Arm Action Planning and Pick-and-Place Objects for a Humanoid Robot

Chih-Cheng Liu, Yi-Chung Lin and Jen-Jia Tien

***Abstract*—In the future, when the labor force is reduced and costs are rising, a humanoid robot can be used to perform work that was originally for a human. In this paper, the function of object gripping is implemented on a small humanoid robot platform Image recognition is used to determine the position of objects on the table and the objects are clamped to the target area in accordance with the planned sequence. First, the robot stands in front of the table and uses the camera to identify the position of the object on the table. Then, it controls the arm to grasp the object and finally places the object in the collection area. Table tennis balls and paper cups are used to represent two different types of garbage. The robot judges the objects and order of picking, and places the objects in a trash can of the respective color next to the table. The computing performance of the robot is not very good, so this paper combines the greedy heuristic method and the traveling salesman method to plan the clamping sequence of the shortest distance. This method also reduces collisions of the arm with other objects on the table when the grasped object moves, avoids the object being overturned or dropped on the floor, and successfully achieves the task of grasping different objects and classifying them.**

***Index Terms*—Humanoid Robot, Kinematics, Action Planning, Pick-and-Place Objects**

I. Introduction

In the field of robotics, a service robot is an important direction of development. To achieve functions such as assisting personnel to place objects, it includes key technologies such as image recognition, gripping strategies, and robotic arm control. How to determine the grip and placement of objects is also a topic worth exploring.

In related research on object grasping, according to the force of the gripping contact area [1] or according to the shape of the object [2, 3], the appropriate picking point is found by considering the geometric constraints of the gripper [4]. Dex-Net [5] uses physical calculation such as the contact friction force of the object or the suction and sealing degree of the object surface, and selects a suction plan with a parallel opening and closing gripper or a vacuum-suction-cup gripper.

In research on object placement, objects are usually put down from a height to a box or table without considering the consequences [1, 2, 6]. However, there has not been any benchmark research. Thus, there has been research on using multiple re-grasping attempts to achieve specific placement of objects [7, 8] and object repositioning [9].

This work was supported in part by the Ministry of Science & Technology of Taiwan, ROC, under grant MOST 111-2221-E-032-029-.

C. C. Liu is with the Department of Electrical and Computer Engineering, Tamkang University, Tamsui Dist., New Taipei City 25137, Taiwan (phone: 886-2-2621-5656 Ext.3226; e-mail: 136382@o365.tku.edu.tw).

Y. C. Lin, and J. J. Tien are with the Department of Electrical and Computer Engineering, Tamkang University, Tamsui Dist., New Taipei City 25137, Taiwan (e-mail: ianlin79928@gmail.com; scania00518@gmail.com).

In related research on arm action planning, Tanaka *et al*. [10] used grids to classify object positions in space and achieve gripping planning and actions through reinforcement learning and learning planning [11]. Prakash and Kamal [12] believed that a robot must be able to work in an environment that changes over time and update instantly and respond to sudden changes. The online prediction planning and execution strategy can do this in a timely manner. Gosselin and Hadj-Messaoud [13] proposed robotic arm path planning with continuous cubic derivatives and artificial neural network algorithms for trajectory interpolation to ensure solutions for arm execution meet safety requirements.

This paper adopts the greedy heuristic (GH) [14, 15] combined with the method of the traveling salesman problem (TSP) [16, 17] to achieve object grasping and plan a better grasping sequence, so that a robot can more efficiently achieve the object gripping function.

II. Humanoid Robot System

A. System architecture

The humanoid robot used to perform operations can be divided into two parts: the industrial PC (IPC) and the SoC FPGA (System on a Chip Field Programmable Gate Array) development board. The robot will first input an image from a camera to the IPC, identify the object through color information, and then control the head motor to align the center of the camera with the object. At this point, the position of the object can be calculated using the motor angle, and then the gripping position that the arm needs to reach is also calculated. This gripping position is sent to the FPGA as the target position of the arm. The FPGA then performs an inverse kinematics calculation to calculate the angles that the motors of each axis of the arm need to rotate. Finally, the motor is controlled to reach the target position for performing the clamping action. The system architecture is shown in Fig. 1. The robot used in this paper has 23 DoF. As show in Fig. 2(a), each arm has 4 DoF. The gripper is shown in Fig. 2(b) is designed in the shape of rounded securities for easy clamping.

