# Safe Interaction Between Human and Robot Using Vision Technique

Ha Quang Thinh Ngo[1,2](✉) and Dang Quy Phan[1,2]

[1] Department of Mechatronics, Faculty of Mechanical Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam
nhqthinh@hcmut.edu.vn

[2] Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc City, Ho Chi Minh City, Vietnam

**Abstract.** In recent years, service robot or assistant robot becomes more and more popular since it appears in most of fields in our lives. Together with the significant development of technology, robot does not only serve the daily requirements but also begin to be a closed partner which could carry the bulky goods for employer, inventory for worker or the heavy luggage for tourists. In this paper, a method to avoid obstacles in the presence of human is introduced. Initially, the idea to autonomously navigate in the crowded environment is based on visual approach. Then, a model of human to reach closely is innovated for autonomous robot. The interactive parameters and the concept of social communication are integrated so that robot could improve its awareness about human. To verify our approach, several experiments are launched in the indoor environment. Three test cases are depicted to imitate the real-world situation. From these results, it could be seen clearly that the proposed scheme is effective, feasible and proper for autonomous robot to avoid obstacles.

**Keywords:** Safe interaction · Autonomous robot · Motion control · Robotics

## 1 Introduction

Autonomous robot is a type of mobile robot that is capable of operating on its own, performing tasks without human intervention. With sensors, they have the ability to perceive their surroundings. Autonomous robots are increasingly meaningful in industries, commerce, medicine, scientific applications and serving human life. With the development of robotics, these robots are more and more capable of operating in different environments, depending on the field of applications, they have many different types such as painting robots [1], welding robots [2], lawn mowing robots [3], etc. ocean exploration robot [4], space work robot [5]. Along with the development of requirements in practice, autonomous robots continue to present new challenges for researchers.

The problem of autonomous robots is how they can operate separately, recognize the environment and perform the tasks set out. The first problem is to move, how the


C. V. Phan and T. D. Nguyen (Eds.): ICCASA 2022, LNICST 475, pp. 162–177, 2023.
https://doi.org/10.1007/978-3-031-28816-6_13

autonomous robot should move, and which moving mechanism is the best choice. Navigation [6] is a fundamental issue in research and construction of autonomous robots. In the research association of autonomous robots, there are two different research directions. In the first direction, research on autonomous robots focuses on navigating at high speed due to information obtained from sensors [7]. This is a type of robots which is capable of operating in both indoor and outdoor environments. It requires the greatly computing power, and is equipped with a high-sensitivity sensor and a large measuring range to be able to control the robot to move at high speed, in environments with complex terrain. For the second direction, it is to solve the problems of autonomous robots that are only used to operate in the room environment [8]. They have a simpler structure than the above type, as well as these robots perform simple tasks.

Navigation challenges for autonomous robots are divided into two categories such global problems and local problems. In the global challenges [9], the working environment of the robot is completely determined, the path and obstacles are completely known in advance. In the local challenges [10], the operating environment of the robot is unknown or only partially known. Sensors and positioning devices allow the robot to identify obstacles, its position in the environment, and help it reach the target.

However, navigating problems for autonomous robots are often not the same as for other types of robots [11, 12]. To be able to steer the autonomous robots, decisions in real-time must be based on continuous information about the environment through sensors, either in indoor or outdoor environments, which is the greatest difference compared to with offline scheduling techniques. Autonomous robots must be able to decide on their own about navigation methods and motion orientations to be able to reach the destination to perform certain tasks.

Navigating an autonomous robot is a job [13] that requires a number of different abilities, including: basic mobility, such as going to a given location; the ability to react to events in real time, such as the sudden appearance of an obstruction; the ability to create, use and maintain an operational environment map; the ability to determine the position of the robot in that map; the ability to make plans to reach a destination or avoid undesirable situations and the ability to adapt to changes in the operating environment.

