

## A Novel Automatic Meal Delivery System

Jhe-Wei Lin<sup>1</sup>, Cheng-Yan Siao<sup>1</sup>, Ting-Hsuan Chien<sup>2,\*</sup> and Rong-Guey Chang<sup>1</sup>

<sup>1</sup>Department of Computer Science and Information Engineering, Advanced Institute of Manufacturing with High Tech Innovations, National Chung Cheng University, Chiayi, 621005, Taiwan

<sup>2</sup>Department of Computer Science and Information Management, Providence University, Taichung, 433303, Taiwan

\*Corresponding Author: Ting-Hsuan Chien. Email: thchien0616@pu.edu.tw

Received: 02 March 2021; Accepted: 05 April 2021

**Abstract:** Since the rapid growth of the Fourth Industrial Revolution (or Industry 4.0), robots have been widely used in many applications. In the catering industry, robots are used to replace people to do routine jobs. Because meal is an important part of the catering industry, we aim to design and develop a robot to deliver meals for saving cost and improving a restaurant's performance in this paper. However, for the existing meal delivery system, the guests must make their meals by themselves. To let the food delivery system become more user-friendly, we integrate an automatic guided vehicle (AGV) and a robotic arm to deliver a meal and put it on a table efficiently. The system consists of four phases. In the first stage, since most food feeding environments are indoors, we must build a map of an environment for a robot and use a laser sensor to construct a surrounding object's real scene. The second stage is global planning. The map is used for indoor navigation by computing the shortest path from the starting point to the map's end point. The third phase is regional planning for obstacle avoidance, immediate avoidance, and re-planning of obstacles on the path. The last stage is how to control a robot arm to grab and put a meal on a table. We use an icon as the hint of the height of a table and operate the arm rotation angle according to this icon's position. Our system provides the concept of an AGV combined with a robotic arm and provides a service for the food delivery system. In this way, waiters do not spend time dealing with meals and guests can automatically obtain their meals. The result shows that the proposed system will be helpful for the robot industry and the catering industry.

**Keywords:** Industry 4.0; robot applications; meal delivery system; automatic guided vehicle

### 1 Introduction

More and more robot applications exist in our life, and more robots have been developed to help them work faster and more automated. We can see that robots have been invested in the logistics, medical, and catering industries to provide unmanned services. Industrial robots sold 420,000 robot combinations in 2019 and are expected to have 580,000 by 2020 and in 2020 [1]. There will be 12% growth each year



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

until 2022. In other categories of robots, such as service robots for home or entertainment, we can find that the number and the growth rate are far greater than industrial robots. Especially due to COVID 19, the world has been caught in an unprecedented pandemic and the unmanned services of robots have been paid more attention to. People seek for new technologies to help people maintain an everyday life while reducing human-to-human contact with the virus infection. Robots have become an important choice.

Recently, with the importance of Industry 4.0, robots have become a popular tool in a variety of industries, and how to apply them in life has been a critical issue of research and development. It is not only in the industrial tool machine research, but also on many applications in our life such as medical, catering, etc. Especially in the catering industry, the robots and AGVs can be integrated together to replace human labor and shorten the high repetitive workload. For example, using robots and AGVs in a restaurant can deliver food to customers automatically without waiters. It is also desirable to improve efficiency and increase enterprise revenue, which is also the purpose of the birth of robots.

To let the food delivery system become more user-friendly, we focus on proposing a complete automation system with robots and AGVs without human intervention and the aid of specific routes in this paper. It can deliver something and move freely like people. We integrate an automatic guided vehicle (AGV) and a robotic arm to provide a meal and put it on a table efficiently. The system consists of four phases. In the first stage, since most food feeding environments are indoors, we must build a map of an environment for a robot and use a laser sensor to construct a surrounding object's real scene. The second stage is global planning. The map is used for indoor navigation by computing the shortest path from the starting point to the end map's end point. The third phase is regional planning for obstacle avoidance, immediate avoidance, and re-planning of obstacles on the path. The last stage is how to control a robot arm to grab and put a meal on a table. We use an icon as the hint of the height of a table and operate the arm rotation angle according to this icon's position. Our system provides the concept of an AGV combined with a robotic arm and provides a service for the food delivery system.

