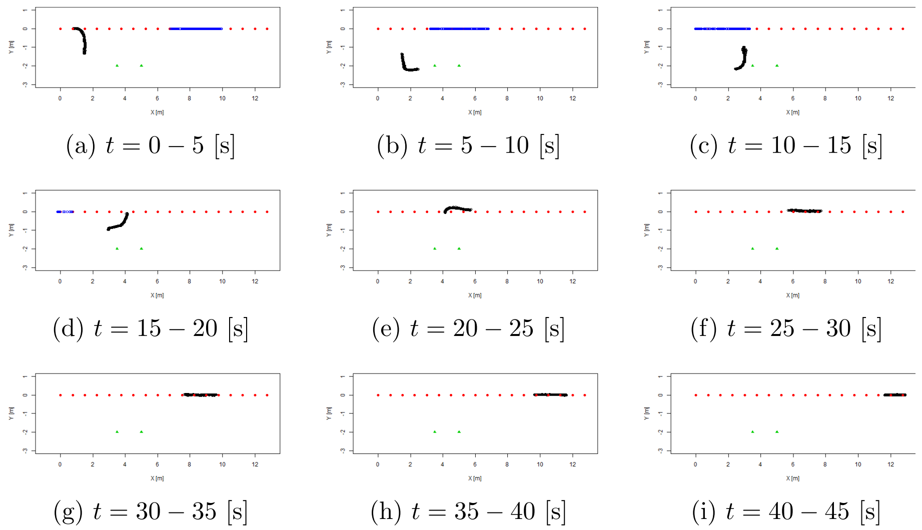


Figure 22. Trajectory of the robot.

Figure 23 shows the short-term trajectories (5 s) of the robot and the pedestrian. The black and blue points show the trajectories of the robot and the pedestrian, respectively. The green points are the positions of the obstacles. The robot detected the pedestrian and avoided them during 0–6 s. Then, it detected another obstacle at 5–10 s. At 10–15 s, it avoided the obstacles. Finally, it returned to the original waypoints. The pedestrian passed by the robot during t = 10–15 s, and then returned to the original waypoints. The trajectory of the pedestrian vanished after 15–20 s, as the robot stopped tracking the pedestrian after it passed by them. The second half of Video S1 shows the video recorded at the experiment. In this video, we clearly confirm that the robot started avoiding the oncoming pedestrian first, after which it avoids the obstacles nearby.



(a) $t = 0 - 5$ [s] (b) $t = 5 - 10$ [s] (c) $t = 10 - 15$ [s]

(d) $t = 15 - 20$ [s] (e) $t = 20 - 25$ [s] (f) $t = 25 - 30$ [s]

(g) $t = 30 - 35$ [s] (h) $t = 35 - 40$ [s] (i) $t = 40 - 45$ [s]

Figure 23. Short-term trajectories of the robot (black) and the pedestrian (blue).

Finally, Figure 24 shows the human–robot distances in the three trials. We can see that the human–robot distances became smaller, and then became large again. As the robot stopped tracking the pedestrian, most data ended after the human–robot distance became minimal. The average of the minimum distances was 1.87 m.

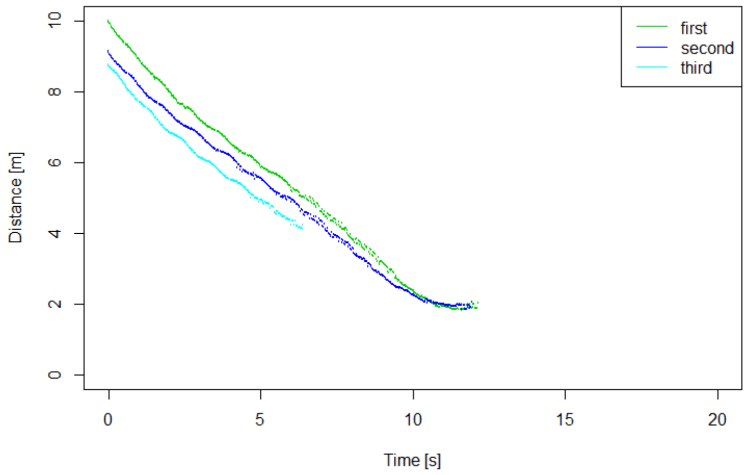


Figure 24. Human–robot distances.

Overall, the system operated properly. The proposed pedestrian avoidance algorithm worked under a more complex situation, and combination with the obstacle avoidance algorithm also worked without any problem. The average minimum human–robot distance when the robot passed by the pedestrian was 1.87 m, greater than the mentally-safe distance, 1.2 m.

