sciendo INTERNATIONAL JOURNAL ON SMART SENSING AND INTELLIGENT SYSTEMS

Article | DOI: 10.2478/ijssis-2022-0017 Issue 1 | Vol. 15 (2022)

A novel design of a smart interactive guiding robot for busy airports

Hoang T. Tran[1], Thanh C. Vo[1], Quan N.A. Nguyen[1], Quang N. Pham[1], Duyen M. Ha[1], Thanh Q. Le[1], Thang K. Nguyen[1], Dong LT. Tran[1], Hai T. Do[2] and Minh T. Nguyen[2,*]

[1]Center of Electrical Engineering, Duy Tan University, Da Nang, Viet Nam.

[2]Thai Nguyen University of Technology, Thai Nguyen, Viet Nam.

*E-mail: nguyentuanminh@tnut.edu.vn

This paper was edited by Subhas Chandra Mukhopadhyay.

Received for publication August 6, 2022.

Abstract

The COVID-19 pandemic has had a massive impact on the global aviation industry. As a result, the airline industry has been forced to embrace new technologies and procedures in order to provide a more secure and bio-safe travel. Currently, the role of smart technology in airport systems has expanded significantly as a result of the contemporary Industry 4.0 context. The article presents a novel construction of an intelligent mobile robot system to guide passengers to take the plane at the departure terminals at busy airports. The robot provides instructions to the customer through the interaction between the robot and the customer utilizing voice communications. The usage of the Google Cloud Speech-to-Text API combined with technical machine learning to analyze and understand the customer's requirements are deployed. In addition, we use a face detection technique based on Multi-task Cascaded Convolutional Networks (MTCNN) to predict the distance between the robot and passengers to perform the function. The robot can guide passengers to desired areas in the terminal. The results and evaluation of the implementation process are also mentioned in the article and show promise.

Keywords

Busy airport, Service robot, Multi-task Cascaded Convolutional Networks (MTCNN), Computer vision, Prediction, Distance adjustment, Hall sensor, Google Cloud Speech-to-Text, Voice recognition.

Introduction

From the initial generation of programmable teaching-reproducing robots to offline programming robots with specific sensing functions and adaptive capacities, until the intelligent robot after the mid-1980s, the robot has gone through approximately 60 years of evolution (Garcia-Haro et al., 2020; Rivai et al., 2020). The related technologies used in intelligent robots are constantly developing, such as multi-sensor information fusion, path planning, robot vision, and intelligent human–machine interfaces, due to the rapid growth of computers, information technology, artificial intelligence, and control theory. Intelligent robots with a range of sensors can efficiently respond to environmental changes through information fusion and have a strong self-adaptive, learning, and autonomous function (Nguyen et al., 2018; Nguyen et al., 2020). Recently, robots facilitated many applications in different fields including civil and military. In particular, service robots are very crucial in hazardous environments or in isolated areas, etc. that humans may not reach (Ahmad et al., 2018; Nguyen et al., 2021).

For both practitioners and scholars working with aviation applications, state-of-the-art solutions for transportation planning combining baggage services, routing, security, and safety are an expanding subject

Open Access. Published by Sciendo. CC BY 4.0
© 2022 Authors. This work is licensed under the Creative Commons Attribution-Non-Commercial-NoDerivs 4.0 License https://creativecommons.org/licenses/by-nc-nd/4.0/

(Nguyen and Teague, 2015; Sendari et al., 2019). With the goal of contributing to improving service quality, reducing work pressure for staff at airports when the number of passengers is being considered, especially by providing flight-related information, along with other necessary information for passengers conveniently and quickly (Joosse et al., 2015; Triebel et al., 2016; Ivanov et al., 2017). In other words, the usage of service robots in airports, tourism, and hospital areas also brings many benefits. Due to the COVID-19 pandemic, reducing contact between staff and passengers, reducing fear and anxiety of passengers infected with COVID-19 (Meidute-Kavaliauskiene et al., 2021). Not much different from the public robot, which can give a voice command or on-screen map to go to a predetermined location in the airport (Muñoz Peña and Bacca Cortés, 2020) or the Korean robot Troika (South China Morning Post, 2017). Passengers can find information about the flight by scanning their flight tickets, then the robot will lead its passenger to the door of the plane if they have a need, but the interaction between the robot and the passenger is also a lot. Limitations do not meet requirements such as facial expressions, and positioning and moving to the desired destination requires a fairly large displacement space with slightly large errors and the use of many complex sensor devices.

On several important issues, researchers have tried to develop various techniques that relate computer vision capabilities to applications and products for tracking moving objects. Visual Assist is a technique for controlling the robot's movements using feedback sent from the vision sensor. In visual control, it is difficult to track a moving subject if the distance of the object cannot be reached (Joosse et al., 2015), in another study, the object to be tracked was the face of a standing or back-and-forth movement or back and forth in front of the camera lens. To estimate the distance from the object, tracking an object must be solved first. Model-based tracking methods require prior knowledge of object shapes for the matching process and for finding objects in the field. These studies also apply the method of calculating the focal length of the camera thanks to an initial face image taken at a given distance (Li et al., 2015). However, the accuracy would be reduced if it was replaced by a different person with a different face. From the above-mentioned starting points, this paper focuses on building an intelligent service robot system that, in addition to the functions that can automatically move the robot, also has the function of interacting with passengers and providing information about the service robot news (namely as IRobt) , as shown in Figure 1. The robot includes some functions as (1) display flight schedule information for the day;

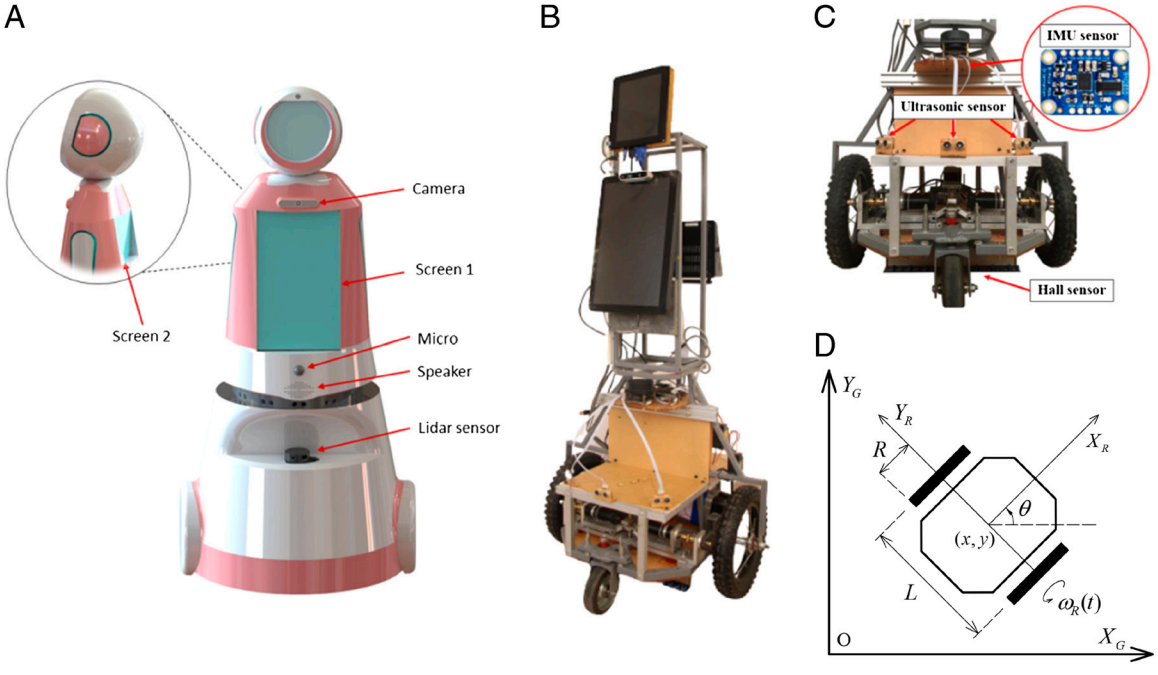

Figure 1: (a) 3D simulation image of iRobt. (b) Crafted iRobt images. (c) Location of sensors, (d) Posture and parameters of the robot in two coordinate system.