In this paper, the proposed system is composed of mobile robots, robotic arms, and LIDAR. The contribution of this paper is highlighted as follows. (1) To let a robot put a meal on a table, we must send the height of the table to a robot so that the meal can be placed well on the table. To achieve this objective, we apply the computer vision techniques to control the robotic arm through the icons at the table. In this way, it can solve many problems arising from the real-world environment. (2) Our work can apply to different robots to deliver meals, such as common mobile robots and AGVs. We must consider the real environment to avoid unexpected occurrences and try to meet many application's requirements. (3) We present an effective delivery system that can be applied to the robot industry and the catering industry. Our system can reduce the labor cost and share the workload of staff.

The remainder of the paper is organized as follows. We describe the previous work related to our work in Section 2. Then we present how to design and implement the meal delivery system in Section 3. In Section 4, we show the simulation and results for the meal delivery system. Finally, we conclude this paper in brief in Section 5.

## **2 Related Work**

This section discusses food delivery robot, indoor map construction for a robot, and the positioning and indoor navigation for a robot as follows.

### ***2.1 Food Delivery Robots***

For some existing robot feeding systems, a waiter puts a meal on the robot like a dining car, and then the robot can send the feed to the location according to the specific route [1]. Then customers can take the feed from the robot by themselves. Checking this whole process, we can find that it is not an automatic system because it needs the help of a waiter and a customer.

The work presented by Li et al. [2] mentioned that for the delivery robot, the first generation focuses on following the tape track on the ground to move. It is easy to encounter poor anti-jamming ability, the influence of light and surface stains. Due to the preset path, it also results in poor user experience and the increasing cost of the pre-set installation trajectory work to control movement. Thus, some people put forward laser sensors to solve the trajectory's difficulties and inconveniences. Cochrane et al. [3] proposed to use lidar to increase the elasticity of robot movement. In general, lidar is also used for robot positioning, laser positioning system for fast and mobile robot position and guidance. In this way, it can improve the overall delivery stability and reliability. In their design, they placed the lidar on top of the robot and adds a supersonic sensor in the base to avoid collisions.

## **2.2 Map Construction and Positioning**

Generating a map of an unknown environment and locating the robot is the main task of the robot. Much of the research has focused on how to represent the environment and the location of robots. However, this problem is very complicated, because robots construct maps, but robots need maps to locate themselves. To overcome the problems with mapping an unknown environment and locating locations, SLAM (Simultaneous localization and mapping) algorithm is one of the important solutions proposed to solve this problem based on sensors such as laser-scanning radar [4,5]. Balasuriya et al. submitted a workshop to design and navigated a robot autonomously based on Robot Operating System (ROS) [6]. Their work had divided the concept of SLAM [7–9] into localization and mapping. The Gmapping algorithm is explicitly used to solve the SLAM problem based on particle filtering [10–12]. Yassin et al. presented a comparison for different SLAM, including HectorSLAM, Gmapping, KartoSLAM [13,14].

## **2.3 Indoor Navigation**

Indoor path planning is also a critical issue of it in the field of robotics. The difficult point is how to find the best path as soon as possible. Abdelrasoul et al. [15] applied A\* algorithm to solve the problem of indoor navigation. It was used earlier to move characters in the game. In contrast to the well-known algorithm proposed by Loong et al. [16], although Dijkstra's algorithm can guarantee to find the shortest path, however, it is not as fast and straightforward as the A\* algorithm.

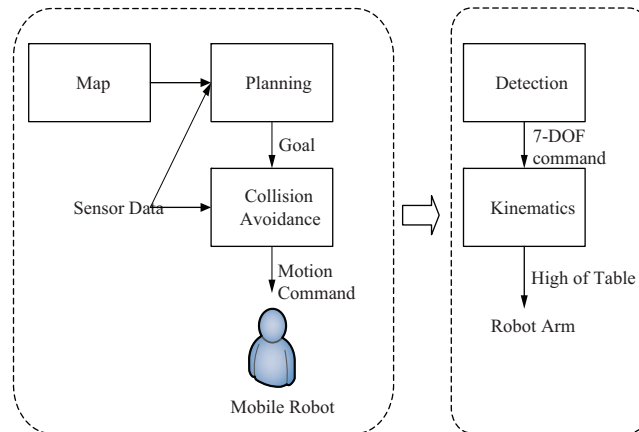
## **3 Method**

In this section, we will introduce the meal deliver system in detail.

