ROS based Human assistant robot

Eunji Kwak, Sungin Hwang, Woohyeon Jo, Xuemeng Li, Yujia Hu
University of Birmingham
Intelligent Robotics

***Abstract*—Since its creation, the robot has been intended to serve humans. To maximise the utilisation, robots have been put into many commercial markets in recent years. This report introduces a unique service robot to strengthen the human-robot interaction. In order to create a robot and improve the quality of humans' life, it is necessary to assume plenty of situations between robots and humans and prevent the problems that have a terrible effect on humans' lives. This paper aims to design and develop hotel guest assistant robots that help guests in various ways - for example, carrying heavy luggage and serving food and drinks. SLAM and navigation play the most essential roles in this project. Simulation verifies the validation of this approach by proving the efficiency in SLAM scenarios in restricted environments.**

***Index Terms*—human-robot interaction, navigation, simultaneous localization and mapping (SLAM), hotel guest assistant robots assistant robots**

I. Introduction

RECENT trends in Robotics have led to a proliferation of studies that have pushed the commercialisation of robots to a peak and continue to rise[1]. In robotics, human-robot interaction is an essential factor that has a decisive impact on the quality of life. Service robots are one of the attractive types of robots which could develop a high rate of human life. The project aims to strengthen the relationship between humans and robots using service robots, notably similar to hotel assistant robots; allowing robots to create more value in using robots for humans. With the development of Artificial Intelligence (AI) technology, much research has focused on how robots can replace human labour in the real world. In response to this tendency, this project has designed service robots for hotel guests.

This report includes the use of laser distance sensors (LDS) and camera sensors to guide hotel guests to their rooms and transport carriers. In this project, the LDS sensor is divided in detail; it identifies all sides of the hotel service robot through 360° rotation and helps avoid obstacles on the planned path. At the same time, once the customer enters a particular location, this robot will calculate and memorise the shortest path and guide the hotel guest to the target room in a short period.

With this service robot being commercialised and put into hotels, the robot could replace tasks that hotel staff used to do[2]. Taking over human tasks could prevent hotel guests from waiting due to delays in reception duties and waiters. Also, it contributes significantly to reducing labour costs. Customer service will be skewed towards satisfaction, which is vital for hotels if they invest the rest of their budget in facilities or the environment.

II. Related Works

Recently, various service robots have appeared frequently in many indoor environments [3]. Robots for human interaction are becoming popular and play an essential role in real life. For shopping malls, guidance [4] and recommender robots communicate with customers and find the target restaurant. Furthermore, restaurant robots can perform ordering and delivery services[5][6]. In real-life scenarios, the use of robots in hotels is not yet as widespread as in other scenarios (e.g. factories.) R. Pinillos [7] used information collected from a hotel that has actually existed for a long time in a study. Based on this, they created the robot as a waiter, providing navigation and information services. In a study by y. Zhang [8] describes a check-in robot that can interact with customers via voice. Room allocations can also be planned for different floors.

The application of SLAM algorithms is essential to achieve the navigation function. The robot works in indoor environments, making the operation of SLAM algorithms more stable and accurate [9]. In most indoor environments, the main structures (e.g. walls, windows, doors, etc.) can be represented by parallel or perpendicular groups. It is sufficient to reconstruct the required map to extract and maintain these main lines [10]. Due to the rapid development of sensors, newer sensors are more accurate, cheaper, smaller and even more intelligent. Using SLAM algorithms makes it very feasible to implement the navigation function of a robot at a lower cost. As mentioned by researchers [11], SLAM can be used well in indoor environments. This is precisely what is required for this project, so a lightweight SLAM algorithm implements the navigation function of this robot.

The OpenCV has been crucial in the achievement [8]. When a robot is equipped with a camera sensor, it will be able to access pictures or video information about its environment. Once the camera has obtained the relevant image information, reading and writing this information using the RGB or BGR formats supported by OpenCV will enable the robot to gain the ability to recognise colours [8].