(2) show station map; (3) guide to finding eateries, cafes, shopping stores; (4) notice of prohibited items on the plane; (5) providing passenger photography and emailing services; (6) display directions, location directions, locations, and information about terminal regulations and flight rules based on passenger inquiries. All the display instructions, location and location instructions, information on station regulations, and flight rules are based on frequent questions from passengers.

To interact with passengers, we recommend adding speech recognition via the Google Cloud Speech-to-Text API for language processing. and the job of leading passengers to the area requires using a superimposed convolutional neural network to recognize human faces and then based on the similarity ratio of two triangles to determine and track the distance from the robot to the passenger. The precision tracking movement control part is applied according to the methods previously published by the research team (Thanh et al., 2021) and in addition to avoiding unexpected obstacles is also presented in the paper.

The rest of this article is organized as follows. The section "The system descriptions" presents the electronic hardware structure of the multi-sensor and communication systems in robots. The section "Proposed method" summarizes the interaction between passenger and robot, face detection based on superposition neural network and distance prediction technique, then presents motion tracking and obstacle avoidance control. The section "Experimental results" provides the experimental results with the robot built and discussed. Finally, conclusions and future work are addressed in section "Conclustions and future work" (Figure 1).

The system descriptions

To meet the application requirements for airport robots as mentioned in the section "Introduction", we propose a sensor system and actuators that can meet those requirements. Figure 2 provides an overview block diagram of the robot system. In this section, we focus on the sensing system and the PID controller, as shown in Figure 2.

Encoder

The optical encoder has an LED light source, a light detector, a "code" disc/wheel mounted on the shaft, and output signal processor, Figure 3. The disc has alternating opaque and transparent segments and is placed between the LED and photodetector. As the encoder shaft rotates, the light beam from the LED is interrupted by the opaque lines on the "code" disk before being picked up by the photodetector. This produces a pulse signal: light = on; no light = off. The signal is sent to the counter or controller, which will then send the signal to produce the desired function.

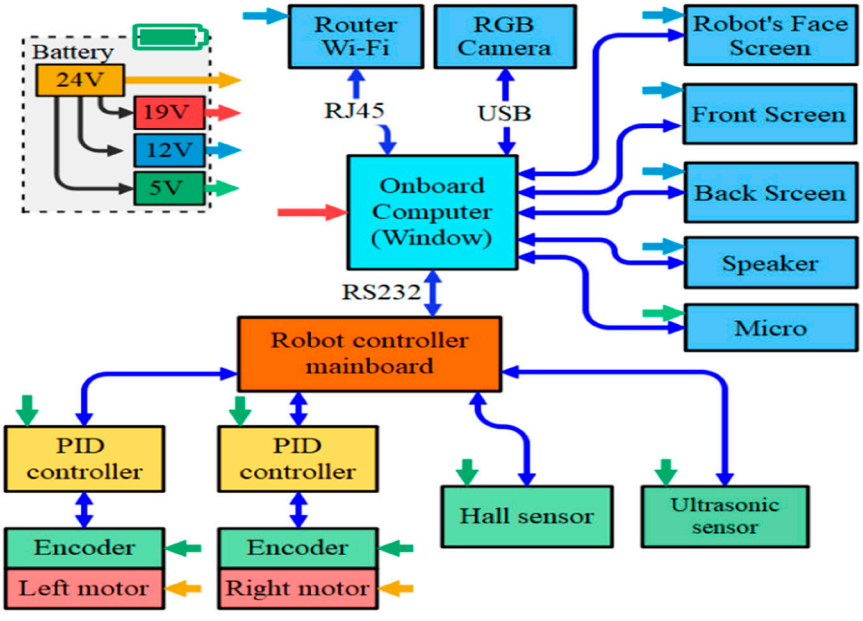

Figure 2: Block diagram of the robot system.

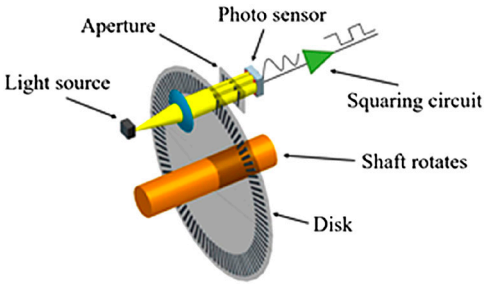

Figure 3: Structure of the rotary encoder.

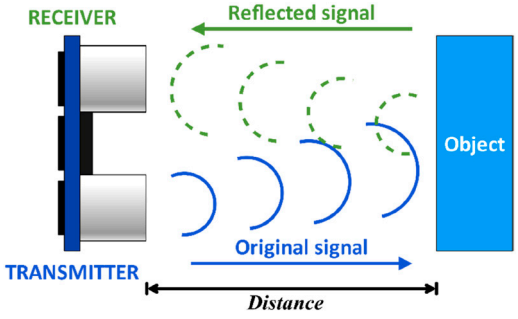

Figure 4: Ultrasonic sensor.



Figure 5: Logitech BRIO 4K.

In mobile robotics, the encoder is used to measure the movement (direction and speed) of each of the wheels of the robot. Determining the position of the robot by this encoder is a popular method in the world called the Odometry method (Qingqing et al., 2019; Thanh et al., 2021; Tran et al., 2022).

Ultrasonic sensor

An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (the sound that humans can hear). Ultrasonic sensors have two main components: the transmitter (which emits the sound using piezoelectric crystals) and the receiver (which encounters the sound after it has travelled to and from the target).

The ultrasonic sensors are used primarily as proximity sensors. They can be found in automobile self-parking technology and anticollision safety systems. Therefore, we use ultrasonic sensor SRF05 for the robot so that the robot can avoid obstacles, as illustrated in Figure 4.

RGB camera

The robot is designed in addition to the functions mentioned above, it also has the function of guiding the passengers to move in the airport. Accordingly, the robot both moves and tracks the passenger's face to maintain a distance from the passenger during movement. If the passenger stops or slows down, the robot will stop to wait, which is made possible by a method that predicts the distance from the image sensor to the person's face using the monocular camera (Pathi et al., 2019). We use the Logitech BRIO

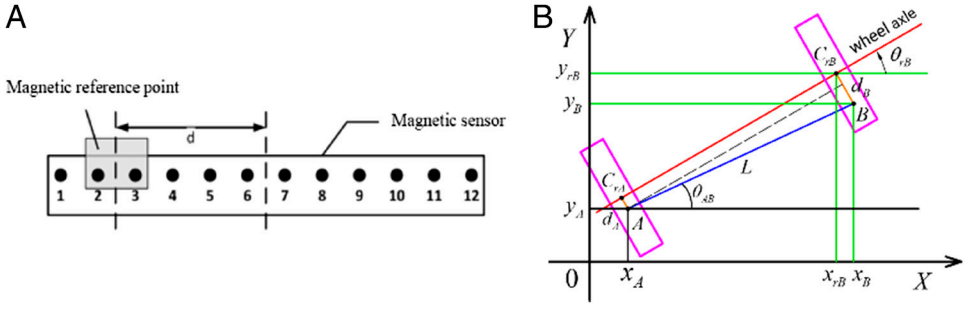

Figure 6: (a) Magnetic sensor model and magnet reference point. (b) Calculation of position and orientation of Robot at the reference point.