### **3.1 Overview**

We first use laser scanning sensors as the primary basis to construct a map. Then the map becomes the input to indoor navigation. Next, we perform global planning and regional planning. Global planning can guide a robot to move from point A to point B according to the map's walkable area. As to regional planning, a robot must avoid obstacles and replan the path to point B based on Dynamic Window Approach (DWA) [17]. In our system, the robot must place the meal on the table in a restaurant while it reaches the destination. To achieve this goal, we have to detect the height of a table and the location to be placed via the image. Fig. 1 shows the overall architecture of the proposed system.

To design a social delivery system with high mobility, we must consider the cost factors so that it is easy to promote in the future. We must abandon the traditional route such as glue tape on the ground to let a robot move free. Thus, we use laser scanning sensors to increase behavior elasticity and apply Gmapping to construct the map efficiently. Finally, A\* algorithm is applied to our system to guide a robot. In the completed map, it is very efficient.



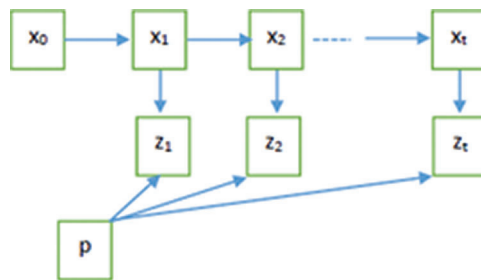
**Figure 1:** The overall architecture

### 3.2 Map Construction

We apply the Gmapping algorithm to construct a map for a robot presented as a grid diagram. We use the Rao-Blackwellized Particle Filter (RBPF) to create a map with laser scanning. The principle of RBPF is the hypothetical state, where each particle remains in the state of the measurement obtained by the laser scanner. Each landmark is associated with the corresponding particle reserved for a given weight. The arguments  $x_{1:t}$ ,  $u_{1:t-1}$  and  $z_{1:t-1}$  in Eq. (1) are represented for the robot's position and the mobile distance, and the sensor information, respectively.

$$p(x_{1:t}, m|z_{1:t}, u_{1:t-1}) = p(m|z_{1:t}, u_{1:t-1}) p(x_{1:t}|z_{1:t}, u_{1:t-1}) \quad (1)$$

In the following example in Fig. 2,  $x_t$  is computed by  $x_{t-1}$ . Each location information is obtained from an external environment  $z_t$  to continually updating the map  $P$ . The algorithm can be sorted according to  $x_t, z_t$ .



**Figure 2:** An example of a map construction

### 3.3 Path Planning

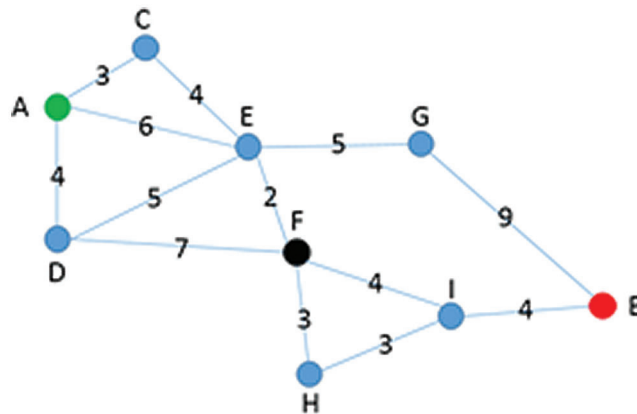
As mentioned before, path planning is divided into global planning and regional planning in indoor navigation. Global planning aims at planning paths for the map, while regional planning focuses on avoiding obstacles that are not on the map and planning new paths.