## 2 Vision-Based Library and Digital Camera

PCL [14] is a support library for n-D Point Cloud and for image processing in 3D space. The library is built with many algorithms such as filtering, surface reconstruction, segmentation, feature estimation, … PCL can be used on many platforms such as Linux, MacOS, Windows and Android. To simplify the development, PCL is broken down into several smaller libraries that can be compiled individually. PCL is completely free for the research or development of commercial products. It could be said that PCL is a combination of many small modules. These modules are essentially libraries that perform individual functions before being packaged by the PCL. These basic libraries are:

- Eigen: an open library that supports linear operations, used in most of PCL's mathematical calculations.

- FLANN: (Fast Library for Approximate Nearest Neighbors) supports fast search of neighboring points in 3D space.
- Boost: helps to share pointers across all modules and algorithms in PCL to avoid duplicating data that has been retrieved in the system.
- VTK: (Visualization Toolkit) supports many platforms in obtaining 3D data, supporting the display and estimation of object volumes.
- CMinPack: an open library for solving linear and non-linear maths.

Kinect as Fig. 1 is a Microsoft product based on camera technology developed by PrimeSense, the first products sold in North America on November 4, 2010. Kinect is considered as a peripheral for Xbox 360, allowing to communicate with people through gestures, bringing exciting feelings to Xbox gamers. Kinect's ability to understand human gestures is based on two main features: depth map information, the ability to detect and track human body characteristics (body skeleton tracking).

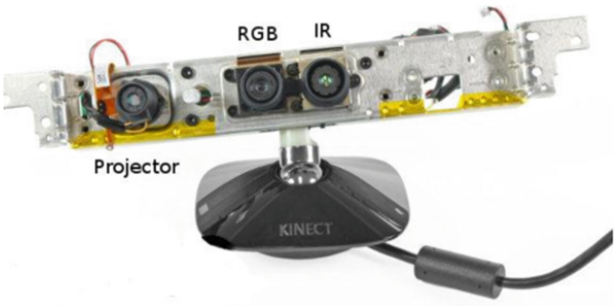



**Fig. 1.** Inside structure of Kinect camera.

Kinect includes: RGB camera, depth sensor (3D Depth Sensors), microphone array (Multi-array Mic) and motorized tilt angle control (Motorized Tilt).

- RGB Camera: like a normal camera, has a resolution of $640 \times 480$ at 30 fps.
- Depth sensor: depth is obtained by the combination of two sensors: infrared illuminator (IR Projector) and infrared camera (IR camera).
- Multi-microphone array: includes four microphones arranged along the Kinect as shown in the picture, used for voice control applications.
- Lift angle control motor: is a fairly small DC motor, allowing us to adjust the camera up and down to ensure the camera has the best viewing angle.

Differentiating from the stereo camera technique which uses the same pair of cameras to build a depth map, or the Time-Of-Flight (TOF) technique, that defines the distance by estimating the travel time of the light ray going back and forth, Light Coding technique uses a continuously projected infrared light source combined with an infrared camera to calculate the distance. This computation is done inside the Kinect using PrimeSense's PS1080 SoC chip. This new technology is said to be more accurate and cheaper for usage in indoor environments.

## 3 Proposed Algorithm

In practice, the working environment of robot is defined by a matrix $M_{m \times n}$. Each component in location (x,y) from origin is performed by 0 or 1. It is assumed that there are N persons who locate around robot, matrix $P = \{p_1, p_2, p_3, \ldots, p_N\}$ with $p_i$ in corresponding to the i$^{th}$ person.

The framework of safety and comfort for human in the social context is launched by 6 functional blocks as Fig. 2 such human states extraction, extended personal space, social interaction detection, social interaction space, dynamic social zone, and approaching humans).