5. Conclusions

In this paper, we have proposed a method for a robot to avoid an oncoming pedestrian in a corridor while moving along a recorded path. The proposed method involves methods for detecting the pedestrian, determining the area to avoid, calculating the distance to move aside, generating the avoidance path, determining the passing-by of the pedestrian, and returning to the original path. When avoiding the pedestrian, the distance to the pedestrian was 9.3 m on average, which was far enough to keep a public distance. When passing by the pedestrian, the distance to the pedestrian was 1.19 m, which was far enough, considering the person's impression [20]. All of the avoiding behaviors were performed while guiding a person by moving in front of them.

In addition, we carried out an experiment on pedestrian avoidance in an environment with other obstacles. As a result, the robot could avoid the obstacles while avoiding the pedestrian from 8 m away.

There are four limitations to our method. First, only one oncoming pedestrian was considered; if there were more pedestrians, the robot could not maintain a public distance and uses only the existing obstacle avoidance method to avoid pedestrians. Second, this method assumes that the pedestrian moves toward the robot from the front. Third, the corridor must be wide be enough for the robot to avoid the pedestrian. Finally, the floor of the corridor must be flat, such that obstacles on the floor can be detected by the LRF.

Our method can be applied for controlling an autonomous mobile robot, not only for the proposed situation but also for broader situations, such as for guiding robots in hospitals [38,39] or museums [40–42], where there are a relatively small number of persons and long, flat corridors. It may also be applicable in intelligent wheelchairs [43,44]. Our future work is to design avoidance behavior methods for larger spaces with many people [45]. In the present work, we used a simple control strategy for maintaining a distance to the following person. This may be improved by introducing a more sophisticated control method, which considers the distance to the follower [46].

Supplementary Materials: The following are available online at http://www.mdpi.com/2218-6581/8/4/97/s1, Video S1: Video of pedestrian avoidance experiments described in Sections 4.2 and 4.3.

Author Contributions: Conceptualization, methodology, software, validation, Y.H.; formal analysis, investigation, Y.H. and A.I..; resources, Y.H.; data curation, Y.H.; writing–original draft preparation, Y.H.; writing–review and editing, A.I.; visualization, A.I.; supervision, A.I.; project administration, Y.H.; funding acquisition, Y.H.

Funding: This research was funded by JSPS Kakenhi JP16K00363.

Acknowledgments: Yudai Miyauchi, Koki Syono, and Keisuke Sakai contributed to conducting the experiment.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Iwata, H.; Sugano, S. Design of human symbiotic robot TWENDY-ONE. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'09), Kobe, Japan, 12–17 May 2009; pp. 580–586.
2. Kanda, T.; Shiomi, M.; Miyashita, Z.; Ishiguro, H.; Hagita, N. An affective guide robot in a shopping mall. In Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction, La Jolla, CA, USA, 9–13 March 2009; pp. 173–180.
3. Gandhi, T.; Trivedi, M.M. Pedestrian collision avoidance systems: a survey of computer vision based recent studies. In Proceedings of the 2006 IEEE Intelligent Transportation Systems Conference, Toronto, ON, Canada, 17–20 September 2006; pp. 976–981. [CrossRef]
4. Snape, J.; van den Berg, J.; Guy, S.J.; Manocha, D. Smooth and collision-free navigation for multiple robots under differential-drive constraints. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4584–4589. [CrossRef]
5. Hoy, M.; Matveev, A.S.; Savkin, A.V. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica* **2015**, *33*, 463–497. [CrossRef]
6. Almasri, M.; Elleithy, K.; Alajlan, A. Sensor fusion based model for collision free mobile robot navigation. *Sensors* **2016**, *16*, 24. [CrossRef] [PubMed]
7. Little, K.B. Personal space. *J. Exp. Soc. Psychol.* **1965**, *1*, 237–247. [CrossRef]
8. Hall, E.T. Proxemics. *Curr. Anthropol.* **1968**, *9*, 83–108. [CrossRef]
9. Tamura, Y.; Fukuzawa, T.; Asama, H. Smooth collision avoidance in human-robot coexisting environment. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 3887–3892.
10. Sardar, A.; Joosse, M.; Weiss, A.; Evers, V. Don'T Stand So Close to Me: Users' Attitudinal and Behavioral Responses to Personal Space Invasion by Robots. In Proceedings of the 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI '12), Boston, MA, USA, 5–8 March 2012; ACM: New York, NY, USA, 2012; pp. 229–230. [CrossRef]