III. Algorithms

A. Simultaneous Localization and Mapping (SLAM)

By definition, simultaneous localisation and mapping is a critical issue that involves generating a map for an unknown environment and predicting the appropriate position of the robot in the map at the same time. In other words, the basic concept of SLAM is that the robot simultaneously constructs the map itself by moving and updating its position. Nevertheless, with the increasing magnitude of uncertain environments, mapping has become a significant problem for

SLAM. Thus, SLAM has been developed by optimising maps by implementing various algorithms - such as particle filters, extended Kalman filters ,and landmarks. [12]

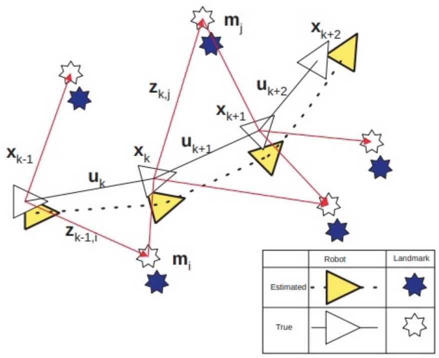

Fig. 1. SLAM problem by using landmarks

Gmapping is one of the ROS packages that OpenSLAM provides. According to Grisetti.G at el.[13], they state that gmapping generates the grid maps using laser data with Rao-Blackwellized particle filters (RBPF).

$$p(x_{1:t}, m \mid z_{1:t}, u_{1:t-1}) \\ = p(m \mid x_{1:t}, z_{1:t}) \cdot p(x_{1:t} \mid z_{1:t}, u_{1:t-1}).$$

Fig. 2. Factorization for Rao-Blackwellized particle filters (RBPF)

Regarding gmapping, this package achieves grid-based SLAM with particle filters. A popular technique is used to represent the environment of a robot in a known position, i.e. the occupancy grid map. In general, the gmapping package is widely used to indicate 2D maps. Hence, it requires several sensors such as Laser Distance Sensors (LDS) and Light Detection and Ranging (LiDAR) to identify obstacles in 2D maps.

Algorithm 1 Gmapping Algorithm

ALGORITHM OCCUPANCY GRID MAPPING$(\mathbf{l_{t-1,i}}, \mathbf{x_t}, \mathbf{z_t})$
  for all cells m_i do
    if m_i perceptual field of v $\mathbf{z_t}$ then
      $\mathbf{l_{t,i}} = \mathbf{l_{t-1,i}}$ +inverse sensor model $(\mathbf{m_i}, \mathbf{x_t}, \mathbf{z_t})$ - $\mathbf{l_0}$
    else
      $\mathbf{l_{t,i}} = \mathbf{l_{t-1,i}}$
    endif
  endfor
  return $(\mathbf{l_{t,i}})$

B. Navigation

In the real world, navigation is one of the more broadly used systems that suggest the best direction for a user-specified destination. In the case of mobile robots, navigation is a notable area of research that concentrates on how to control the motion and movement process of the robot (from the current position of this robot to the target position). Concerning TurtleBot3, it is possible to move a mobile robot to a defined position in a given environment by running a navigation node.

From a perspective of optimal orientation, four elements are required to design the path to the specified pose - map, localisation, sensing, and motion planning. As a purpose of SLAM, it is possible to use SLAM to build a map for a given environment and localisation at the same time before designing the movement.The system is then supposed to start using Adaptive Monte Carlo Localisation (AMCL) parameters and LDS sensor data to estimate the initial position for suggesting the best movement plan. The ROS uses a specific method known as the Dynamic Window Approach (DWA) for path-finding.