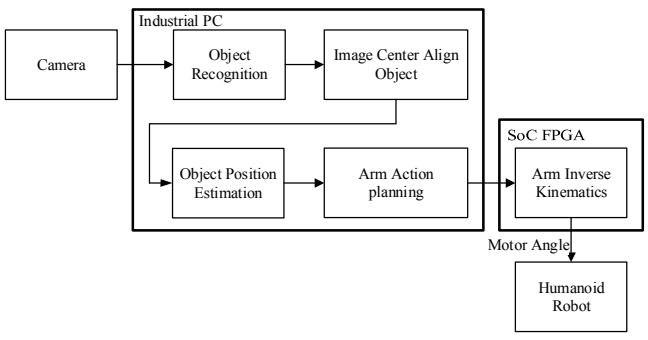
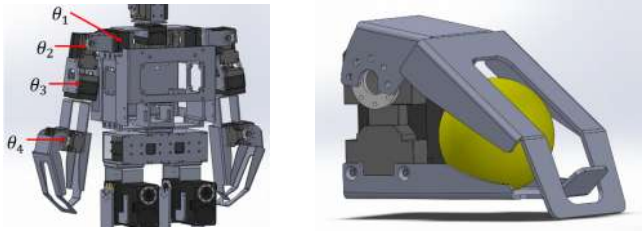


Fig. 1. System architecture.



(a) Arm (b) Gripper

Fig. 2. Arm and gripper design.

B. Object position calculation

The system first uses the image to identify the position of the object on the screen, and then estimates the relative position to the robot according to the geometric relationship. After the center of the image is aligned with the object, it can be calculated based on the known height of the robot and the vertical angle of the head. The relationship is shown in Fig. 3. Here, θ_α is the angle between the camera's viewing angle and the vertical line on the ground, L_h is the height difference between the robot camera and the table object, and L_d is the straight-line distance from the object to the robot. The calculation method is as follows:

$$L_d = \tan(\pi - \theta_\alpha) \cdot L_h . \tag{1}$$

Regarding the straight-line distance from the robot to the object, according to the height of the robot and the angle of the head motor, the horizontal distance R_x and the vertical distance R_y from the robot to the object are calculated using the tangent relationship of the trigonometric function. The relationship is shown in Fig. 4. Here, θ_β is the angle between the camera and the y-axis. The calculation method is as follows：

$$R_x = L_d \cdot \sin(\theta_\beta) \tag{2}$$

$$R_y = L_d \cdot \sin(\pi - \theta_\beta) \tag{3}$$

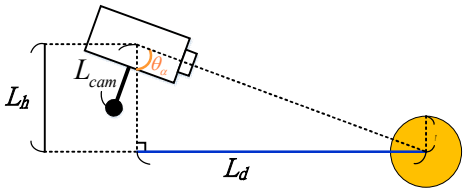


Fig. 3. Schematic diagram of straight-line distance calculation by triangulation.

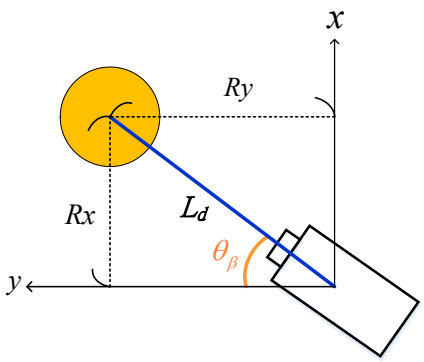


Fig. 4. Schematic diagram of calculation of object distances R_x and R_y.

C. Inverse kinematics

Inverse kinematics refers to using the known coordinates and orientation of the end effector of the robot arm to reverse the angle of the joint of each axis. The humanoid robot has a four-axis robotic arm. Because of the configuration planning of the joint, the movement direction of the third axis will generate a singularity after being given the end effector position when calculating the angles of each axis and there will be infinitely many sets of solutions. As a result, an error occurs when the arm angle plan is generated. In order to reduce the difficulty of use and prevent singularities, the third axis is fixed and the system performs inverse kinematics analysis of the other three axes. The angle of the third axis is set to 0° and 90°. To picture this operation, using the left hand, 0° is when the robot's palm is facing the body and 90° is when the robot's palm is facing down.