#### 3.3.1 Global Planning

We apply A\* algorithm to perform the global planning to find the shortest path between two points,  $A$  and  $B$ , efficiently in a map. The cost function presented in Eq. (2) is used in A\* algorithm to evaluate which choice is best, where  $f(n)$  is the current estimated distance of point  $n$ ,  $g(n)$  is the distance from starting point  $A$  to the current point  $n$ , and  $h(n)$  is the estimated distance from the current point  $n$  to the destination  $B$ .

$$f(n) = g(n) + h(n) \quad (2)$$

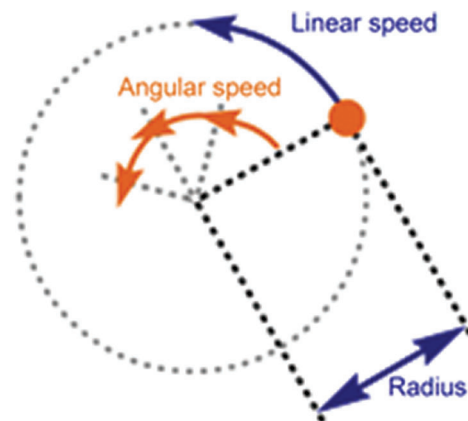
It is easy to find that  $h(n)$  is the key to the cost function. In general,  $h(n)$  is the distance of a straight line from each point to the endpoint. If there are obstacles in the map, such as walls in a real environment, the path will be reselected. For example, assume if node F in Fig. 3 is a wall, the shortest path will become from E to G. As a result, if obstacles have been generated on the map, they are avoided in global planning.



**Figure 3:** Path planning to avoid obstacles

### 3.3.2 Regional Planning

Regional path planning is designed to avoid obstacles on the map and new paths must be planned. We apply Dynamic Window Approach (DWA) [18] to achieve this goal. In Fig. 4, we can use Eqs. (3–5) to compute linear speed, angular speed, and radius in a two-dimensional space. In the dynamic window, all curves outside this dynamic window cannot be reached by the robot in the next step. Thus no avoidance must be considered. Each trajectory is then compared with the information from the lidar. The trajectory is considered safe if the robot can stop before colliding with any object along the path.



**Figure 4:** An example of linear speed and angular speed

$$\text{Linear Speed: } v = \frac{s}{t} \quad (3)$$

$$\text{Angular Speed: } w = \frac{\theta}{t} \quad (4)$$

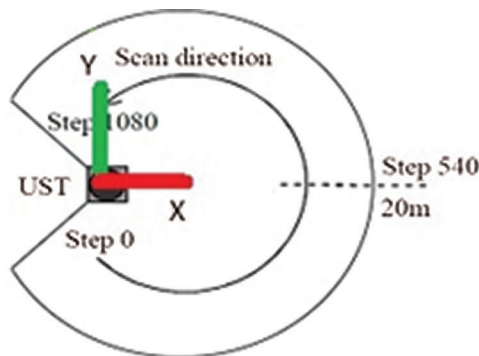
$$\text{Radius: } r = \frac{\text{Linear Speed}}{\text{Angular Speed}} = \frac{v}{w} \quad (5)$$

### 3.3.3 Detecting the Height of a Table

Table information is used to assist the robot arm to place a meal more precisely. To reduce the noise while detecting the table, we use the icon attached on the edge of the table to know the location of the table. In this paper we use the SIFT (scale-invariant feature transform) algorithm for feature matching because it has the advantages of rotation, size, pan, angle of view, and the consistence of brightness. However, since the input is continuous frame information, many pictures will match at the same time. To find the real center point, we add the time information for analysis. Since the matching function of SIFT is exact,  $w$  can compare the center points to ensure that it is the real center point if they are the same point. Thus, we can obtain the position of a table based on the following Eq. (6).

$$\text{center point } (x, y) = (x_t, y_t), \text{ where } x_t - x_{t-1} = 0 \text{ and } y_t - y_{t-1} = 0 \quad (6)$$

After obtaining the center point of the icon, we can precisely know the position and height of the table. However, the depth information is still missing. The computer vision processing, we cannot know the distance between the robot and the table. We use the lidar to obtain the distance in front of the robot, as shown in Fig. 6. Although the depth information obtained is different a little for other LIDAR's, however, they use sweep able range, angle resolution, and the number of particles after a run to obtaining it. In Fig. 5, the lidar supports a 270-degree scanning range, with one particle per 0.25 degrees, for a total of 1080 particles.