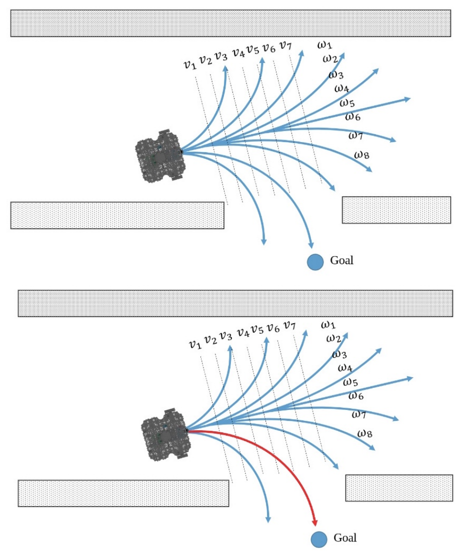

Fig. 3. Dynamic Window Approach (DWA) Algorithm

IV. System Initialization

A. ROS

The Robot Operating System (ROS) is an open source program that provides similar functionality to that offered by operating systems. Meanwhile, several tool programs and libraries are used to acquire, build, write and run multi-machine integration programs. Using ROS [14], it is easier, cheaper and faster to implement navigation, algorithms and robot interfaces and behaviour.

B. Rviz

Rviz, a tool for data visualisation. It can describe the labels written in URDF files and show specific descriptions. Meanwhile, Rviz can also graphically display the information of the robot sensor, the robot's movement status, and the changes in the surrounding environment in real-time. Users can also control the robot behaviours by using buttons, sliders, etc. in Rviz[15].

C. Gazebo

Gazebo is a 3D dynamic simulator that can accurately and effectively simulate robot groups in complex indoor and outdoor environments[16]. It provides a complete set of sensor models, as well as a friendly way of interaction for users and programs.

D. TurtleBot3

Regarding the software platform, TurtleBot is a world-famous open source for education and research. TurtleBot3 is a high quality robot which can be easily assembled, maintained, replaced and morphed in a virtual environment [14]. This open source aspect guides the developer to use the information for research, especially in terms of circuit diagrams, PCB covers and 3D drawings.

Primarily, Turtlebot3 is used in some experiments as a robotics platform in higher education. As we run the simulation instead of a real robot, an IMU (Inertial Measurement Unit) sensor and distance sensor are used to perform the navigation. By using an IMU sensor, it enables us to estimate the factors which affect the robot such as speed, acceleration and angular rate. Moreover, the robot can avoid barricade within the distance sensor by calculating the gap between robot and obstacles.

E. Rqt

ROS provides a monitoring tool called Rqt to check and manage data obtained from robots quickly. Rqt is a Qt framework-based ROS software framework commonly used in GUI development. Rqt contains several GUI plugins, such as Rqt_graph, which this project uses to examine the relationships between nodes on the ROS [18].

F. Open Source Computer Vision Library (OpenCV)

The Open Source Computer Vision Library (OpenCV) library is divided into several modules dedicated to solving a set of computer vision problems. OpenCV supplies functions based on the calibrated image to calculate parameters describing the projection of a 3D point on a 2D image plane. OpenCV offers advanced video analysis of video sequences. It addresses areas such as object or person segmentation, detection and tracking, and camera calibration and 2D shape reconstruction. Moreover, the library includes a complete matrix algebra package to support various algorithms[19].

V. Experiments and Results

A. Experimental Setup

The basic principles of navigation and human tracking are presented, employing these techniques to achieve autonomous route planning and self-directed movement of robots. This process can be classified into two stages: the first stage is the simulation and construction of the environment and the robot , the second is the testing and optimization of the algorithm.

B. Simulation and Construction

Robot Modelling As the critical point of this project is people tracking and navigation, running the robot on Rivz and Gazebo became a compulsory procedure to achieve this goal. The robot is based on a car model for carrying luggage and uses castors to help the wheels move smoothly in different directions. Additional functions are applied to the sensors; the laser sensor allows detection and navigation while the robot is in motion. In combination with this, human following is achieved by the camera sensor. In the case of the different software, Rviz visualises .urdf files; while Gazebo generates .sdf files as robot models. In order to run the robot simultaneously, the URDF file was used to create the robot for this project, which has a camera sensor, a laser sensor and a kinect sensor; the .urdf file was updated with .xacro and run in Gazebo in parallel with Rviz.