11. Kruse, T.; Pandey, A.K.; Alami, R.; Kirsch, A. Human-aware robot navigation: a survey. *Robot. Auton. Syst.* **2013**, *61*, 1726–1743. [CrossRef]
12. Arai, M.; Sato, Y.; Suzuki, R.; Kobayashi, Y.; Kuno, Y.; Miyazawa, S.; Fukushima, M.; Yamazaki, K.; Yamazaki, A. Robotic wheelchair moving with multiple companions. In Proceedings of the 23rd IEEE International Symposium on Robot and Human Interactive Communication, Edinburgh, UK, 25–29 August 2014; pp. 513–518. [CrossRef]
13. Hiroi, Y.; Ito, A. ASAHI: OK for Failure—A Robot for Supporting Daily Life, Equipped with a Robot Avatar. In Proceedings of the 8th ACM/IEEE International Conference on Human-Robot Interaction, Tokyo, Japan, 3–6 March 2013; pp. 141–142.
14. Yuan, F.; Twardon, L.; Hanheide, M. Dynamic path planning adopting human navigation strategies for a domestic mobile robot. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2010; pp. 3275–3281.
15. Sakai, K.; Hiroi, Y.; Ito, A. Teaching a robot where objects are: Specification of object location using human following and human orientation estimation. In Proceedings of the World Automation Congress (WAC), Waikoloa, HI, USA, 3–7 August 2014; pp. 490–495. [CrossRef]
16. Alvarez-Santos, V.; Canedo-Rodriguez, A.; Iglesias, R.; Pardo, X.; Regueiro, C.; Fernandez-Delgado, M. Route learning and reproduction in a tour-guide robot. *Robot. Auton. Syst.* **2015**, *63*, 206–213. [CrossRef]
17. Akai, N.; Morales, L.Y.; Murase, H. Teaching-Playback Navigation Without a Consistent Map. *J. Robot. Mechatron.* **2018**, *30*, 591–597. [CrossRef]
18. Iocchi, L.; Holz, D.; del Solar, J.R.; Sugiura, K.; van der Zant, T. RoboCup@Home: Analysis and results of evolving competitions for domestic and service robots. *Artif. Intell.* **2015**, *229*, 258–281. [CrossRef]
19. Matamoros, M.; Seib, V.; Memmesheimer, R.; Paulus, D. RoboCup@Home: Summarizing achievements in over eleven years of competition. In Proceedings of the IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), Torres Vedras, Portugal, 25–27 April 2018; pp. 186–191. [CrossRef]
20. Paccierotti, E. Evaluation of passing distance for social robots. In Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication, Hatfield, UK, 6–8 September 2006; pp. 315–320.
21. Shiomi, M.; Zanlungo, F.; Hayashi, K.; Kanda, T. Towards a Socially Acceptable Collision Avoidance for a Mobile Robot Navigating Among Pedestrians Using a Pedestrian Model. *Int. J. Soc. Robot.* **2014**, *6*, 443–455. [CrossRef]
22. Yuta, S.; Mizukawa, M.; Hashimoto, H.; Tashiro, H.; Okubo, T. An open experiment of mobile robot autonomous navigation at the pedestrian streets in the city — Tsukuba Challenge. In Proceedings of the 2011 IEEE International Conference on Mechatronics and Automation, Beijing, China, 7–10 August 2011; pp. 904–909. [CrossRef]
23. Sakai, K.; Hiroi, Y.; Ito, A. Proposal of an obstacle avoidance method considering the LRF-based human following. In Proceedings of the Robotics Society Japan Annual Meeting, Fukuoka, Japan, 4–6 September 2014; pp. 2D2–06.
24. Koren, Y.; Borenstein, J. Potential field methods and their inherent limitations for mobile robot navigation. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; Volume 2, pp. 1398–1404. [CrossRef]
25. Fox, D.; Burgard, W.; Thrun, S. The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [CrossRef]
26. Charalampous, K.; Kostavelis, I.; Gasteratos, A. Recent trends in social aware robot navigation: A survey. *Robot. Auton. Syst.* **2017**, *93*, 85–104. [CrossRef]
27. Kim, B.; Pineau, J. Socially Adaptive Path Planning in Human Environments Using Inverse Reinforcement Learning. *Int. J. Soc. Robot.* **2016**, *8*, 51–66. [CrossRef]
28. Chen, Y.F.; Everett, M.; Liu, M.; How, J.P. Socially aware motion planning with deep reinforcement learning. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1343–1350.
29. Kostavelis, I.; Kargakos, A.; Giakoumis, D.; Tzovaras, D. Robot's Workspace Enhancement with Dynamic Human Presence for Socially-Aware Navigation. In *Computer Vision Systems*; Liu, M., Chen, H., Vincze, M., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 279–288.