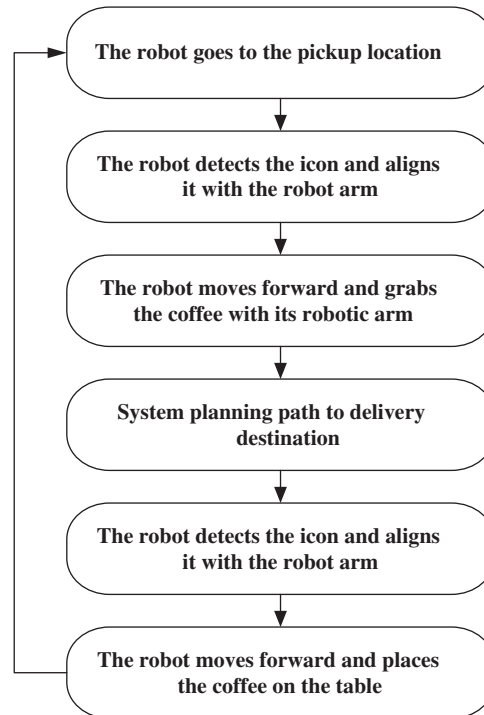
**Figure 5:** The scanning range of a lidar

Thus, we can obtain the distance of lidar from the object in front by using the following Eqs. (7–9) After obtaining the depth information. The meddler can approach the table and pass instructions to the robot arm to place a meal on the table.

$$\text{nodeQuantity} = \text{angle of range/angle} \quad (7)$$

$$\text{Particles} = \{\text{range}[i], i = 1 \sim \text{nodeQuantity}\} \quad (8)$$

$$\text{Distance of front} = \text{range}[\text{nodeQuantity}/2] \quad (9)$$



**Figure 6:** The simulation flow

## 4 Experimental Result

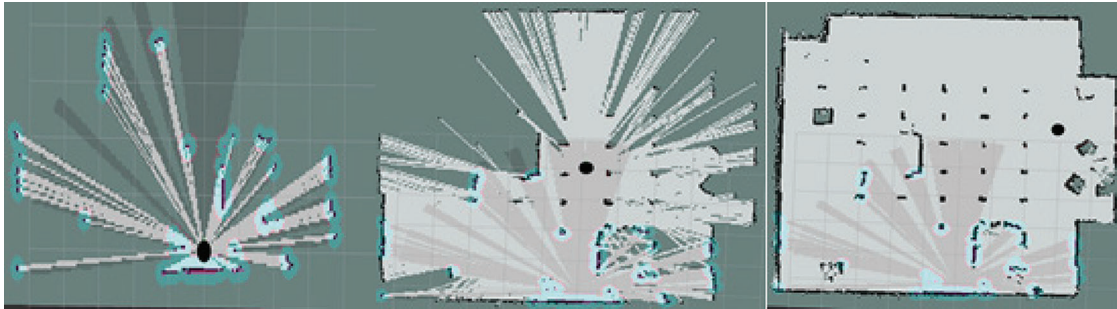
In this experiment, we use the EDI dash go D1 [19] as AGV with the load of 50 kg and a battery life of up to 6 hours. It uses the LiDAR F4 that scans 360 degrees of scattered infrared particles. Besides, we buy the components to assemble a six-axis robotic arm with an MG996R motor on each axis. We use Gorman [20] as a control board to communicate between the arm and the mobile platform.

### 4.1 The Simulation Flow

To simulate the delivery system, we design some situations of food delivery to verify. Fig. 6 is the simulation flow. We first let the robot understand the surrounding environment to establish a map. Then the robot moves the specific place to take the meal according to the map. At this moment, the robot is waiting to receive the table information to send the meal, gets the icon information to grab the coffee cup, then plan the delivery path, and finally put the coffee on the destination table. After finishing the delivery, the robot will return to the original location for the next job.