1) Third axis at 0°

According to the geometric relationship, the position of the arm end effector can be set, and the calculation starts from the first axis in sequence, followed by the fourth axis, and finally the angle of the second axis. In Fig. 5, L_1 is the length of link 1, L_2 is the length of link 2, and L_3 is the length of link 3. The joint 1 angle θ_1 can be calculated from the end effector coordinates (x_c, y_c, z_c). In order to find the joint 4 angle θ_4, it is necessary to first calculate θ_D by the law of cosines. Finally, the joint 2 angle θ_2 is obtained from the calculated joint 4 angle θ_4. It is calculated as follows:

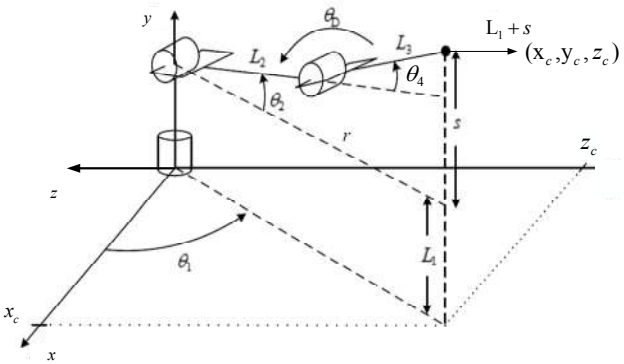


Fig. 5. Analysis diagram of each axis with the third axis of the arm at 0°.

$$\theta_1 = \tan^{-1}\left(\frac{y_c}{x_c}\right), \tag{4}$$

$$\cos\theta_D = \frac{x_c^2 + y_c^2 + (z_c - L_1)^2 - (L_2^2 + L_3^2)}{2L_2L_3}, \tag{5}$$

$$\theta_4 = \tan^{-1}\left(\frac{-\sqrt{1-\cos\theta_D^2}}{\cos\theta_D}\right), \text{ and} \tag{6}$$

$$\theta_2 = \tan^{-1}\left(\frac{z_c - L_1}{\sqrt{x_c^2 + y_c^2}}\right) - \tan^{-1}\left(\frac{L_3\sin\theta_4}{L_2 + L_3\cos\theta_4}\right). \tag{7}$$

2) Third axis at 90°

In Fig. 6, under this link configuration, the rotation axes of joint 1 and joint 4 are in the same direction, and the angle of joint 2 is related to the projection of the joint 4 link. First, the

joint 4 angle θ_4 can be calculated from the position of the end effector and the known conditions with the plane facing the joint 4 axis. Then, the joint 2 angle θ_2 and the joint 1 angle θ_1 can be calculated in sequence, and the calculation method is as follows:

$$\cos\theta_D = \frac{L_2^2 + L_3^2 - ((y_c - L_1)^2 + x_c^2 + z_c^2)}{2L_2L_3}, \tag{8}$$

$$\theta_4 = \tan^{-1}\frac{-\sqrt{1-\cos\theta_D^2}}{\cos\theta_D}, \tag{9}$$

$$\theta_2 = \sin^{-1}\left(\frac{y_c - L_1}{L_2 + L_3\cos\theta_4}\right), \text{ and} \tag{10}$$

$$\theta_1 = \tan^{-1}\frac{L_3\sin\theta_4}{(L_2 + L_3\cos\theta_4)\sin\theta_2} + \tan^{-1}\frac{x_c}{z_c}. \tag{11}$$