The Unified Robot Description Format (URDF) is the typical format for using XML. It is an essential component of robot simulation in ROS and can be converted into a visual robot model using C++'s built-in interpreter. As in [20], in URDF, each robot has link elements and joint elements, each <link> has a 'name' tag, 'visual' for appearance, 'geometry' for shape, 'origin' for offset and tilt arc, and 'material' for colour. There is also 'collision' and 'inertia', all of which Gazebo must have as Gazebo has more detailed functions. Each <joint> makes a safety limit specification and describes more dynamic features.In terms of frame, each robot has a parent frame and one or more child frames [21]. Fig.4 shows the robot structure using the urdf_to_graphiz file.urdf. command.

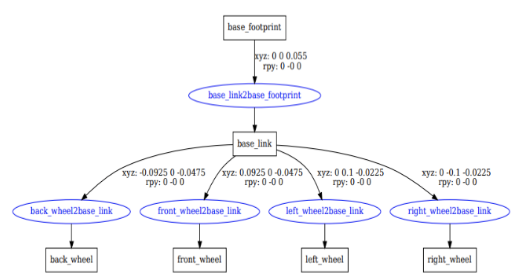

Fig. 4. Structure of the robot in this project

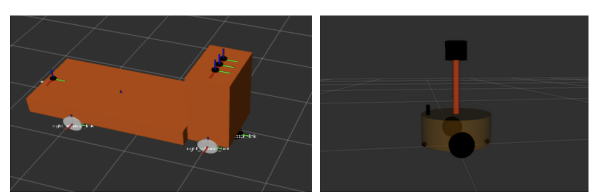

Fig. 5. Prototype of the robot model (left) and Robot model for the project (right)

Since the robot in this project will remain movable, wheels are used for smooth movement, except in the charging area

(Stasis). Fig.5 shows the early prototype of the robot model on the left side and shows the robot used in this project on the right side.

The XML Macros (Xacro) file is essentially equivalent to a URDF file, yet it can be considered an updated version of the URDF file. Whereas the .xacro file allows variable data needed in logic to be exposed as parameters, thus increasing the reuse of code and the security of the program. It is also used to improve writing efficiency.

Map Building To simulate the environment, a map of a hotel running a robot was created. In the hypothesis of this project, the robot is expected to work from the charging point next to the reception desk to the rooms area of the hotel.

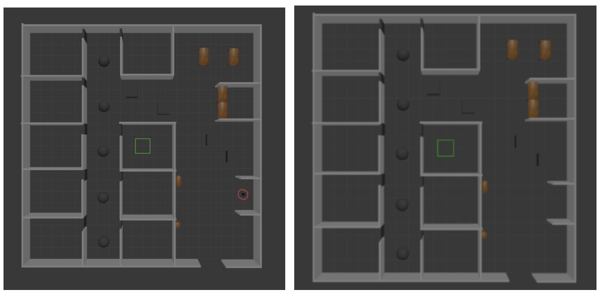

Fig. 6. Run .world file in Gazebo (left) Run .world file within the robot in Gazebo (right)

The multi-storey architectural model map is complicated to materialize for the current situation. Therefore, this project applies this simplified approach to constructing this single storey hotel map with Gazebo as shown in the Fig.6.

C. Testing and Optimization of the Algorithm

Result of Mapping and Localization Generating the map is a first step to find the optimal path to reach the room which is given from the customers. In order to build the map for the restricted environment, the gmapping package is a practical package which is provided from OpenSLAM.