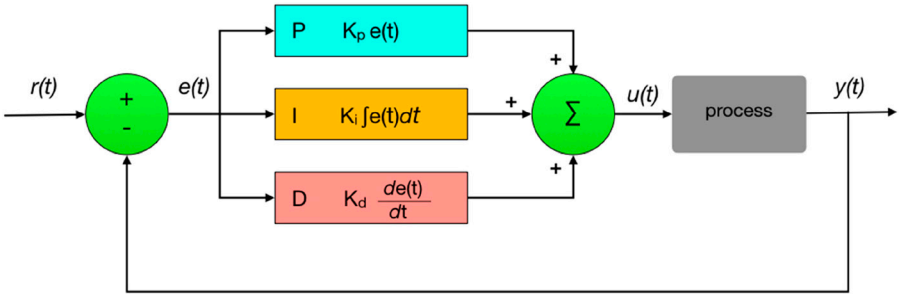

Figure 7: Block diagram of a PID controller in a feedback loop.

4K Camera, as shown in Figure 5, installed on the robot to capture the passenger's face image.

Hall sensor

As the robot travels through these reference points, a magnetic sensor is added to determine the waypoints with known coordinates and readjust the route. The magnetic sensor used is a bar consisting of 12 hall sensors placed in a line and separated by a distance of *l*=20 mm. As a result, the magnetic sensor's width is 240 mm. At the midway of the sensor bar, it will be perpendicular to the body's longitudinal axis.

The hall sensors traveling above the reference point will be engaged when the robot passes through the magnetic reference points on the floor. The deviation *d* of the body's longitudinal axis can be determined from the reference point using the position of the activated hall sensors, as shown in Figure 6 (Thanh et al., 2021).

PID controler

High-speed, high-torque, reversible DC motors are used in the driving system. For accurate location and speed detection, a quadrature optical shaft encoder with 600 pulses per rotation is mounted to each motor. A microprocessor-based electrical circuit with integrated firmware is used to accomplish the motor control, allowing to control of the motor by a PID algorithm shown in Figure 7.

The PID controller is distinguished by a control loop feedback mechanism that calculates the difference between the desired setpoint and the actual output of a process and utilizes the result to correct the operation. PID is an abbreviation for Proportional, Integral, and Derivative. The process's job is to keep a setpoint value constant. You might want a DC motor to maintain a setpoint value *r*(*t*) of 600 encoder pulses per second, for example. The error number *e*(*t*) is calculated by subtracting the actual motor speed *y*(*t*) from the setpoint value of 600. Based on the calculated error value, the PID

Table 1. Keyword and label data sheet.

No	Keywork	Tag
1	Flight information, flight schedule, etc.	__label__#fly_schedule#
2	Station map, etc.	__label__#station_map#
3	Restaurant, cafeteria, food, food, etc.	__label__#restaurant#
4	Things not to bring on board, dangerous items, etc.	__label__#dangerous_object#
5	Smile, take photo, etc.	__label__#capture_photo#
6	Hello, hello robot, etc.	__label__#greeting#
7	bye bye, goodbye, thank you, etc.	__label__#goodbye#
8	give me information about the weather, etc.	__label__#weather#
9	…	…

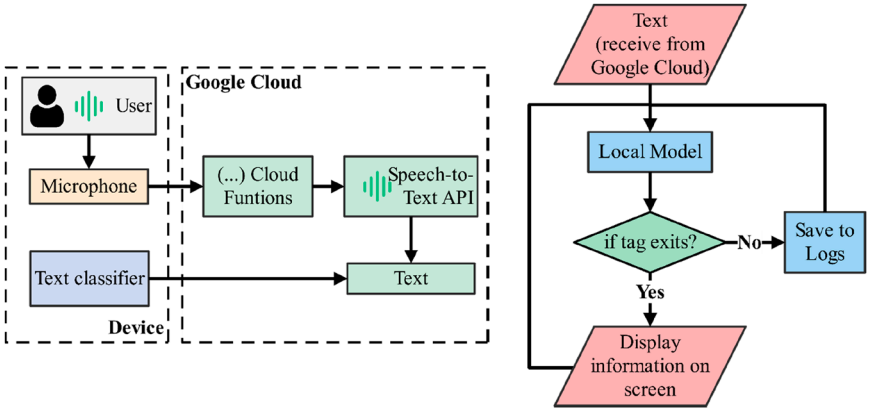

Figure 8: (a) Voice recognition model. (b) Flowchart of the passenger answering program algorithm.

controller computes the new control value $u(t)$ to apply to the motor. The control value for a DC motor would be a Pulse Width Modulation (PWM) signal. The overall control functionality is as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

K_p, K_i, and K_d are non-negative coefficients for the P, I, and D terms, respectively. In this scenario, K_i and K_d are substituted by K_p/T_i and K_pT_d, with the benefit that T_i and T_d have some physical significance since they indicate an integration time and a derivative time, respectively. The time constant K_pT_d is used by the controller to attempt to reach the setpoint. K_p/T_i specifies how long the controller will tolerate an output that is consistently above or below the setpoint:



Figure 9: The image pyramid object.

$$u(t) = K_p \left(e(t) + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d \frac{de(t)}{dt}\right) \quad (2)$$

Proposed method

Voice interation with passengers

The voice interaction function consists of three phases. In the first phase, we build a text classification model by machine learning using the fasttext library (Joulin et al., 2016), an open-source library that allows users to learn text representations and text classifiers to classify text strings. Accordingly, we build keyword data and label the keyword combinations as follows:

According to Table 1, there are more than 100 labels in which each label represents a type of information provided to the user depending on the user's question.

In the second phase, the user's voice is recorded in real-time from the microphone, then using API functions to transmit to Google's Speech-to-Text API audio processing system. The text string will be returned immediately and put into the Text Classifier, as shown in Figure 8(a).

In the final phase, the text data converted from the passenger's voice will be classified by the model built in phase 1 to determine its label from which the computer program on the robot will respond correctly necessary information to respond to the passenger, as shown in the flowchart of Figure 8(b). As shown in Figure 8, in case the program does not determine which label the text belongs to, this text will be stored in a log file for us to check and add to the machine learning data.