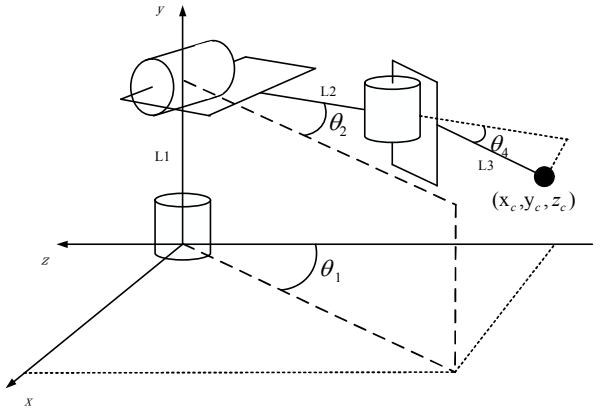


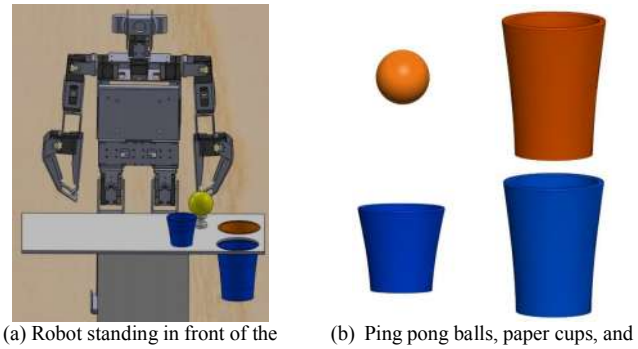
Fig. 6. Analysis diagram of each axis with the third axis of the arm at 90°.

With the formula derived above, the motors of the 1st, 4th, and 2nd axes of the arm joint can be rotated to the angles of θ_1, θ_4, and θ_2, respectively, and the end effector of the arm can be controlled to move to the target position.

III. Object Picking Planning

A. Experimental scenario

Fig. 7 illustrates the situation used in this study. As seen in the picture, the robot is standing in front of the table; the width, depth, and height of the table are 40 cm, 20 cm, and 25 cm, respectively. There are objects of different colors on the table representing different kinds of garbage, including orange table tennis balls and blue paper cups. There are circular holes with diameter 6 cm on both sides of the table, and there are orange and blue trash cans under the holes. The task to be performed is to put the orange table tennis ball in the orange trash can, and the blue paper cup in the blue trash can. Due to the limited arm length of the robot, the maximum gripping depth is 28 cm, which is greater than the depth of the table, so the workable area is shown in Fig. 8. The red framed area is the gripping area, and the blue framed area is the placing area. The width of gripping area is around 20 cm x 20 cm, and the placing area is around 40 cm x 20cm.



(a) Robot standing in front of the table (b) Ping pong balls, paper cups, and trash cans

Fig. 7. Illustration of the experimental scenarios.

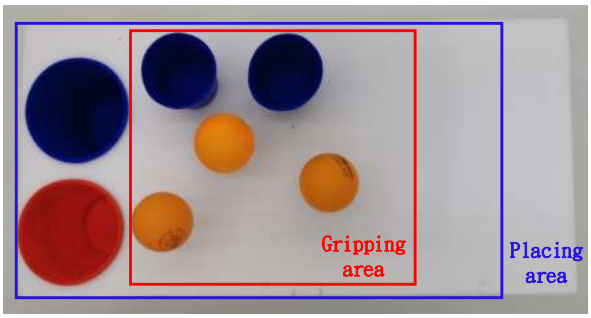


Fig. 8. Picking and placing areas.

B. Object picking function

To ensure the smooth movement of the arm and the claw, the object closest to the trash can is planned to be grasped first. This method ensures that the task can be performed smoothly and it minimizes the risk of other objects being touched when gripping. The method used is a combination of GH and TSP, which is used to find the object closest to the hole for clamping. The purpose is to reduce the distance the arm can move and also reduce the possibility of collision with other objects. Therefore, first object to be clamped will be the one closest to the hole. After grabbing the first object, the system recalculates the positions of the remaining objects on the table to find the next object closest to the opening. It repeats the above method until all objects are clamped.

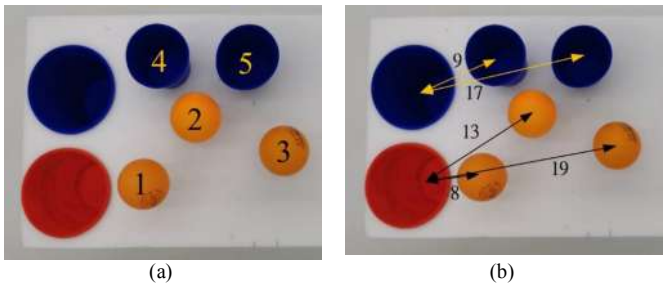
C. Object picking algorithm

First, the positions of the object and of the trash can are identified through the image, and the distance d_i, ($i \in 1\cdots N$) between each object and the trash can is obtained via a Euclidean distance calculation, where N represents the number of objects. Then, the object closest to the trash can is chosen as the reference value D_i for the next clamping sequence. An example scene is shown in Fig. 9. The first grabbed object is directly designated as the object closest to the trash can, so the example will start from the second one. It is calculated as follows:

$$\rho_i = \frac{D_i}{\sum_{j=2}^{N} D_j} = \frac{D_i}{D_2 + D_3 + \cdots + D_N} \text{ and} \tag{12}$$

$$D_i = \frac{N-1}{N} \times \frac{1}{d_i}. \tag{13}$$

Here, ρ represents the weight of grabbing. The higher the value defined, the greater the weight, and it will be grabbed first. The distance in Fig. 9(b) is substituted into the formula if the position of the object is inconvenient during the clamping process. The clamping sequence according to object number in Fig. 9(a) is obtained as $1 \to 4 \to 2 \to 5 \to 3$. Therefore, this method can indeed arrange and plan according to the distance of the objects from left to right.



(a) (b)

Fig. 9. Example of traveling salesman problem application

IV. Experimental Results

A. Object position calculation error

The measurement method used in this study is the trigonometric method. However, due to the limitations of the mechanism, motor gear set, and motor control, each time the humanoid robot turns and stands, there will be different errors. Table I shows the results of object positioning for different positions on the table. The *x* coordinate is the distance along the width of the table, that is, the left and right sides of the robot: 0 represents the standing position of the robot, 50 represents the position 50 mm to the left of the robot, and −50 represents the position 50 mm to the right of the robot. The *y* coordinate is the distance along the depth of the table, which is the front of the robot. The data in Table II are the errors between the positioning results and the actual position according to Table I. It can be seen that there is a smaller error. These errors are all within the allowable range and the arm can still grip objects normally. However, at the position 50 mm in front of the robot, especially on the left and right sides far away from the robot, there is a very large error. This is because, when the ping-pong ball and paper cup are too close to the robot, the camera cannot see the whole ping-pong ball or paper cup, and part of the viewing angle is blocked by the mechanism of the robot body, resulting in inaccurate positioning, so the object cannot be placed too close to the robot.

TABLE I
Locations of Object Recognition (Unit: mm).

location	140	100	50	0	−50	−100	−140
50	(75, 75)	(62, 85)	(59, 42)	(54, 1)	(−56.4,−44)	(-60,−83)	(−66,−102)
100	(108, 131)	(105, 93)	(100, 46)	(100, 2)	(−102,−47)	(−100,−91)	(−106,−126)
150	(148, 138)	(149, 95)	(152, 50)	(148, −7)	(148,−52)	(−146,−102)	(−147,−129)
200	(192, 144)	(194, 104)	(195, 54)	(196, 3)	(−198,−54)	(−196,−104)	(−195,−146)

TABLE II
Location Errors of Object Recognition (Unit: mm).

location	140	100	50	0	−50	−100	−140
50	42	7	1.5	3.8	0	10	26
100	2	1	3	0.2	0	6	7
150	5	4	2	2	1	2	9
200	4	3	4	4	1	2	1

B. Trash can on the left side of the table

In this experiment, three ping pong balls and two paper cups will be placed on the table, and the red and blue trash cans are on the left side of the table, as shown in Fig. 10. The robot first records the position of the object and then calculates the distance from the trash can. The numbers in Fig. 10 represent the straight-line distance from the item to the trash can. The shortest distance is 9 cm to the leftmost ping pong ball and the farthest is 19 cm to the rightmost ping pong ball. After end planning, the clamping sequence is as shown in Fig. 11, and the numbers are the the clamping sequence. The robot will pick up the object closest to the trash can. The gripping process of the robot is shown in Fig. 12. The first step is to start with the No. 1 table tennis ball on the far left, as shown in Fig. 12(a). It clamps the first ball, then, as shown in Fig. 12(b), it clamps the No. 2 paper cup that is closest to the leftmost blue trash can. Next, as shown in Fig. 12(c), the No. 3 ping pong ball that is closest to the trash can on the table is clamped. As shown in Fig. 12(d), the No. 4 paper cup that is closest to the hole on the table is clamped. As shown in Fig. 12(e), the last No. 5 table tennis ball left on the table is clamped into the trash can. Finally, the clamping task is completed, and there is no object left on the table, as shown in Fig. 12(f).