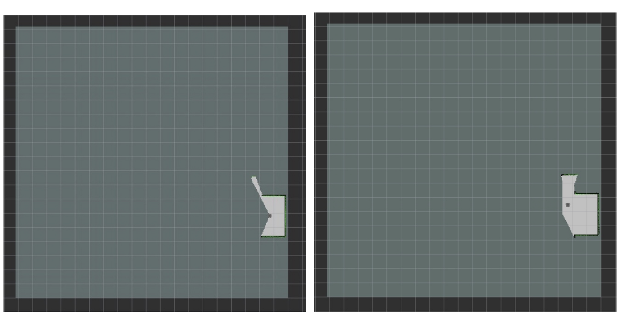

Fig. 7. Mapping in Rviz by implementing the teleop_key package

The difference between Rviz and Gazebo is that Rviz emphasises the visualisation of existing data and is a 3D visualisation tool, while Gazebo focuses on creating a virtual simulation environment and is a 3D physical simulation platform. As in [22], it is important to integrate the display of environmental information perceived by Rviz's robot with sensors and simulate Gazebo's external environment in order to achieve the goal.

By connecting the Rviz and Gazebo using the .launch file, the robot model enables mapping the environment using both environments. It offers not only to build the map in Rviz but also use our own keyboard to send the velocity by implementing the teleop_key package. By implementing the gmapping package and teleop_key package in ROS, our robot can approach mapping for the restricted environment we set successfully.

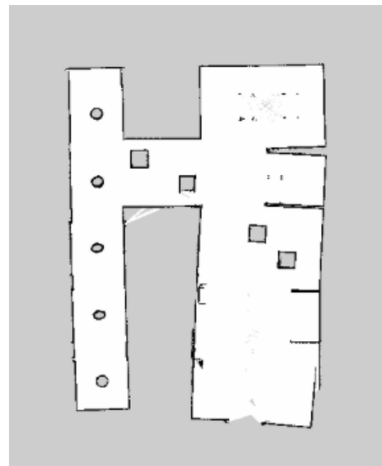

Fig. 8. Result of mapping

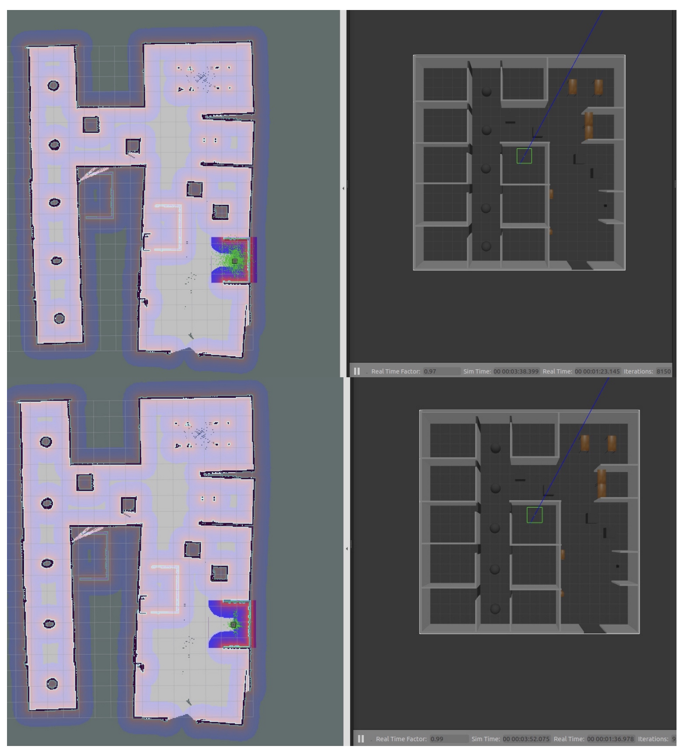

Fig. 9. Localization in both Rviz and Gazebo environments

In terms of SLAM, localization is also the basic step to estimate the pose of the robot in the environment. AMCL system is used for localization which uses the particle filter for probabilistic localization. Fig.9 describes the pose that the robot localized itself in Rviz environment.

Navigation Since the robot gets the map of the environment and localization successfully, it is able to perform navigation to carry the luggage from the charging point to the room which is given from the customer.