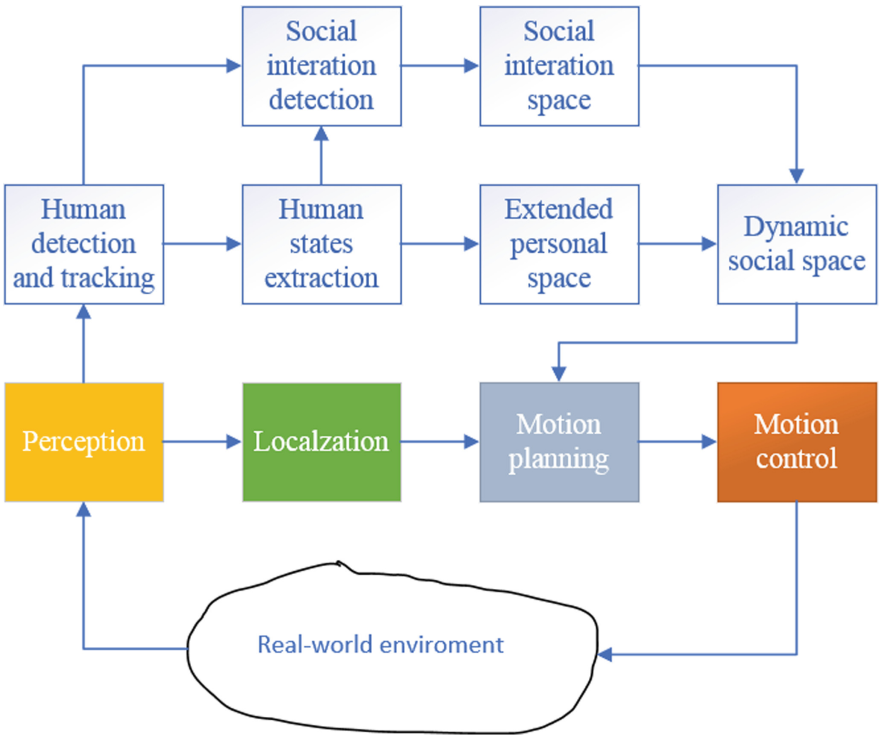



**Fig. 2.** Block diagram of overall system

*A. Comfortable and Safe Zone for Human*

Human states play an important role in the framework of human safety and comfort since they provide both spatial and temporal characteristics of the person. Relationship between robot and human illustrates the interactive patterns of the human state including position, gaze direction, velocity, human hand posture, and vision.

The field of view (FoV) from one person is also important to response what human see, where and whom stays in this environment. In this article, it is defined by a vector for field of view $H_i$ for a person $p_i$ which vector $H_i$ has the direction as $\theta_i^{pv}$. FoV for person $p_i$ is $2\alpha$ which is computed by both left side and right side.

The pose of human is considered as an important source to identify the human's activities. It is essential for robot to improve its awareness. Though, to simplify the input data, the information of human's hand is utilized. The locations of left hand and right hand of human $p_i$ in the xy plane are symbolized by $(x_i^{lp}, y_i^{lp})$ and $(x_i^{rp}, y_i^{rp})$. The magnitude $(d_i^{lp}, d_i^{rp})$ and direction $(\theta_i^{lv}, \theta_i^{rv})$ of vector of hand starting from center point of person $p_i$ to left hand and right hand are illustrated as

$$d_i^{lh} = \sqrt{\left(x_i^p - x_i^{lh}\right)^2 + \left(y_i^p - y_i^{lh}\right)^2} \tag{1}$$

$$d_i^{rh} = \sqrt{\left(x_i^p - x_i^{rh}\right)^2 + \left(y_i^p - y_i^{rh}\right)^2} \tag{2}$$

$$\theta_i^{lh} = tan^{-1}\left(\frac{y_i^{lh} - y_i^p}{x_i^{lh} - x_i^p}\right) \tag{3}$$

$$\theta_i^{rh} = tan^{-1}\left(\frac{y_i^{rh} - y_i^p}{x_i^{rh} - x_i^p}\right) \tag{4}$$

*B. Basic Personal Zone*

The Gaussian function in 2d is deployed to depict the personal zone. In Fig. 3, this area of person $p_i$ is determined by function $f_i^p(x, y)$ with the largest value at center point $\left(x_i^p, y_i^p\right)$ and gradually decreased around person $p_i$.