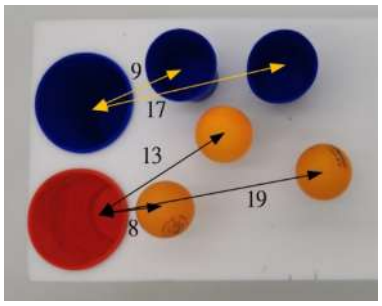


Fig. 10. Distances between the objects and the trash cans.

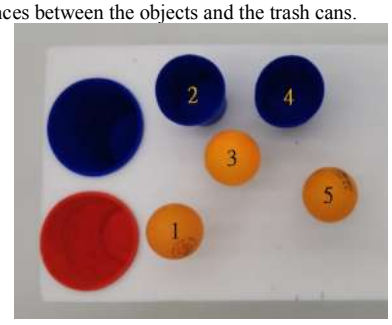
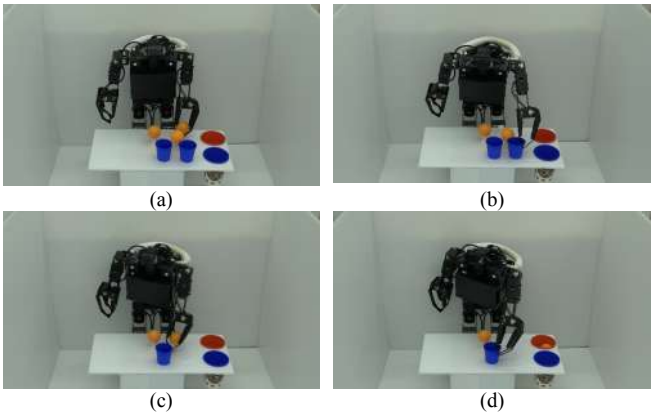
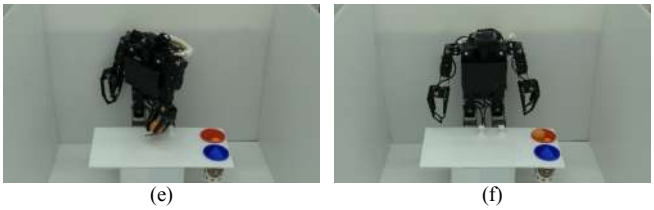


Fig. 11. Object picking order.



(a) (b)

(c) (d)



(e) (f)

Fig. 12. Experiment with the trash can on the left side of the table.

C. Trash cans located on both sides of the table

In this experiment, three table tennis balls and two paper cups are placed on the table. The red trash cans are on the left side of the table and the blue trash cans are all on the right side of the table, as shown in Fig. 13. The robot first records the position of the object and then calculates the distance from the trash can. The numbers in Fig. 13 represent the straight-line distances from the item to the trash can. The shortest distance, 6 cm, is to the leftmost ping pong ball and the farthest, 16.5 cm, is to the rightmost ping pong ball. After end planning, the clamping sequence is shown in Fig. 14, and the numbers are the clamping sequence. The robot will pick up the object closest to the trash can. The picking process is shown in Fig. 15. The first step is to start with the No. 1 table tennis ball on the far left, as shown in Fig. 15(a). After clamping the first ball, as shown in Fig. 15(b), the second table tennis ball, which is the second closest to the hole, is clamped. Next, in order to shorten the movement path of the arm gripping, as shown in Fig. 15(c), the No. 3 paper cup that is closer to the current arm is gripped. As shown in Fig. 15(d), it is the No. 4 paper cup that is closer to the blue trash can. As shown in Fig. 15(e), the last No. 5 ping pong ball left on the table is clamped into the trash can. Finally, the clamping task is completed, and there is no object left on the table, as shown in Fig. 15(f).

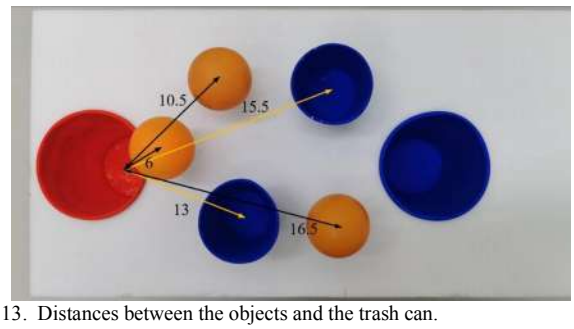


Fig. 13. Distances between the objects and the trash can.