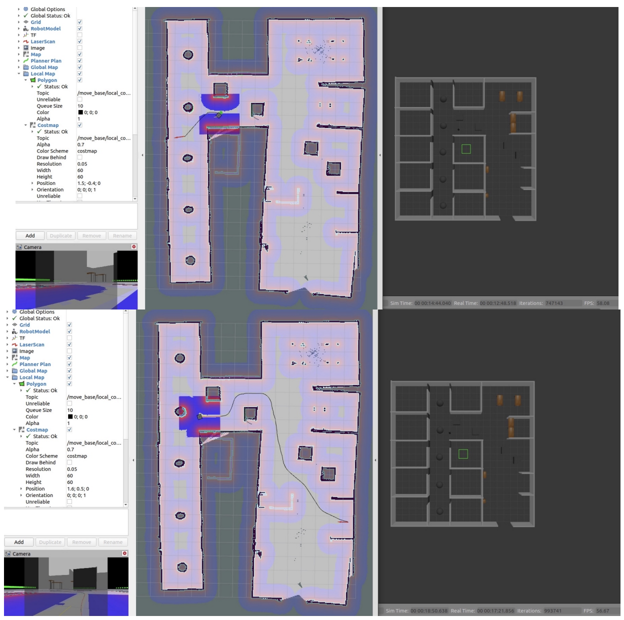

Fig. 10. Navigation in both Rviz and Gazebo environment

Regarding the ROS navigation stack, the package implements the AMCL system, thus it enables us to estimate the pose of the robot in Rviz and Gazebo respectively. Fig. 10 describes the optimal path that the robot finds in both Rviz and Gazebo environments. The left side of Fig. 10 shows the situation when the robot moves from the starting point to room and the right side shows that the robot goes back to the charging point after the users put down their luggages.

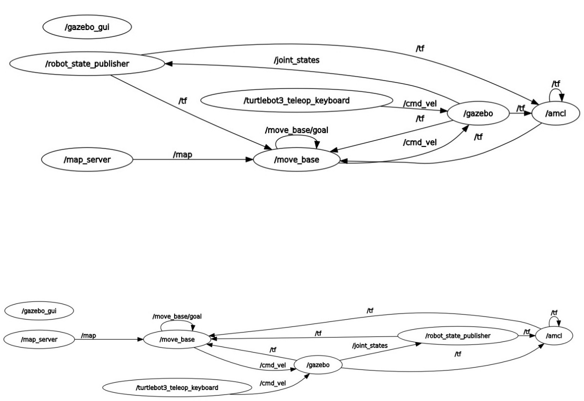

Fig. 11. Rqt_graph from Fig.10

One of the visualization tools which shows all the running nodes and processes, rqt_graph, describes the overview of our simulation. Following figure illustrate the rqt_graph of the Fig.10.

Color Recongition

Fig. 12. Result of color recognition using OpenCV (top left) Original Photo. (top right) Contours (bottom left)Binary image (bottom right) Extracting blue from the original photo .

At the beginning of the project, an experiment was made on colour recognition for human following. The experimental environment consists of a robot with a camera and objects for recognition (implemented in a virtual space, e.g. Gazebo and Rviz). As the camera is positioned in front of the robot, it is able to capture the object to be recognised while it is driving. cv2.VideoCapture() is used to receive frames of a moving human by capturing the video.

The findContours is a function that utilises color detection, it is applied to connect the boundaries of the same colour intensity. Meanwhile, the cv2.RETR_EXTERNAL was utilized to return the counter found and the image hierarchy, and the contour was drawn as shown in (b) in Fig. 12 using drawContours. Alternatively, in order to remove unnecessary contours, a new arrangement of contours is drawn by using ArcLength to append only values of length above 100 to the list, but there is no significant difference.