$$f_i^p(x, y) = A^p e^{-\left(\frac{dcos\left(\theta - \theta_i^p\right)}{\sqrt{2}\sigma_0^{px}}\right)^2 + \left(\frac{dsin\left(\theta - \theta_i^p\right)}{\sqrt{2}\sigma_0^{px}}\right)^2} \tag{5}$$

with d, $\theta$, $\theta_i^p$, it could be estimated as

$$d = \sqrt{\left(x - x_i^p\right)^2 + \left(y - y_i^p\right)^2} \tag{6}$$

$$\theta = \tan^{-1}\left(\left(y - y_i^p\right), \left(x - x_i^p\right)\right) \tag{7}$$

$$\theta_i^p = \{ \begin{matrix} \theta_i^{pv} n\hat{e}'u\ v_i^p > 0 \\ \theta_i^{ph} n\hat{e}'u\ v_i^p = 0 \end{matrix} \tag{8}$$

(x,y) is the location of current point in the matrix $M_{(nxm)}$, $\left(x_i^p, y_i^p\right)$ and $\theta_i^p, \theta_i^{pv}, \theta_i^{ph}$ are the position, direction, moving direction, FoV of person $p_i$. $A^p$ is the selected amplitude. $\sigma_0^{px}, \sigma_0^{vx}$ is the standard deviation of the Gaussian function. A set of three main parameters $\left[A^p, \sigma_0^{px}, \sigma_0^{vx}\right]$ is utilized in function $f_i^p(x, y)$.

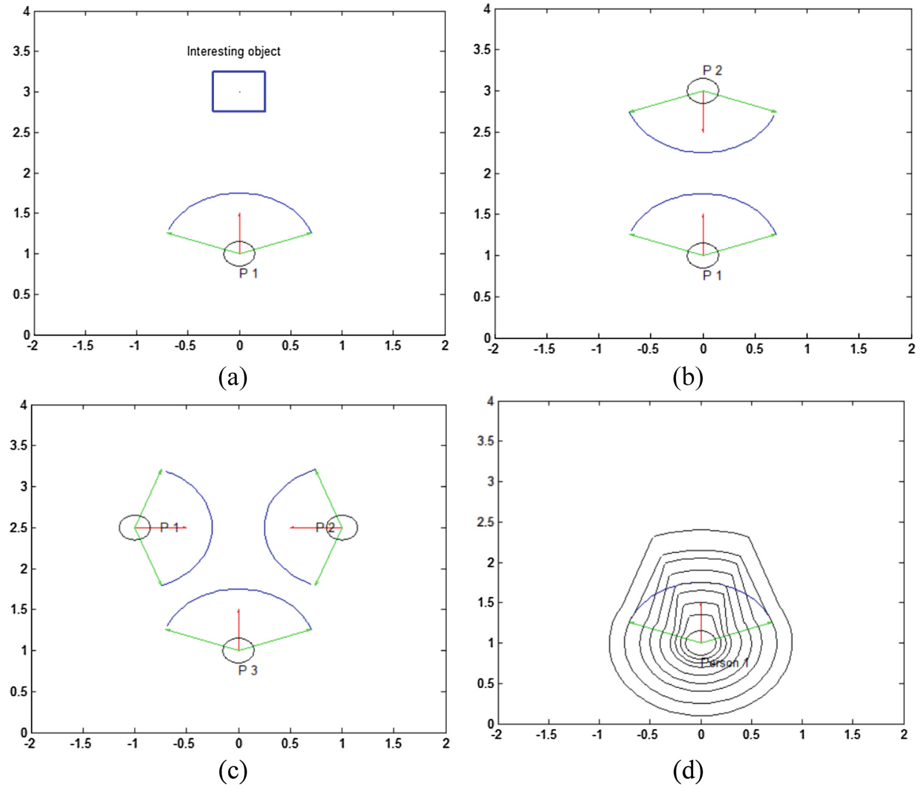



**Fig. 3.** Model of human and some interactions.

*C. Extended Personal Zone*

The size and profile of the basic personal zone depends on these parameters $A^p$, $\sigma_i^{px}$, $\sigma_i^{vx}$. The environment surrounding a person could be classified into two parts, for example (1) front view including FoV of human and (2) rear view. In the front view, some parameters are well-defined as $f_v, f_{front}$ *and* $f_{fov}$ consisting of $\sigma_i^{py}$.