### 4.2 Map Construction

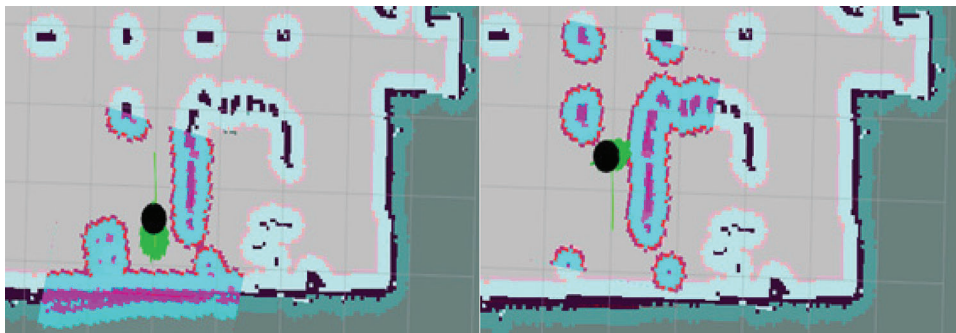
We first construct the initial, the intermediate, and the final map of our testing restaurant, as shown in Fig. 7.



**Figure 7:** The initial construction, the intermediate, and the final map

#### 4.3 Taking the Meal

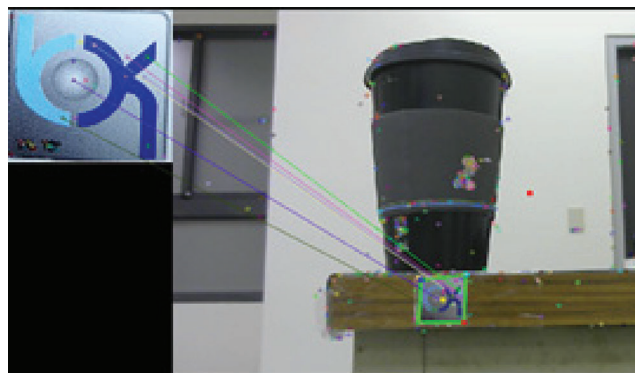
With the map's aid, the robot can send the meal to the destination table by following the green line, as shown in Fig. 8. Note that the pink and blue parts are the regions detected by lidar.



**Figure 8:** The robot starts in the left location and arrives at the right location to take the meal

#### 4.4 Icon Detection

After the two stages above, Fig. 9 shows that we can recognize the icon. As mentioned earlier, we use the SIFT algorithm to detect the icon attached to the table, then find the center point of the icon after moving the robot and aligning the icon. The large icon on the left corresponds to the object in the picture on the right. Two pictures of different color circles are the special points to find the corresponding object according to both sides feature points.