These meaningless results led to the colour recognition of garments used to implement human tracking. To facilitate the classification of colours, the BGR image is converted to an HSV image according to the RGB to HSV conversion formula. Having identified a desired colour range in the HSV image, cv2.inRange generated a binary image as shown in (c) in Fig. 12 by rendering the pixels within that range white and the rest black. In determining the colour range, extremely dark colours and extremely light colours are not used to attain high recognition accuracy. For this reason, as shown in (d) in Fig. 12, only the part corresponding to the colour range can be obtained from the original image.

Fig. 13 illustrates the results of running the process mentioned above without OpenCV. Because OpenCV was not implemented for this graph, it was difficult to determine the colour areas. Therefore, the results obtained are shown in Fig. 13.

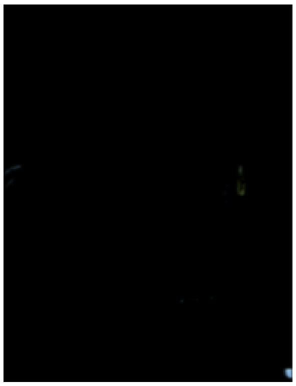

Fig. 13. Result of color recognition without OpenCV

Unfortunately, these experiments explained above attempted to implement nodes using the CvBridge function converts between ROS image information and OpenCV images to be tested using Rviz and Gazebo, but failed. In addition, adding human models to the simulation environment (virtual spaces such as Rviz and Gazebo) is unlikely to be achievable in the short term for this project. Even if colour detection for human tracking is accomplished, it is not possible to present complete human tracking on Rviz and Gazebo due to the lack of knowledge on the use of Rviz and Gazebo.

VI. Conclusion and Future works

In summary, this report describes the robot for this project as a hotel service robot. A "bellboy" that can do navigation, carries various items to do automatic path planning and implementation of it. For these functions, it has been realized that path-finding can be corrected and obstacles can be avoided by the robot. When items are placed on the robot, the customer experiences freedom and relaxation on the way to the room. More significantly for the hotel operator, customer satisfaction can enhance the reputation of the hotel; above all, by reducing human labour costs, the robot can generate further benefits for the hotel. Besides, the robot can be a selling point to attract customers. Since it is not yet as widely used as other human-interaction robots, hopefully in the near future hotel service robots will be used extensively.

The ultimate aim for robot development is to help people in real life. There are also purposes to run the robot for this project in the real world, better results can be obtained by using filters such as cv2.GuassianBlur and cv2.medianBlur during colour recognition. As a filter for noise reduction, the median filter is employed as a non-linear filter for contour detection. By maintaining the edges of the shape relatively, it reduces real-world noise, e.g. light and shadow. Simultaneously, subtle reinforcement learning is used to connect with real robots and to facilitate human following, so that the robot can continue to follow humans even at a large distance from them.

The plan in future work is to update it using the Kinect sensor. This will improve the system and allow for better control of the speed. This would mean that it will move at a faster speed rather than the current constant speed and increasing efficiency. The robot's recognition and detection of colours will also be kept in focus. Once more static and dynamic objects are detected, avoidance rates will rise; robots will be able to deliver items more safely and reliably. There is also the addition of a voice-activated sensor to communicate with customers being considered for robots when it is actually in use. This would greatly benefit the user experience.

Git Repository https://github.com/ezi-sab/IR_FinalProj

References

[1] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "*The evolution of robotics research*," IEEE Robot. Autom. Mag., vol. 14, no. 1, pp. 90–103, 2007.

[2] Y. Choi, M. Choi, M. (moon) Oh, and S. (sam) Kim, "*Service robots in hotels: understanding the service quality perceptions of human-robot interaction*," J. hosp. mark. manag., vol. 29, no. 6, pp. 613–635, 2020.

[3] N. Bellotto and H. Hu, "*Multisensor-based human detection and tracking for mobile service robots*," IEEE Trans. Syst. Man Cybern. B Cybern., vol. 39, no. 1, pp. 167–181, 2009.