In order to embed the information from human's hand, the distance is modeled as Eq. (5). $f_i^{lh}(x, y)$ represents the space of left hand, centered at $\left(x_i^p, y_i^p\right)$, direction $\theta_i^{lh}$ are evaluated as Eq. (3). Then, the parameters are set for a model of left hand $\left[A^p, \sigma_0^{hx}, \sigma_0^{lhy}\right]$ which $\sigma_0^{hx}$ is determined in advance and $\sigma_0^{lhy}$ is recognized as

$$\sigma_0^{lhy} = \left(1 + f_h d_i^{lh}\right)\sigma_0^{hy} \tag{9}$$

where $d_i^{lh}$ is a distance from location of human to left hand, $f_h$ is the coefficient of normalization and $\sigma_0^{hy}$ is known.

Similarly, function $f_i^{rh}(x, y)$ characterizes the space of right hand, centered at $\left(x_i^p, y_i^p\right)$, direction $\theta_i^{rh}$. Later, the setting parameters for a model of right hand

$\left[A^p, \sigma_0^{hx}, \sigma_0^{rhy}\right]$ which $\sigma_0^{hx}$ is known and $\sigma_0^{rhy}$ is

$$\sigma_0^{lhy} = \left(1 + f_h d_i^{lh}\right)\sigma_0^{hy} \tag{10}$$

where $d_i^{rh}$ is a distance from location of human to right hand, $f_h$ is the coefficient of normalization and $\sigma_0^{hy}$ is known.

The extended personal space is estimated as

$$f_i^{eps}(x, y) = \max\left(w_1 f_i^p(x, y), w_2 f_i^{lh}(x, y), w_2 f_i^{rh}(x, y)\right) \tag{11}$$

(x,y) is the coordinate in matrix, $w_1$ is the weight of personal zone and $w_2$ is the weight of both left hand and right hand. Function $F_{eps}(x, y)$ that denotes for the extended personal zone of every person surrounding robot, is evaluated

$$F_{eps}(x, y) = \max\left(f_1^{eps}(x, y), \ldots, f_N^{eps}(x, y)\right) \tag{12}$$

*D. Social Personal Zone*

In the social environment, human behavior is influenced by other individuals or objects around, especially the objects with which people are interacting. To ensure human safety, mobile robots need to be aware of the context of social interactions. The space around the robot, social interaction should be classified according to the types of conditions of human-object interaction or interaction between a group of people.

In a real-world environment, it is paid attention to special objects such as televisions, refrigerators and phones in homes, screens in airports or paintings in museums. Depending on the context of human-object social interaction, the robot has to estimate the human-object interaction because that is the key to define the space of human-object interaction.

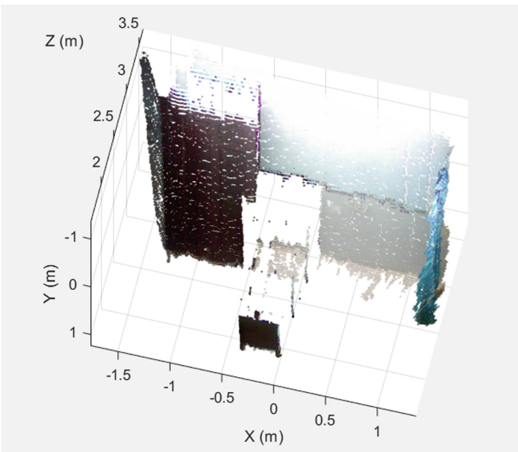



**Fig. 4.** Modeling of working environment by using PCL library and Kinect.

*E. Modeling of Working Environment*

To be able to model the operating environment, it must be known the position of objects in the space in front of the robot. As Fig. 4, it should use the depth camera such Kinect which is a pair of infrared transceivers to be able to recognize the position from the robot to the object in space.

(a) (b)

**Fig. 5.** Example of depth map obtained from Kinect, (a) practical image, (b) depth map.

Therefore, the usage of Point Cloud Library (PCL) is recommended since it carries out the results with high accuracy, the position of objects on the 3-dimensional coordinate axis is intuitive, making it easy to locate objects. Because of the limited capacity of host computer, it must have to calculate the environment based on Kinect's depth camera as Fig. 5. Kinect's depth map returns a 480 × 640 matrix with the value of each pixel being the depth value of that point in mm.