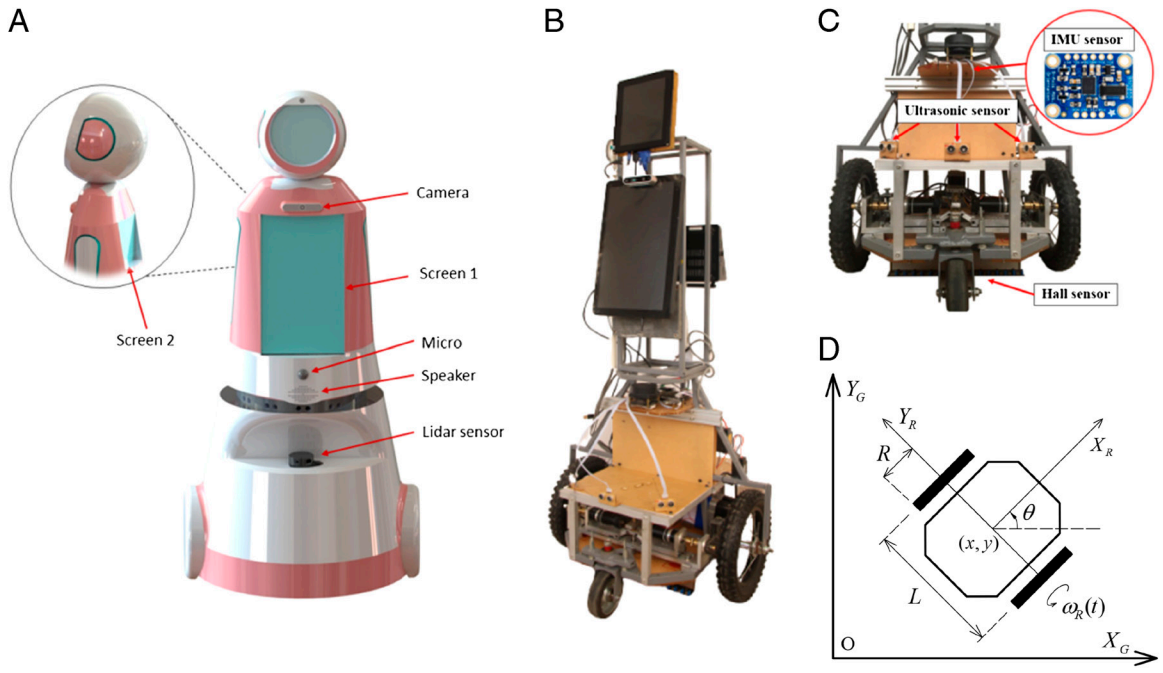

Figure 10: (a) P-net, R-net and image processing results of NMS. (b) O-net and face detection result.

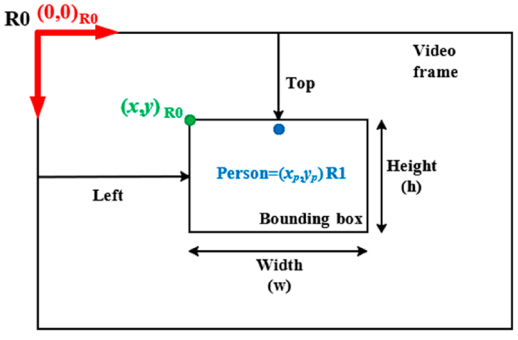

Figure 11: Coordinates and dimensions of the bounding box in the image.

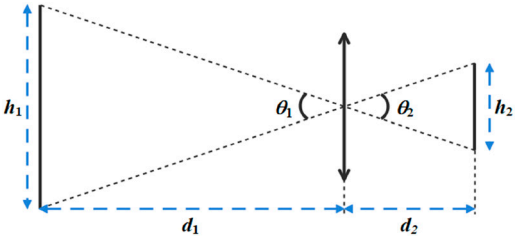

Figure 12: Two similar triangles opposite each other.

Face detection based on superimposed convolutional neural network and distance prediction technique

Face detection in images based on superimposed convolutional neural network

In practical, we often see a lot of people appearing in the lens, which will lead to many faces appearing in the frame with different sizes, as illustrated in Figure 9. Therefore, the MTCNN method to use image resizing creates a series of copies of the original with multiple dimensions called image pyramids (Hossain and Mukit, 2015).

We have each copy that will use the kernel 12×12 pixels and stride = 2 to scan the image for faces. MTCNN will recognize faces thanks to different dimensions. Next, we will pass the kernel through the P-net to find the coordinates of the four corners of each bounding box.

When removing the kernel and cells, we use two methods: NMS (Non-Maximum Suppression) to delete cells with overlapping percentages and set confidence Threshold—delete cells with low confidence level, as shown in Figure 10.

When we have found and deleted the unsuitable boxes, we convert the coordinates to the original

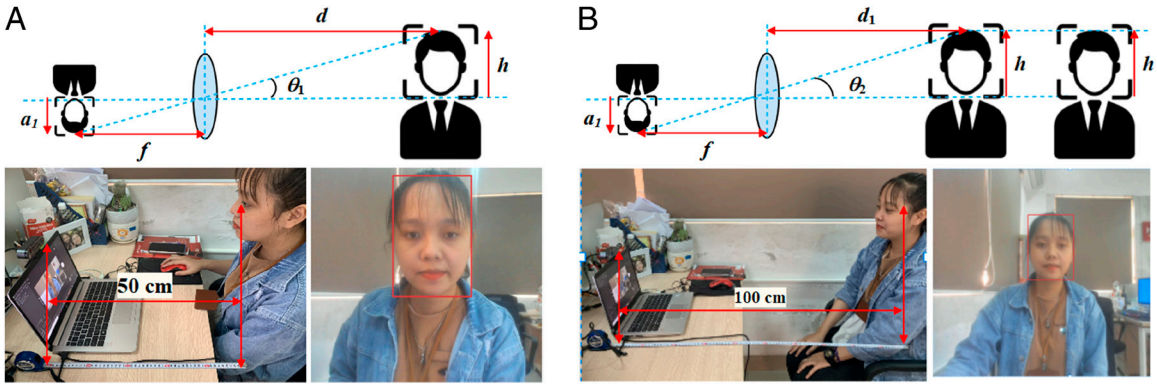

Figure 13: (a) Method of determining the focal length of the camera. (b) Determine the distance using the triangle similarity method.

image. Then calculate the length and width of the kernels based on the original image, multiply the coordinates when normalize to about (0,1) and add the coordinates corresponding to the kernel. The output results are the coordinates of the corresponding boxes, get the new coordinates for the R-net network.

R-net will do the same thing as P, but it will use the padding method to insert empty pixels into the missing bounding boxes. All boxes will resize and deliver the results over the O-net.

At O-net will resize the boxes to 48×48. The results return 3 values including 4 coordinates of the bounding box (*out [0]*), coordinates of 5 landmark points on the face shown in Figure 6, including 1 nose, 2 eyes, 2 sides of the mouth (*out [1]*) and the confidence score of each box (*out [2]*). After determining the coordinates of the boxes, we will determine the size of the boxes including width and height to support distance prediction.

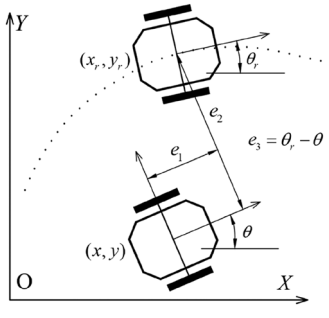

Figure 14: The positional error between the robot's actual coordinates and the reference coordinates in the trajectory.

According to Figure 11, we have a box covering the face with coordinates (*x*,*y*) and dimensions *h***w* where *h* is the height and *w* is the width (in pixels). We use Python language and OpenCV library for this test and extract values from the MTCNN library package (Choudhary et al., 2012; Jawad and Saleh, 2021).

Technique to predict the distance from the camera to the face

Disance prediction

This paper aims to develop a method for predicting the distance from the camera to the person in front of it based on the similarity of two triangles. In this study, the similarity is a property of the on-camera image sensor. The camera is the remaining side of two opposing triangles, then we have a pair of similar triangles that are symmetrical about the vertical axis, as illustrated in Figure 12. With angle $\theta_1 = \theta_2$ and $\frac{h1}{h2} = \frac{d1}{d2} = k$ where *k* is the scaling factor.

Applying the above property, we predict the distance from the sensor to the camera. Assume a model with a face of known height *h*, stands about *d* from the camera, as shown in Figure 13(a). Then the focal length *f* will be calculated as follows:

$$f = \frac{h}{d} \times a \quad (3)$$

In there: a is the height of the human face in the image, *f* is focal length, *h* is the height of the actual human face and d_1 is the distance from person to camera prism.