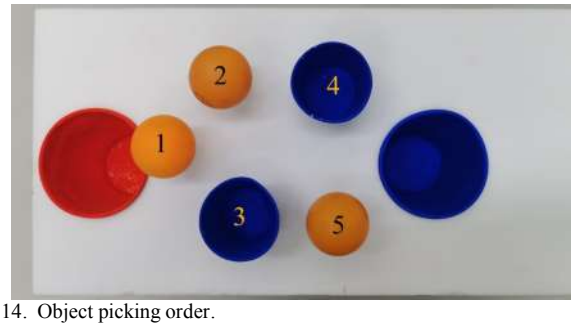
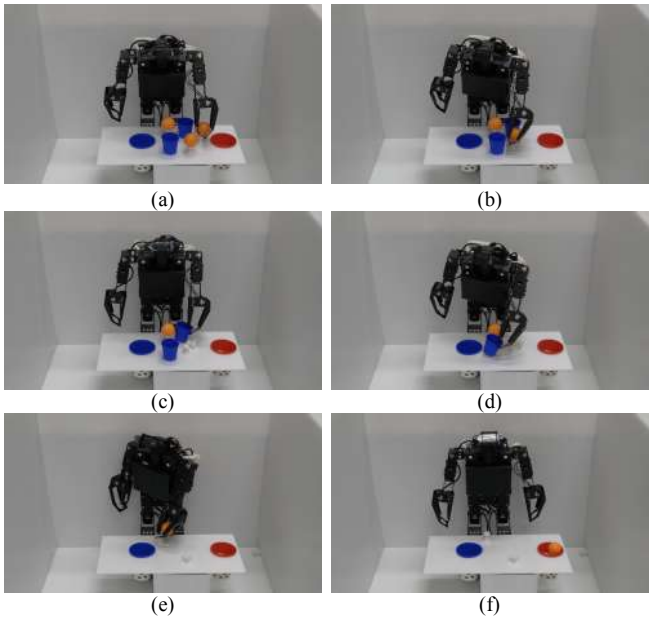


Fig. 14. Object picking order.



(a) (b) (c) (d) (e) (f)

Fig. 15. Experiment with trash cans on both sides of the table.

V. Conclusion

This study used simple image color recognition and geometric relationships to locate objects on a table and simplified the four-axis arm of a humanoid robot into a mathematical model of a three-axis arm according to the task requirements. Inverse kinematics were used to control the robot with relatively simple arithmetic. Finally, the object clamping order was determined according to the distance between the object and the trash can. When the two trash cans were both on the left side of the table or were on either side, the gripping task was successfully completed.

This study used only one robotic arm for picking and placing objects. In the following study, two robotic arms will be used together to increase the range and efficiency of the gripping. However, the path planning methods of both arms must be considered together to avoid mutual interference and collision during the clamping processes of the two arms. In the future, deep-learning training can also be performed on target objects in images. With an object recognition algorithm operated by a convolutional neural network, it is only necessary to add a variety of objects of the same type to the training data to achieve class-level object recognition, marking the location of the object, and then positioning it. It is not necessary for the user to pre-select which features to extract, but to make the model learn the features of the target object from large-scale data.

References

[1] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *Robotics: Science and Systems*, pp. 58-72, 2017.

[2] D. Morrison, J. Leitner, and P. Corke, "Closing the loop for robotic grasping: a real-time, generative grasp synthesis approach," *Robotics: Science and Systems*, pp. 21-31, 2018.

[3] D. Guo, T. Kong, F. Sun, and H. Liu, "Object discovery and grasp detection with a shared convolutional neural network," *IEEE International Conference* on *Robotics and Automation*, pp. 2038–2043, 2016.