The data from depth map (D) is gained and stored into the environmental matrix (M) 480 × 640. In the x axis, measured values are the depth in cm unit

$$x_M = \frac{x_D}{10} \tag{13}$$

In the same way, these values in the y axis are in cm unit. The unit of size is converted from pixel to cm via the extended angle of depth map

$$y_M = \frac{y_D - 320}{320} * \tan\left(29 * \frac{pi}{180}\right) * x_M \tag{14}$$

In Fig. 6, white color indicates the obstacles which values are 1 while the others are 0. After inserting the estimated values, the environmental matrix M has been shown.

*F. Segmentation*

Data clustering (or just clustering), also known as cluster analysis, is the segmentation analysis, categorical analysis or unsupervised classification. It is a method of creating groups of objects or clusters. The 2D point cloud in the previous section will require an efficient data clustering method.

**Fig. 6.** Result of the image processing data from depth map.

In this stage as Fig. 7, it might use the association of pixel points on the environmental matrix to define separate objects in space in order to create a moving environment for the robot. This process is to filter pixels that have no association with 4 pixels around them to remove noise and get only the object where each pixel is associated with at least 4 pixels around it.

*G. Identification of the Bounded Objects*

Firstly, it is considered to define some rough contours of objects in space. This contour is derived from the convex vertices of each object and they are joined to form the contour for each object. The contour is a set-in real vector space V such that for two points x, y, hence the line segment [x, y] is also contained in the set. But this contour also includes the concave parts of the object and it occupies quite a large space. So, it must have to find a way to concave this space line to be able to reduce the occupied area thereby increasing the active area. For a given discrete data point, the convex contour can be uniquely determined: find the polygon whose vertices are a subset of the data points and maximize the area while minimizing the circumference. In contrast, the concave shell definition will not define a single polygon: find a polygon whose vertices are a subset of the data points and minimize area and perimeter simultaneously. It is clear that minimizing both area and circumference at the same time is a conflicting goal, and it is therefore possible to define many different concave shells for a given data point.

For the ambient matrix, the algorithm is applied to find the convex contour of the object. To do this, the standard set of Matlab functions was used to return the set of convex polygon vertices in CCW or CW order. Figure 8(left) shows the result of applying the aggregation function to each data cluster presented in Fig. 8(right) to obtain the corresponding convex bodies.