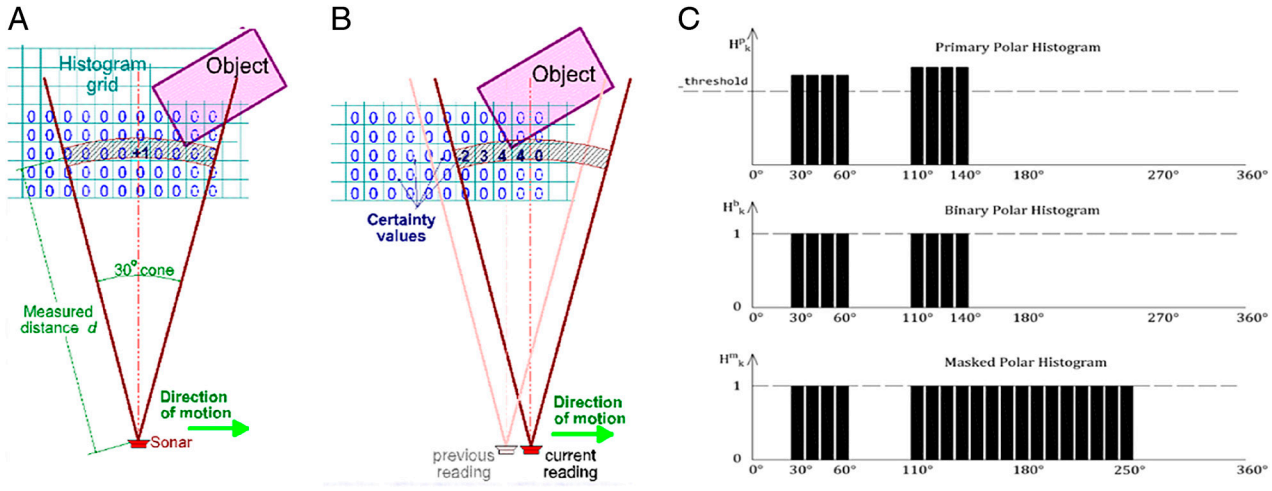

Figure 15: (a) Histogram grid. (b) enlargement angle from robot to obstacle. (c) Example of blocked directions.

Next, in Figure 13(b), the model moves closer to or away from the camera a certain distance, applying the principle of similarity of triangles to determine the distance d_1 according to the formula:

$$\frac{a_1}{f} = tan\,\theta_2 = \frac{h}{d_1} \quad (4)$$

Then the distance d_1 is calculated as

$$d_1 = h \times \frac{f}{a_1} \quad (5)$$

In there: a_1 is the height of the human face in the image, f is focal length, h is the height of the actual human face and d_1 is the distance from person to camera prism.

Calibration method

With the above distance prediction method, we found that there is still a factor leading to an error in the results since each person will have a different face, the size in the photo will also be different. Thus, the experimental process will be erroneous. To limit the error, for every 10 measurements we ask the user to approach the camera at a distance of 1m to recalibrate the *f*-parameter because here the focal length value of *f* is constant.

Control the robot to follow the trajectory and avoid obstacles

Control the robot to follow the trajectory

To accurately locate the robot's position in the operating environment, we apply the method as

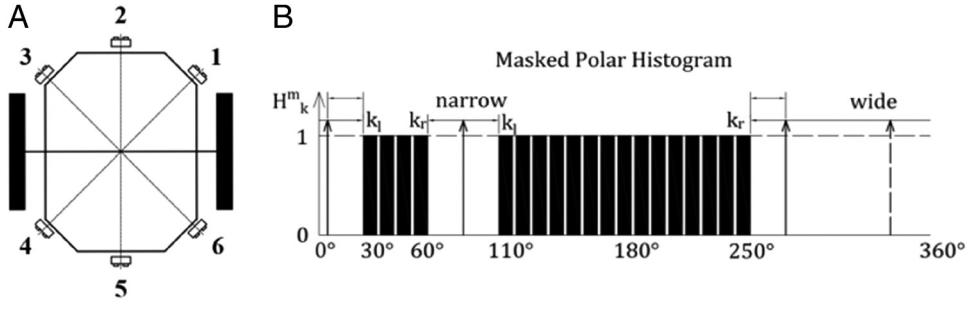

Figure 16: (a) Layout diagram of ultrasonic sensors. (b) Selecting the most optimal direction among candidate direction.

```
In  [91]: model.predict(ds.iloc[2, 0])
Out [91]: (('__label__#station_map#',), array([0.940898]))
```

Figure 17: Testing the model on a single sentence.

```
In  [85]: model.test('dia.train')
Out [85]: (1241, 0.8582, 0.8582)
```

Figure 18: Model's performance on the test set.

reported in Thanh et al. (2021). The main goal is to control the mobile robot to follow a certain trajectory. A different trajectory with a path with time constraints added to it, which makes the control target not only minimize the distance between the robot and the path, but also to ensure the travel time.

We define the actual robot state as: $X = [x\ y\ \theta]^T$ and according to the pattern trajectory is: $X_r = [x_r\ y_r\ \theta_r]^T$

When the robot moves, the error will appear:

$$e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix}. \quad (6)$$

From the kinetic and derivative model (4), we get the error model as follows:

$$\begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{e}_3 \end{bmatrix} = \begin{bmatrix} \cos e_3 & 0 \\ \sin e_3 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v_r \\ \omega_r \end{bmatrix} + \begin{bmatrix} -1 & e_2 \\ 0 & -e_1 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (7)$$

where v_r, ω_r, are the linear and angular velocity of the robot according to the trajectory.

The controller for the robot is built as follows:

$$\begin{aligned} v_{fb} &= k_1 e_1 \\ \omega_{fb} &= k_2 v_r \left(\frac{\sin e_3}{e_3}\right) e_2 + k_3 e_3 \end{aligned}, \quad (8)$$

where k_1 and k_3 are gain functions and k_2 is a constant gain which are calculated following De et al., (2001) and Klancar et al. (2005). Then the control law (6) for trajectory tracking will be rewritten as follows:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ \omega_r + k_2 v_r \left(\frac{\sin e_3}{e_3}\right) e_2 + k_3 e_3 \end{bmatrix}. \quad (9)$$

According to Klancar et al. (2005) with $k_1 > 0$, $k_2 > 0$, $k_3 > 0$, the tracking error e will be 0 when the time $t \to \infty$ follows the Lyapunov stability criterion. In Figure 14, the positional error between the robot's actual coordinates and the reference coordinates in the trajectory are illustrated.

Table 2. Test result data sheet.

Researcher	Number of correct classifications	Number of times of misclassification	Number of times unclassifiable
1	26	2	2
2	28	1	1
3	27	2	1
4	23	4	3
5	29	0	1

Table 3. The results of the experimental measurements of the non-calibrated program.

Number of measurements	Actual distance	Average prediction distance	Average error	Percent error average
1–10	1.2	1.19	0.01	0.83%
11–20	1.8	1.73	0.07	3.89%
21–30	2.4	2.01	0.39	16.25%

Obstacle avoidance using VFH + with ultrasonic sensors

Along the way, the robot must be able to detect and avoid unexpected obstacles. For these cases, a proxizmity sensor system is used. That is 12 ultrasonic distance sensors installed on the robot, as shown in Figure 14a, allowing to detect obstacles in front and two sides of the robot.