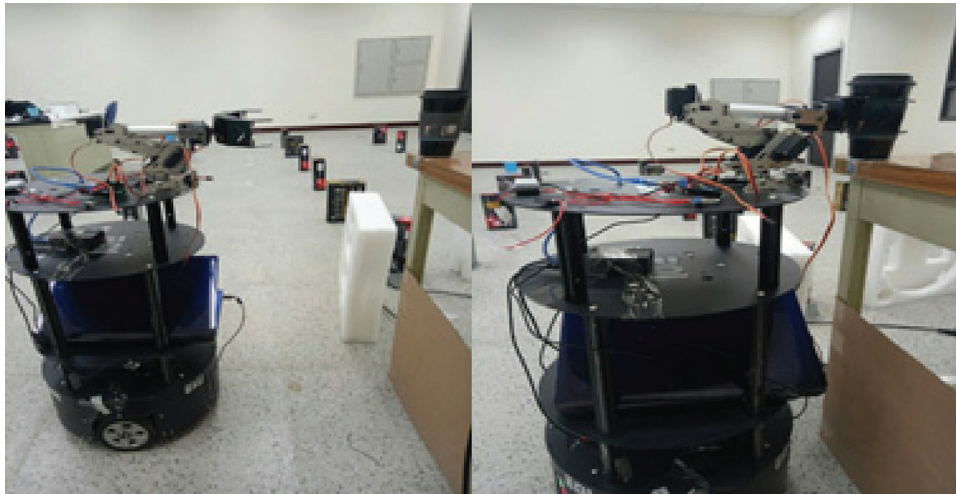


**Figure 9:** An example to recognize an icon



#### 4.5 Grabbing a Cup

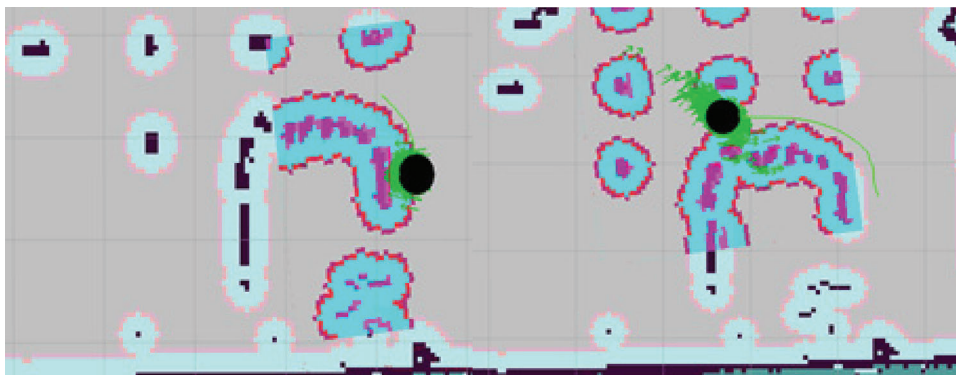
Fig. 10 shows the process of grabbing a coffee cup with the help of icon recognition.



**Figure 10:** The robot can grab a cup

#### 4.6 Path Planning

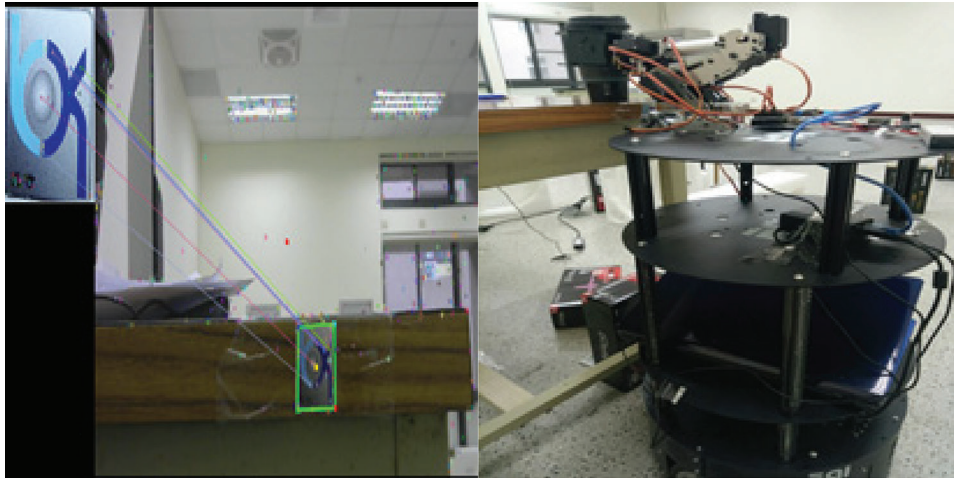
After accepting the delivery destination's coordinates, the robot plans the shortest path marked by the green line to the destination. The following Fig. 11 show this process with the navigation.



**Figure 11:** An example that a robot sends a meal

#### 4.7 Putting the Cup on the Table

At this stage, we apply the SIFT algorithm to detect the icon again to put the cup on the destination table, as shown in Fig. 12.



**Figure 12:** The process to put a cup

## 5 Conclusions

We have presented a delivery system for a robot. From the experiment, we prove that the proposed work can work well to solve this problem. With our work's help, our system can really reduce the labor cost and can be applied to the real environments.