**Fig. 7.** Segmentation of objects in environment.

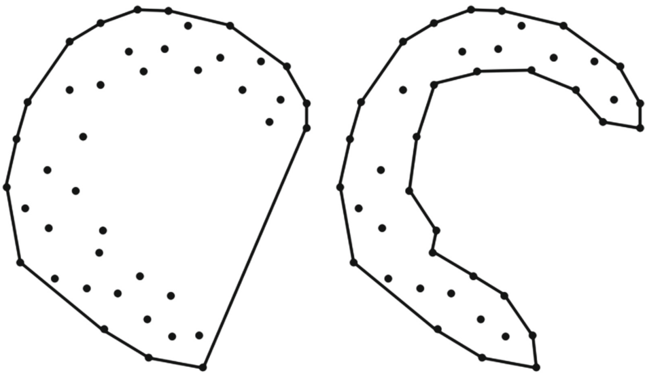


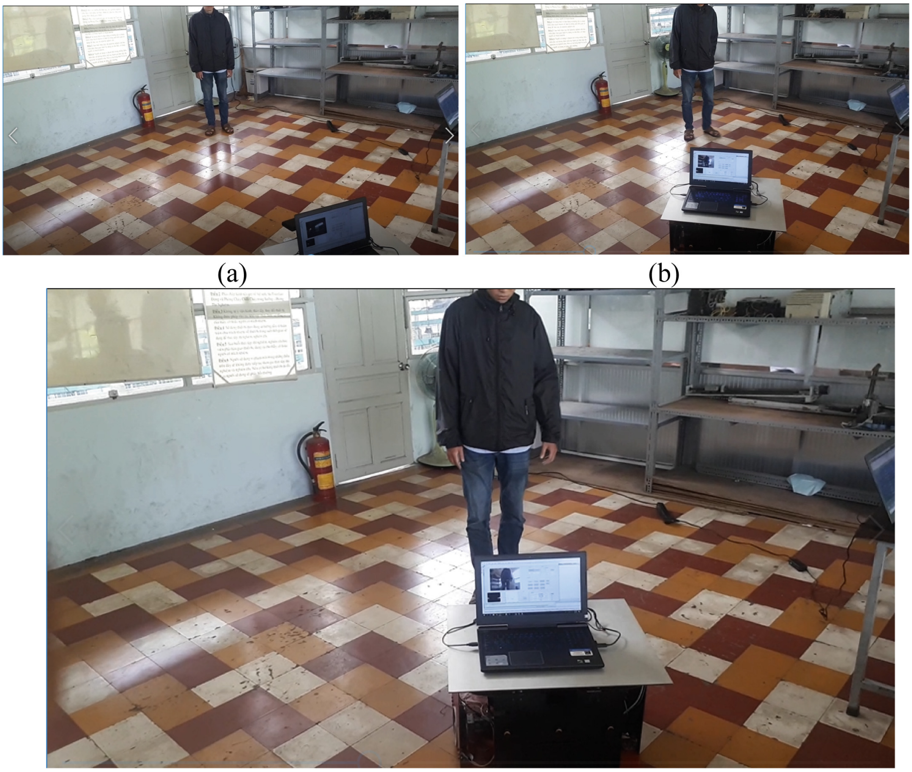
**Fig. 8.** Differences between convex-hull and concave-hull with 2D points.

The algorithm for calculating the concave shell of a dataset is based on the idea presented by Peter Wasmeier [15], who implemented a function in Matlab called hullfit. First, it runs the complex function to obtain the vertices of the corresponding convex body and arranges them clockwise. It then performs a distance calculation between all neighboring vertices to find any distance greater than the maximum allowed distance. If it finds any matching distance then it uses the following two conditions to find another suitable point in the data set as the next vertex:

1. Since we know that the contour definition is clockwise, the next selected vertex is the point with the smallest positive angle with which the contour intersects.
2. The distance to a vertex must be less than the length of the line.

## 4 Results in Experiments

To validate the proposed approach, several experiments have carried out in our laboratory. The type of robot is WDD (Wheeled Differential Drive) which provide the flexible motion and stable movement. It has two side wheels for driving while both front and rear wheel are to balance. Robot turns left or right owing to the differences between left wheel and right wheel correspondingly. The driving actuators are two DC servo motors 100 W which is directly connected to the shaft of wheel.



(a) (b) (c)

**Fig. 9.** Experiment in following a human, (a) turn right, (b) turn left and (c) go back and forth.

In this section, there are three test scenarios so that the effectiveness of our method, the feasible application in practice and the popular situation for the assisted mission are proved. Firstly, robot should follow human to support or supervise as Fig. 9. This task might be necessary in the common places such as super market, warehouse, office or lobby of hotel. The wheeled robot could carry the bulky goods for employer, inventory for worker or the heavy luggage for tourists. In order to complete this mission, robot must recognize human via digital camera located on the top. After identifying the target, it could track as soon as possible. However, it always keeps a safe distance from human's position for emergent case.

(a) (b) (c)

**Fig. 10.** Experiment in navigating between human and object with the proposed approach.

(d)

(e)

(f)

**Fig. 10.** (*continued*)

In the second test, the proposed method for autonomous navigation is deployed. There are commonly several obstacles, i.e. table, chair or fan in the indoor environment since robot works in the office or warehouse. In Fig. 10, the experimental context consists of some plastic chairs, tables, fan and fire extinguisher. Especially, human stays on the way go to destination. It means that autonomous robot must pass those obstacles and human to reach the target. In this case, it recognizes human and some things through vision technique. It is aware of human presence and adjust the system parameters related to model. Robot must respect the personal zone and does not violate the interactive rules. Also, for ensuring the safety and comfort, it would generate the trajectory to cover both

(a)

(b)

**Fig. 11.** Experiment in navigating between human and object without the proposed approach.

(c)

(d)

**Fig. 11.** (*continued*)

human and obstacles. The series of images from Fig. 10a to Fig. 10f to demonstrate the entire operation.