VFH + algorithm

The VFH + method uses a histogram grid to map the environment around the robot. This map is continuously updated with distance to obstacle data obtained from the ultrasonic sensors mounted on the robot, as shown in Figure 15. The method will find the optimal direction of movement when encountering obstacles, and appropriate velocity control for the robot (linear velocity, angular velocity).

- *The histogram grid:* a two-dimensional grid C (Histogram grid) that represents the obstacles in the world reference frame, is generated containing the information transmitted from the ultrasonic sensors (selected C has size 81 × 81 and resolution 0.1 m/cell). Each cell holds a certainty value between 0 and c_{max} A cell's certainty value is increased by 1 for each sensor reading that detects an obstacle in that cell.
- *The active window* C_a : A much smaller two-dimensional grid that follows the robot (the selected C_a is 33 × 33). Each cell holds an "obstacle vector" that consists of a magnitude $m_{i,j}$ and direction $\beta_{i,j}$, where $m_{i,j}$ is a function of the cells distance to the robot's center $d_{i,j}$ and corresponding ertainty value $c_{i,j}$ and $\beta_{i,j}$ as described by (1), is the direction from the robot's center to the cell:

$$m_{ij} = c_{ij}^2\left(a - bd_{ij}^2\right) \tag{10}$$

- *The primary polar histogram* H^p: a one-dimensional histogram of the angular sectors of width α so that $n = 360° / \alpha$ is an integer. Each sector holds a polar obstacle density which is the sum of the magnitude of all the cells in C_a that fall within that sector. An enlargement angle for

Table 4. Results of experimental measurements of the calibrated program.

Number of measurements	Actual distance	Average prediction distance	Average Error	Percent error Average
1–10	1.2	1.18	0.02	1.67%
11–20	1.8	1.76	0.04	2.22%
21–30	2.4	2.14	0.26	2.67%

Table 5. Compare the distance measured results with the images in the laboratory environment.

Light conditions	Outside light	Light of the light bulb
In the practice room	50%	75%

Table 6. Compare the distance measured results with outdoor images.

Light conditions	Harsh light	Weak outdoor light	Average outdoor light
Distance measurement ratio	52%	57%	90%

cells is also defined based on the robot's radius r_r and a parameter for minimum obstacle distance r_s, so a single cell can add to more than one sector.

- *The binary polar histogram* H^b: a one-dimensional histogram that maps each sector on H^p to 0 (free) or 1 (blocked) based on its value H_k^p. Two thresholds τ_{low} and τ_{high} are defined, $H_k^p < \tau_{low}$ then $H_k^p = 0$, if then $H_k^p = 1$, otherwise H_k^b remains unchanged from its previous value. Thus, the Binary Polar Histogram shows which direction is free and the robot will immediately change its direction of motion without encountering an obstacle.
- *The masked polar histogram* H^m: additional sectors in H^b are blocked based on the robot's direction of movement and minimum steering radius.
- Consecutive free sectors in H^m: are classified as wide or narrow valleys according to their size, and candidate directions for each valley are then added to a list.

Selection of the steering direction

The VFH + approach first determines a collection of probable candidate directions by finding all vacancies in the masked polar histogram. These candidate directions are then subjected to a cost function that considers more than simply the difference between the candidate and goal directions.

The candidate direction k_n with the lowest cost is then chosen to be the new direction of motion.

Selecting the most optimal direction among candidate directions in the first step, the right and left borders k_r and k_r of all openings. Similar to the original VFH method (Zhang and Wang, 2017), two types of openings are distinguished, namely, *wide* and *narrow* ones. As shown in Figure 16, an opening is considered wide if the difference between its two borders is larger than s_{max} time α (in our system $s_{max} = 16$ sectors). Otherwise, the opening is considered narrow. The only one candidate direction can be chosen for narrow openings:

$$c_n = \frac{(k_r + k_l)}{2} \tag{11}$$

For wide openings, three candidate directions are selected as

$$\begin{aligned} c_1 &= k_r + \frac{s_{max}}{2} \\ c_2 &= k_l - \frac{s_{max}}{2} \\ c_3 &= k_{target} if\ k_{target} \in [c_1, c_2] \end{aligned}, \tag{12}$$

where k_{target} is the target direction. We substitute the selected directions into the cost function:

$$g(c) = \mu_1 \Delta(c, k_{target}) + \mu_2 \Delta(c, k_\theta) + \mu_3 \Delta(c, c^{-1}) \tag{13}$$

With k_θ is current direction of the robot. The function *g(c)* with the lowest result will be the next optimal direction for the robot to move. The higher μ_1 is, the more goal-oriented the robot's behavior. The higher μ_2 the more the robot tries to execute an efficient path with a minimum change of direction of motion. The higher μ_3 is, the more the robot tries to head towards the previously selected direction and the smoother is the trajectory.

In order to o prioritize towards the goal, we choose

$$\mu_1 > \mu_2 + \mu_3. \tag{14}$$



Figure 19: (a) Passengers interact with the robot. (b) The robot leads the passenger to the check-in counter.

Experimental results

Voice interation with passengers

There are three methods to test our model. All of them are being slightly different. In the first test, we chose an observation which belonged to the category label "__label__#station_map#" and tested the model against it. As you can see below, it correctly predicted the category of "__label__#station_map#" and did so with a probability of 94%, as shown in Figure 17.

In the next test, we evaluate the keyword classifier on the entire dataset (1241 samples), which yielded values for precision at one as 0.85 and recall of one as 0.85 as well, as shown in Figure 18.

In both tests, the precision is the number of the correct labels predicted by the classifier among all the labels and the recall the number of labels successfully predicted among the real labels.

The third experiment is conducted in practice by setting up a microphone connected to a computer to receive voice with the participation of 5 researchers. Each person will repeat 2 times a group of keywords and continuously read 30 groups of keywords. After each reading, it will wait for the computer to respond to information before continuing to read the next keyword. This test method is intended to evaluate the audio acquisition, text-to-text, and classification capabilities of the program. The results of the test are shown in Table 2.

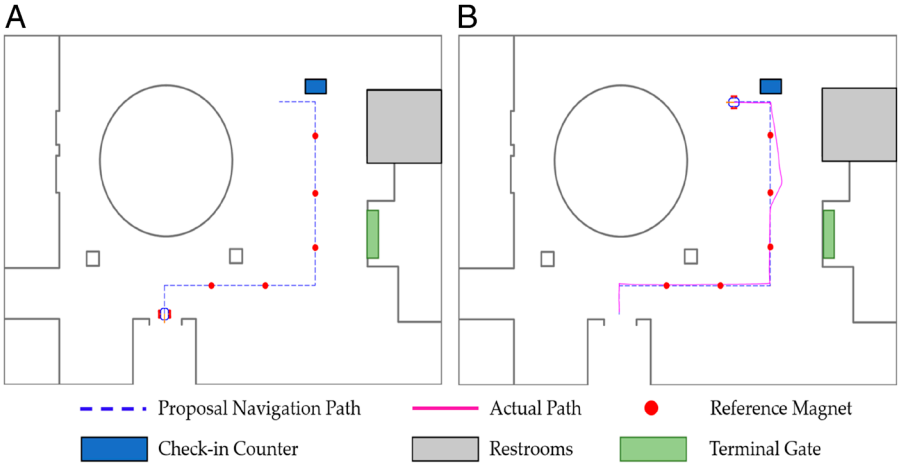

Figure 20: (a) Airport terminal map and robot path. (b) the actual path of the robot during passenger guidance.