[4] T. Kanda, M. Shiomi, Z. Miyashita, H. Ishiguro, and N. Hagita, "*An affective guide robot in a shopping mall*," in Proceedings of the 4th ACM/IEEE international conference on Human robot interaction - HRI '09, 2009.

[5] S. H. Ivanov, C. Webster, and K. Berezina, "*Adoption of robots and service automation by tourism and hospitality companies*," 2017.

[6] J. Hwang, S. Park, and I. Kim, "*Understanding motivated consumer innovativeness in the context of a robotic restaurant: The moderating role of product knowledge*," J. Hosp. Tour. Manag., vol. 44, pp. 272–282, 2020.

[7] V. W. S. Tung and R. Law, "*The potential for tourism and hospitality experience research in human-robot interactions*," Int. J. Contemp. Hosp. Manag., vol. 29, no. 10, pp. 2498–2513, 2017.

[8] P. Tavitiyaman, X. Zhang, and W. Y. Tsang, "*How tourists perceive the usefulness of technology adoption in hotels: Interaction effect of past experience and education level*," J. China tour. res., pp. 1–24, 2020.

[9] J. Aulinas, Y. Petillot, J. Salvi, and X. Lladó, "*The SLAM problem: a survey*," in Artificial Intelligence Research and Development, Amsterdam, NY: IOS Press, 2008, pp. 363–371.

[10] V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, "*Orthogonal SLAM: A step toward lightweight indoor autonomous navigation*," in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006.

[11] T. Tomic et al., "*Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue*," IEEE Robot. Autom. Mag., vol. 19, no. 3, pp. 46–56, 2012.

[12] J. E. Guivant and E. M. Nebot, "*Optimization of the simultaneous localization and map-building algorithm for real-time implementation*," IEEE Trans. Rob. Autom., vol. 17, no. 3, pp. 242–257, 2001.

[13] G. Grisetti, C. Stachniss, and W. Burgard, "*Improved techniques for grid mapping with Rao-blackwellized particle filters*," IEEE Trans. Robot., vol. 23, no. 1, pp. 34–46, 2007.

[14] G. Gargioni, M. Peterson, J. B. Persons, K. Schroeder, and J. Black, "*A full distributed multipurpose autonomous flight system using 3D position tracking and ROS*," in 2019 International Conference on Unmanned Aircraft Systems (ICUAS), 2019, pp. 1458–1466.

[15] S. Hart, P. Dinh, and K. Hambuchen, "*The Affordance Template ROS package for robot task programming*," in 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 6227–6234.

[16] M. Zhang et al., "*A high fidelity simulator for a quadrotor UAV using ROS and Gazebo*," in IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society, 2015, pp. 002846–002851.

[17] R. Amsters and P. Slaets, "*Turtlebot 3 as a robotics education platform*," in Robotics in Education, Cham: Springer International Publishing, 2020, pp. 170–181.

[18] R. Guzman, R. Navarro, M. Beneto, and D. Carbonell, "*Robotnik—professional service robotics applications with ROS*," in Studies in Computational Intelligence, Cham: Springer International Publishing, 2016, pp. 253–288.

[19] P. N. Druzhkov et al., "*New object detection features in the OpenCV library*," Pattern Recognit. Image Anal., vol. 21, no. 3, pp. 384–386, 2011.

[20] M. Mortimer, B. Horan, M. Joordens, and A. Stojcevski, "*Searching Baxter's URDF robot joint and link tree for active serial chains*," in 2015 10th System of Systems Engineering Conference (SoSE), 2015, pp. 428–433.

[21] C. Crick, G. Jay, S. Osentoski, and O. C. Jenkins, "*ROS and Rosbridge: Roboticists out of the loop*," in Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction - HRI '12, 2012, pp. 493–494.

[22] P. Vyavahare, S. Jayaprakash, and K. Bharatia, "*Construction of URDF model based on open source robot dog using Gazebo and ROS*," in 2019 Advances in Science and Engineering Technology International Conferences (ASET), 2019, pp. 1–5.