To compare with the traditional method, Fig. 11 illustrates the conventional navigation in the front of human. Usually, robot treats human as a regular obstacle. There is no human-aware information in this case. Therefore, this robot moves to destination by coming across human although the comfort or safety could be interrupted.

## 5 Conclusions

In this paper, a method to autonomously navigate in the presence of human was mentioned. The target control is the robot type of WDD and the target technique is based on vision. The knowledge of human model is embedded into robot so that it could behave socially and safely. The experimental validation on the real hardware has been described in three cases. According to these results, the feasibility, effectiveness and properness of our approach are verified.

**Acknowledgements.** We acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

## References

1. Asadi, E., Li, B., Chen, I.M.: Pictobot: a cooperative painting robot for interior finishing of industrial developments. IEEE Robot. Autom. Mag. **25**(2), 82–94 (2018)
2. Yang, L., et al.: A welding quality detection method for arc welding robot based on 3D reconstruction with SFS algorithm. Int. J. Adv. Manuf. Technol. **94**(1–4), 1209–1220 (2017). https://doi.org/10.1007/s00170-017-0991-9
3. Daniyan, I., Balogun, V., Adeodu, A., Oladapo, B., Peter, J.K., Mpofu, K.: Development and performance evaluation of a robot for lawn mowing. Procedia Manuf. **49**, 42–48 (2020)
4. Bernardi, M., et al.: AURORA, a multi-sensor dataset for robotic ocean exploration. Int. J. Robot. Res. **41**(5), 461–469 (2022)
5. Peng, J., Xu, W., Liang, B., Wu, A.G.: Virtual stereovision pose measurement of noncooperative space targets for a dual-arm space robot. IEEE Trans. Instrum. Meas. **69**(1), 76–88 (2019)
6. Jalali, S.M.J., Ahmadian, S., Khosravi, A., Mirjalili, S., Mahmoudi, M.R., Nahavandi, S.: Neuroevolution-based autonomous robot navigation: a comparative study. Cogn. Syst. Res. **62**, 35–43 (2020)
7. Kalogeiton, V.S., Ioannidis, K., Sirakoulis, G.C., Kosmatopoulos, E.B.: Real-time active SLAM and obstacle avoidance for an autonomous robot based on stereo vision. Cybern. Syst. **50**(3), 239–260 (2019)
8. Wang, J., Luo, C.: Automatic wall defect detection using an autonomous robot: a focus on data collection. In: Computing in Civil Engineering 2019: Data, Sensing, and Analytics, pp. 312–319. American Society of Civil Engineers, Reston, VA (2019)
9. Gharajeh, M.S., Jond, H.B.: Hybrid global positioning system-adaptive neuro-fuzzy inference system based autonomous mobile robot navigation. Robot. Auton. Syst. **134**, 103669 (2020)
10. Gul, F., Rahiman, W., Nazli Alhady, S.S.: A comprehensive study for robot navigation techniques. Cogent Eng. **6**(1), 1632046 (2019)
11. Arrouch, I., Ahmad, N.S., Goh, P., Mohamad-Saleh, J.: Close proximity time-to-collision prediction for autonomous robot navigation: an exponential GPR approach. Alex. Eng. J. **61**(12), 11171–11183 (2022)
12. Malavazi, F.B., Guyonneau, R., Fasquel, J.B., Lagrange, S., Mercier, F.: LiDAR-only based navigation algorithm for an autonomous agricultural robot. Comput. Electron. Agric. **154**, 71–79 (2018)
13. Alatise, M.B., Hancke, G.P.: A review on challenges of autonomous mobile robot and sensor fusion methods. IEEE Access **8**, 39830–39846 (2020)
14. Rusu, R.B., Cousins, S.: 3D is here: point cloud library (PCL). In: 2011 IEEE International Conference on Robotics and Automation, pp. 1–4. IEEE (2011)
15. Thuro, K., et al.: New landslide monitoring techniques–developments and experiences of the alpEWAS project (2010)