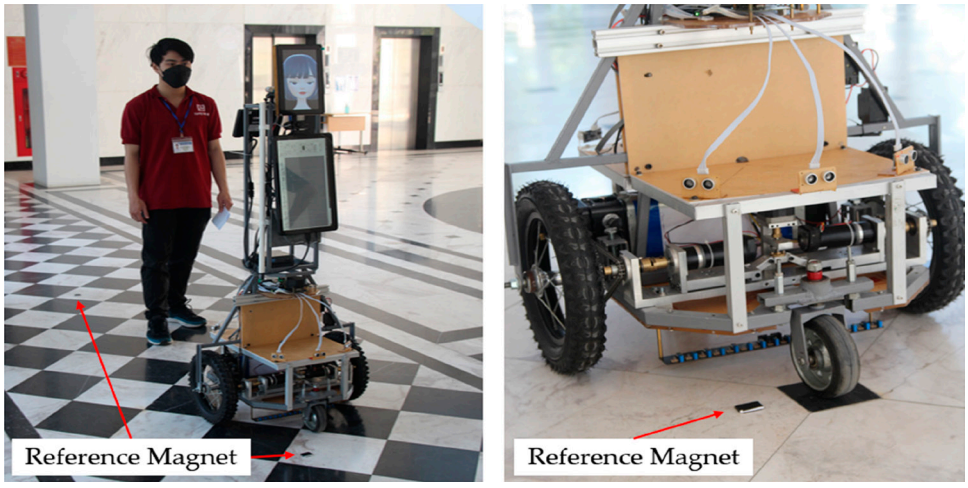

Figure 21: Position of the reference magnets on the floor.

According to the above table, the ability to receive and classify the text converted from the user's voice is quite good, the times of misclassification or failure to classify the label are determined by us due to the duplication of words. lock when the user reads a group of keywords with a small number of words, that is, a statement that is too short to contain the keywords. This can be solved by displaying on the computer screen prompts to ask questions to the passenger.

Passenger identification

In the first test, the program will test at three distances of 1.2 m; 1.8 m, and 2.4 m. For each distance, 10 measurements will be made, and the average data is in Table 3.

The measurement results in Table 3 show that at the shorter distance, the lower the percentage of error, the higher the accuracy of the measurement. The mean error of 30 measurements is 6.99%. In the second test, we test the program at three similar distances, while applying an additional calibration method. The results are shown in Table 4.

According to Table 4, the average error of 30 measurements is 3.52%. When applying the correction of the recalculation of the *f*-value, the accuracy of the 30 measurements increased, but not by much. The range that the program can also find faces in the image is from 0.5 m to 4 m. In actual conditions, this test will be affected by ambient conditions, especially lighting. Using light in the laboratory will give two different results in Table 5. The resulting image can still be used for detection and distance measurement.

Here we will have a table comparing the images obtained from the environments.

Table 6 shows that average lighting conditions will give the best results. The inspection in the corridor also depends a lot on the light intensity at that time, if in cloudy weather conditions. Therefore, image acquisition for distance measurement is also greatly affected by ambient light conditions.

Experimenting with a robot that guide passengers and avoids obstacles

The robot was tested in a simulation of a miniature airport terminal with a size of about 20 m×20 m. In the test environment, there are full locations such as doors, departure gates, check-in counters, coffee shops. The starting position of the robot will be at the door. When someone communicates with the robot and asks the robot to lead the way to the desired location on the map, the robot will map out the most optimal trajectory and will lead the passenger, as shown in Figure 19. The screen behind the robot will show passengers the way to go. The rear camera is also used to identify and determine if the passenger is still following the robot, if the passenger does not follow the robot, the robot will return to the starting position.

On the floor of the airport terminal, there are magnets in fixed positions so that the robot can locate. When the robot guides, it will find the optimal path through these reference magnet points and closest to the current position of the robot. The position of the magnets can be seen in Figures 20 and 21. The distance between the magnets is optimized so that

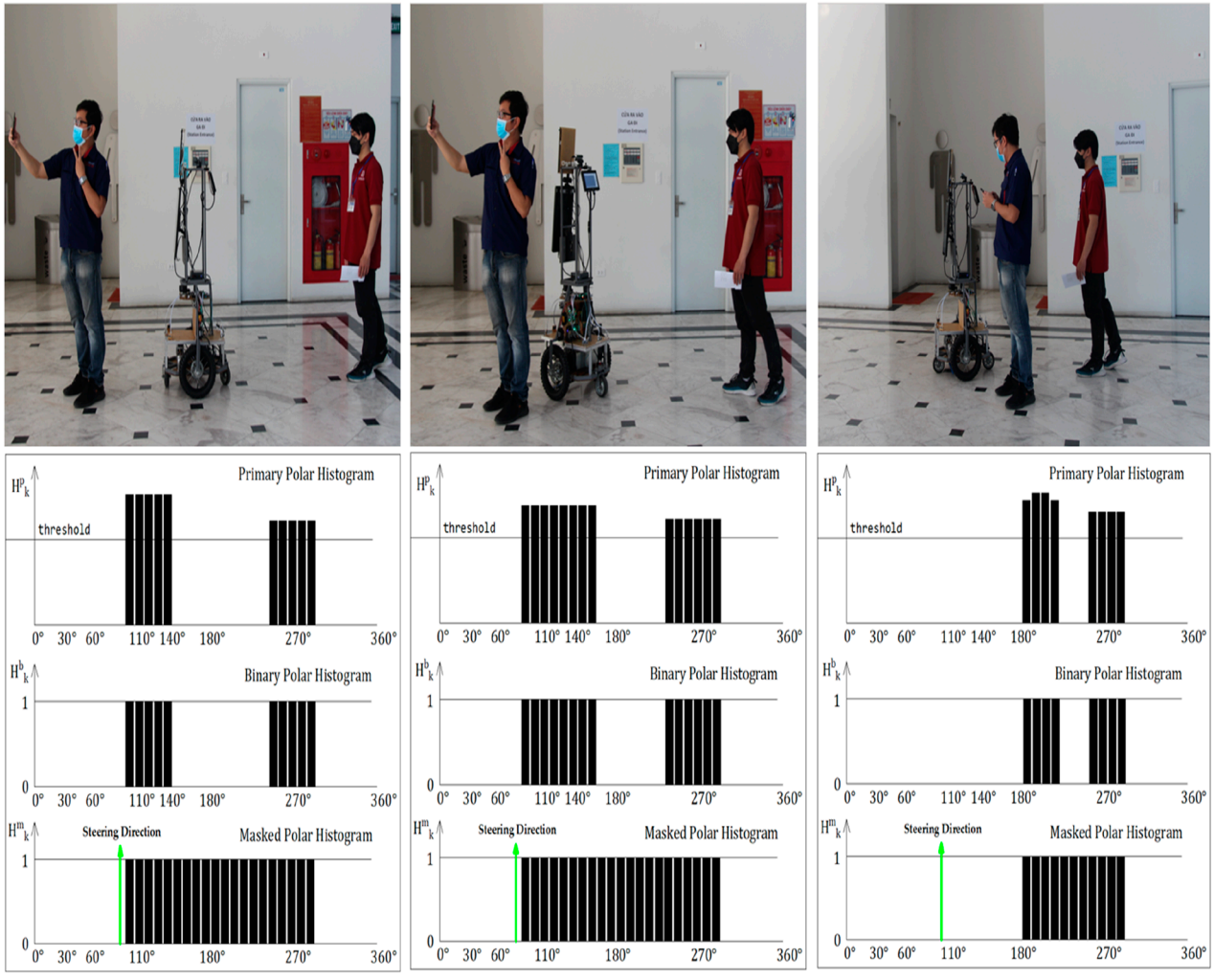

Figure 22: Human avoidance robot process and VFH + polar histograms.

the robot can be prevented from deviating from its orbit during movement, here the distance of the magnets is about 3m.