**Acknowledgement:** The authors would like to acknowledge the financial support of the National Science Council, Taiwan, R. O. C. under the grant of project MOST 108-2218-E-194-007.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] M. Haegele, *Executive Summary World Robotics 2019 Service Robots*. Frankfurt, Hessen, Germany: International Federation of Robotics, 2019. [Online]. Available at: <https://ifr.org/downloads/press2018/Executive%20Summary%20WR%202019%20Industrial%20Robots.pdf>.
- [2] C. Li, X. Liu, Z. Zhang, K. Bao and Z. T. Qi, "Wheeled delivery robot control system," in *Proc. MESA*, Auckland, AKL, New Zealand, pp. 1–5, 2016.
- [3] W. A. Cochrane, X. Luo, T. Lim, W. D. Taylor and H. Schnetler, "Design of a micro-autonomous robot for use in astronomical instruments," *International Journal of Optomechatronics*, vol. 6, no. 3, pp. 199–212, 2012.
- [4] J. H. Tzou and L. S. Kuo, "The development of the restaurant service mobile robot with a laser positioning system," in *Proc. CCC*, Kunming, KMG, China, pp. 662–666, 2008.
- [5] R. J. Yan, J. Wu, S. J. Lim, J. Y. Lee and C. S. Han, "Natural corners-based SLAM in unknown indoor environment," in *Proc. URAI*, Daejeon, QTW, South Korea, pp. 259–261, 2012.
- [6] J. Folkesson and H. Christensen, "Graphical SLAM—a self-correcting map," in *Proc. ICRA*, New Orleans, LA, USA, pp. 383–390, 2004.
- [7] B. Vincke, A. Elouardi and A. Lambert, "Design and evaluation of an embedded system based SLAM applications," in *Proc. SSI*, Sendai, SDJ, Japan, pp. 224–229, 2010.
- [8] B. Vincke, A. Elouardi, A. Lambert and A. Merigot, "Efficient implementation of EKF-SLAM on a multi-core embedded system," in *Proc. IECON*, Montreal, QC, Canada, pp. 3049–3054, 2012.

- [9] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis and R. Siegwart, "A comparison of line extraction algorithms using 2D range data for indoor mobile robotics," *Autonomous Robots*, vol. 23, no. 2, pp. 97–111, 2007.
- [10] J. E. Guivant and E. M. Nebot, "Optimization of the simultaneous localization and map-building algorithm for real-time implementation," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 242–257, 2001.
- [11] P. Yang, "Efficient particle filter algorithm for ultrasonic sensor-based 2D range-only simultaneous localisation and mapping application," *IET Wireless Sensor Systems*, vol. 2, no. 4, pp. 394–401, 2012.
- [12] Z. Zhang, H. Guo, G. Nejat and P. Huang, "Finding disaster victims: A sensory system for robot-assisted 3D mapping of urban search and rescue environments," in *Proc. ICRA*, Roma, RM, Italy, pp. 3889–3894, 2007.
- [13] B. L. E. A. Balasuriya, B. A. H. Chathuranga, B. H. M. D. Jayasundara, N. R. A. C. Napagoda, S. P. Kumarawadu *et al.*, "Outdoor robot navigation using gmapping based SLAM algorithm," in *Proc. MERCon*, Moratuwa, COL, Sri Lanka, pp. 403–408, 2016.
- [14] Q. Lin, Z. Ke, S. Bi, S. Xu, Y. Liang *et al.*, "Indoor mapping using gmapping on embedded system," in *Proc. ROBIO*, Macau, MO, China, pp. 2444–2449, 2017.
- [15] Y. Abdelrasoul, A. B. S. H. Saman and P. Sebastian, "A quantitative study of tuning ROS gmapping parameters and their effect on performing indoor 2D SLAM," in *Proc. ROMA*. Ipoh, MBI, Malaysia, pp. 1–6, 2016.
- [16] W. Y. Loong, L. Z. Long and L. C. Hun, "A star path following mobile robot," in *Proc. ICOM*, Kuala Lumpur, KL, Malaysia, pp. 1–7, 2011.
- [17] Y. Deng, Y. Chen, Y. Zhang and S. Mahadevan, "Fuzzy Dijkstra algorithm for shortest path problem under uncertain environment," *Applied Soft Computing*, vol. 12, no. 3, pp. 1231–1237, 2012.
- [18] C. Kim, R. Sakthivel and W. K. Chung, "Unscented FastSLAM: A robust and efficient solution to the SLAM problem," *IEEE Transactions on Robotics*, vol. 24, no. 4, pp. 808–820, 2008.
- [19] B. Fang, J. Ding and Z. Wang, "Autonomous robotic exploration based on frontier point optimization and multistep path planning," *IEEE Access*, vol. 7, no. 1, pp. 46104–46113, 2019.
- [20] M. Gorman, *Understanding the Linux Virtual Memory Manager*. Upper Saddle River, NJ, USA: Prentice Hall, 2004. [Online]. Available at: <https://www.cs.miami.edu/home/burt/learning/Csc521.081/notes/understand.pdf>.