[4] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics* Research, vol. 34, no. 5, pp. 705–724, 2015.
[5] J. Mahler, M. Matl, V. Satish, M. Danielczuk, B. DeRose, S. McKinley, and K. Goldberg, "Learning ambidextrous robot grasping policies," *Science Robotics*, vol. 4, no. 26, 2019.
[6] K. Wada, K. Okada, and M. Inaba, "Joint learning of instance and semantic segmentation for robotic pick-and-place with heavy occlusions in clutter," *International* Conference *on Robotics and Automation*, pp. 9558–9564, 2019.
[7] W. Wan, H. Igawa, K. Harada, H. Onda, K. Nagata, and N. Yamanobe, "A regrasp planning component for object reorientation," *Autonomous Robots*, vol. 43, no. 5, pp. 1101–1115, 2019.
[8] W. Wan, M. T. Mason, R. Fukui, and Y. Kuniyoshi, "Improving regrasp algorithms to analyze the utility of work surfaces in a workcell," *IEEE International Conference on Robotics and Automation*, pp. 4326-4333, 2015.
[9] A. Nguyen, D. Kanoulas, D. G. Caldwell, and N. G. Tsagarakis, "Preparatory object reorientation for task-oriented grasping," *IEEE International Conference on Intelligent Robots and Systems*, pp. 893–899, 2016.
[10] T. Tanaka, T. Kaneko, M. Sekine, V. Tangkaratt and M. Sugiyama, "Simultaneous Planning for Item Picking and Placing by Deep Reinforcement Learning," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA, 25-29 Oct. 2020, pp. 9705-9711.
[11] Y. Deng, X. Guo, Y. Wei, K. Lu, B. Fang, D. Guo, H. Liu, F. Sun, "Deep Reinforcement Learning for Robotic Pushing and Picking in Cluttered Environment," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, 4-8 Nov. 2019, pp. 619-626.
[12] N. R. Prakash and T. S. Kamal, "Intelligent planning of trajectories for pick-and-place operations," *Smc 2000 conference proceedings. IEEE international conference on systems, man and cybernetics,* Nashville, USA, 8-11 Oct. 2000, pp. 55-60.
[13] C. M. Gosselin, and A. Hadj-Mrssaoud, "Automatic Planning of Smooth Trajectories for Pick-and-place Operations," *ASME Journal of Mechanical Design*, Vol. 115, pp.450-456,1993.
[14] R. Liu, J. Cui and Y. Song, "Forward Greedy Heuristic Algorithm for N-Vehicle Exploration Problem (NVEP)," *8th International Symposium on Computational Intelligence and Design*, Beijing, China, 12-13 Dec. 2015, pp. 243-246.
[15] D. Zhang, S. Guo, W. Zhang and S. Yan, "A novel greedy heuristic algorithm for university course timetabling problem," *Proceeding of the 11th World Congress on Intelligent Control and Automation*, Shenyang, China, 29 June - 4 July. 2014, pp. 5303-5308.
[16] A. M. Sheveleva and S. A. Belyaev, "Development of the Software for Solving the Knapsack Problem by Solving the Traveling Salesman Problem," *IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering*, St. Petersburg, Russia, 26-29 Jan. 2021, pp. 652-656.
[17] J. Liu and W. Li, "Greedy Permuting Method for Genetic Algorithm on Traveling Salesman Problem," *8th International Conference on Electronics Information and Emergency Communication*, Beijing, China, 15-17 June. 2018, pp. 47-51.

Chih-Cheng Liu was born in Taiwan, in 1982. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Tamkang University, Taiwan, in 2004, 2006, and 2014 respectively. He joined the Department of Electrical and Computer Engineering of Tamkang University in 2017 and now is an assistant professor. His research interests include humanoid robots, motion control, robotic applications, FPGA design, and SOPC design.

Yi-Chung Lin was born in Taiwan, in 1990. He received the B.S., M.S., and Ph.D. degrees in electrical engineering from Tamkang University, Taiwan, in 2013, 2015, and 2022 respectively. He joined the Intelligent Automation and Robotics Center of Tamkang University in 2022 and now is a postdoctoral fellow. His major research interests include humanoid robot, intelligent systems, and robotic applications.

TJen-Jia Tien was born in Taiwan, in 1996. He received the B.S. degrees in electrical engineering from National Kaohsiung University of Science and Technology, Taiwan in 2020, and M.S. degrees in electrical engineering from Tamkang University, Taiwan, in 2022. His research interests include humanoid robots, robots mechanism, electric circuit.