30. Tai, L.; Zhang, J.; Liu, M.; Burgard, W. Socially compliant navigation through raw depth inputs with generative adversarial imitation learning. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1111–1117.
31. Hoshino, S.; Maki, K. Safe and efficient motion planning of multiple mobile robots based on artificial potential for human behavior and robot congestion. *Adv. Robot.* **2015**, *29*, 1095–1109. [CrossRef]
32. Papadakis, P.; Rives, P.; Spalanzani, A. Adaptive spacing in human-robot interactions. In Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 14–18 September 2014; pp. 2627–2632.
33. Aotani, Y.; Ienaga, T.; Machinaka, N.; Sadakuni, Y.; Yamazaki, R.; Hosoda, Y.; Sawahashi, R.; Kuroda, Y. Development of Autonomous Navigation System Using 3D Map with Geometric and Semantic Information. *J. Robot. Mechatron.* **2017**, *29*, 639–648. [CrossRef]
34. Matsumaru, T. Development of Four Kinds of Mobile Robot with Preliminary-Announcement and Indication Function of Upcoming Operation. *J. Robot. Mechatron.* **2007**, *19*, 148–159. [CrossRef]
35. Nakamori, Y.; Hiroi, Y.; Ito, A. Multiple player detection and tracking method using a laser range finder for a robot that plays with human. *ROBOMECH J.* **2018**, *5*, 25. [CrossRef]
36. Yoda, M.; Shiota, Y. Mobile Robot's Passing Motion Algorithm Based on Subjective Evaluation. *Trans. Jpn. Soc. Mech. Eng.* **2000**, *66*, 156–163.
37. National Institute of Advanced Industrial Science and Technology. AIST Anthropometric Database 1991–1992. Available online: https://www.airc.aist.go.jp/dhrt/91-92/ (accessed on 6 November 2019).
38. Takahashi, M.; Suzuki, T.; Shitamoto, H.; Moriguchi, T.; Yoshida, K. Developing a mobile robot for transport applications in the hospital domain. *Robot. Auton. Syst.* **2010**, *58*, 889–899. [CrossRef]
39. Ljungblad, S.; Kotrbova, J.; Jacobsson, M.; Cramer, H.; Niechwiadowicz, K. Hospital Robot at Work: Something Alien or an Intelligent Colleague? In Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work (CSCW '12), Seattle, WA, USA, 11–15 February 2012; ACM: New York, NY, USA, 2012; pp. 177–186. [CrossRef]
40. Burgard, W.; Cremers, A.B.; Fox, D.; Hähnel, D.; Lakemeyer, G.; Schulz, D.; Steiner, W.; Thrun, S. Experiences with an interactive museum tour-guide robot. *Artif. Intell.* **1999**, *114*, 3–55. [CrossRef]
41. Kuno, Y.; Sadazuka, K.; Kawashima, M.; Yamazaki, K.; Yamazaki, A.; Kuzuoka, H. Museum guide robot based on sociological interaction analysis. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, San Jose, CA, USA, 28 April–3 May 2007; pp. 1191–1194.
42. Yamazaki, A.; Yamazaki, K.; Ohyama, T.; Kobayashi, Y.; Kuno, Y. A techno-sociological solution for designing a museum guide robot: Regarding choosing an appropriate visitor. In Proceedings of the 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI), Boston, MA, USA, 5–8 March 2012; pp. 309–316. [CrossRef]
43. Parikh, S.P.; Grassi, V., Jr.; Kumar, V.; Okamoto, J., Jr. Integrating human inputs with autonomous behaviors on an intelligent wheelchair platform. *IEEE Intell. Syst.* **2007**, *22*, 33–41. [CrossRef]
44. Matsumoto, O.; Komoriya, K.; Toda, K.; Goto, S.; Hatase, T.; Nishimura, H. Autonomous traveling control of the "TAO Aicle" intelligent wheelchair. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 4322–4327.
45. Morishita, K.; Hiroi, Y.; Ito, A. a Crowd Avoidance Method Using Circular Avoidance Path for Robust Person Following. *J. Robot.* **2017**, *2017*, 3148202. [CrossRef]
46. Fujiwara, Y.; Hiroi, Y.; Tanaka, Y.; Ito, A. Development of a mobile robot moving on a handrail—Control for preceding a person keeping a distance. In Proceedings of the 24th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Kobe, Japan, 31 August–4 September 2015; pp. 413–418.