In the process of moving, if the robot encounters an obstacle or a person crosses the trajectory, the robot uses the VFH + method with distance data from the ultrasonic sensor.

Obstacle avoidance results are shown in Figure 22. The histograms of the VFH + method and the selected direction, respectively, at the positions of the robot during obstacle avoidance.

Conclusions and future work

This paper provides a framework of building an intelligent robot system to assist passengers in smart airports that can be expanded other areas, especially in COVID-19 situations. The system is built on an autonomous intelligent mobile robot system, which is simulated on a departure terminal. To ensure the accuracy of moving in the desired trajectory when there is no (or weak) GPS signal, new positioning algorithms using Ultra Wideband technology with the synthesis of sensor data from lidars, and encoder sensors are proposed. The program that applies segmentation and feature point extraction algorithms to meet the requirements of local environment mapping and avoidance of obstacles is also proposed. The services interact between the robot and the passenger by voice communication combined with machine learning techniques to analyze and understand the requirements of the passengers. In addition, the face detection technique is based on superimposed convolutional neural network to predict the distance between robot

and passenger to perform the function of guiding passengers to areas in the airport. The tasks to assist airline passengers have been surveyed, simulated surveys, and evaluated experimental results and proven the effectiveness of the proposed methods.

The next quantitative studies will be applied to the installed robot system, which promises to have more useful results contributing to the research field of mobile robots, a potential product introduced in the market. to work in locations where information is essential but may be scarce or ever-changing. They are especially useful in places where customers are crowded and hurry, such as airports and transport hubs, information desks, shopping malls, medical services, etc.

Acknowledgments

The authors would like to thank Thai Nguyen University of Technology, Viet Nam for the support.

Literature Cited

Ahmad, H., Othman, N. A. and Ramli, M. S. 2018. A solution to partial observability in extended kalman filter mobile robot navigation. *Telkomnika* 16.1:134–141.

Choudhary, B. K., Sinha, N. K. and Shanker, P. 2012. Pyramid method in image processing. *Journal of Information Systems and Communication* 3 (1): 269.

De, L. A., Oriolo, G. and Vendittelli, M. 2001. Control of wheeled mobile robots: An experimental overview. *Ramsete*:181–226.

Garcia-Haro, J. M., Oña, E. D., Hernandez-Vicen, J., Martinez, S. and Balaguer, C. 2020. Service robots in catering applications: a review and future challenges. *Electronics* 10.1:47.

Hossain, M. A. and Mukit, M. 2015. A real-time face to camera distance measurement algorithm using object classification. 2015 International Conference on Computer and Information Engineering (ICCIE), 107–110.

Ivanov, S. H., Webster, C. and Berezina, K. 2017. Adoption of robots and service automation by tourism and hospitality companies. *Revista Turismo & Desenvolvimento* 27 (28): 1501–1517.

Jawad, S. R. and Saleh, M. H. 2021. Online 3D path planning for Tri-copter drone using GWO-IBA algorithm. *Telkomnika* 19(4):1334–1341.

Joosse, M., Lohse, M. and Evers, V. 2015. How a guide robot should behave at an airport insights based on observing passengers. CTIT Technical Report Series, no. TR-CTIT-15–01.

Joulin, A., Grave, E., Bojanowski, P., Douze, M., Jégou, H. and Mikolov, T. 2016. Fasttext. zip: Compressing text classification models. arXiv 1612: 03651.

Klancar, G., Matko, D. and Blazic, S. 2005. Mobile robot control on a reference path. Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control 2005: 1343–1348.

Li, H., Lin, Z., Shen, X., Brandt, J. and Hua, G. 2015. A convolutional neural network cascade for face detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 5325–5334.

Meidute-Kavaliauskiene, I., Yldz, B., Çiğdem, Ş. and Činčikaitė, R. 2021. The effect of COVID-19 on airline transportation services: a study on service robot usage intention. *Sustainability* 13 (22): 12571.

Muñoz Peña, K. and Bacca Cortés, B. 2020. GUI3DXBot: an interactive software tool for a tour-guide mobile robot. *Ciencia E Ingenieria Neogranadina* 30 (1): 59–74.

Nguyen, M. T. and Teague, K. A. 2015. Compressive and cooperative sensing in distributed mobile sensor networks. MILCOM 2015–2015 IEEE Military Communications Conference, pp. 1033–1038, doi: 10.1109/MILCOM.2015.7357581.

Nguyen, M. T. and Boveiri, R. H. 2020. Energy-efficient sensing in robotic networks. Measurement, 158, p. 107708

Nguyen, M. T., La, H. M. and Teague, K. A. 2018. Collaborative and compressed mobile sensing for data collection in distributed robotic networks. *IEEE Transactions on Control of Network Systems* 5 (4): 1729–1740.

Nguyen, M. T., Nguyen, C. V., Do, H. T., Hua, H. T., Tran, T. A., Nguyen, A. D., Ala, G. and Viola, F. 2021. UAV-assisted data collection in wireless sensor networks: a comprehensive survey. *Electronics* 10(21):2603.

Pathi, S. K., Kiselev, A., Kristoffersson, A., Repsilber, D. and Loutfi, A. 2019. A novel method for estimating distances from a robot to humans using egocentric RGB camera. *Sensors* 19 (14): 3142.

Qingqing, L., Fu, Y. J., Pena, Q., Tuan, N. G., Hannu, T., Zhuo, Z. and Tomi W. 2019. Edge computing for mobile robots: multi-robot feature-based lidar odometry with FPGAs. 2019 Twelfth International Conference on Mobile Computing and Ubiquitous Network (ICMU), pp. 1–2, doi: 10.23919/ICMU48249.2019.9006646.

Rivai, M., Hutabarat, D. and Nafis, Z. M. J. 2020. 2D mapping using omni-directional mobile robot equipped with LiDAR. *Telkomnika* 18(3):1467–1474.

Sendari, S., Afandi, A. N., Zaeni, I. A. E., Mahandi, Y. D., Hirasawa, K. and Lin, H. I. 2019. Exploration of genetic network programming with two-stage reinforcement learning for mobile robot. *Telkomnika* 17(3):1447–1454.

South China Morning Post. 2017. https://www.scmp.com/news/asia/east-asia/article/2101253/meet-troika-chinese-speaking-robot-helping-tourists-south-korean.

Thanh, V. C., Quan, N. N. A., Dong, T. L. T., Hoang, T. T. and Nguyen, M. T. 2021. Fusion of inertial and magnetic sensors for autonomous vehicle navigation and freight in distinctive environment. International Conference on Engineering Research and Applications, pp. 431–439.

Tran, H. T., Thanh C. V., Dong, L. T., Quan, N. N., Duyen, M. H., Quang, N. P., et al. 2022. Extended Kalman filter (EKF) based localization algorithms for mobile robots utilizing vision and odometry. 2022 IEEE 21st Mediterranean Electrotechnical Conference (MELECON), pp. 91–96, doi: 10.1109/MELECON53508.2022.9843066.

Triebel, R., Kai, A., Rachid, A., Lucas, B., Stefan, B., Raja, C., Mohamed, C., et al. 2016. Spencer: a socially aware service robot for passenger guidance and help in busy airports. *Field and service robotics*, pp. 607–622.

Zhang, Y. and Wang, G. 2017). *An improved RGB-D VFH + obstacle avoidance algorithm with sensor blindness assumptions.* 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE), pp. 408–414, doi: 10.1109/ICRAE.2017.8291420.