

Received May 8, 2022, accepted May 31, 2022, date of publication June 10, 2022, date of current version June 23, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3182114

Generic Development of Bin Pick-and-Place System Based on Robot Operating System

CHING-CHANG WONG^{1,2}, CHI-YI TSAI^{1,2}, (Senior Member, IEEE), REN-JIE CHEN^{1,2},
SHAO-YU CHIEN^{1,2}, YI-HE YANG^{1,2}, SHANG-WEN WONG^{1,2},
AND CHUN-AN YEH^{1,2}

¹Department of Electrical and Computer Engineering, Tamkang University, New Taipei City 25137, Taiwan

²Doctoral Program in Robotics, College of Engineering, Tamkang University, New Taipei City 25137, Taiwan

Corresponding author: Ching-Chang Wong (wong@ee.tku.edu.tw)

This work was supported in part by the Ministry of Science and Technology (MOST), Taiwan, under Grant MOST 109-2221-E-032-038 and Grant MOST 110-2221-E-032-046.

ABSTRACT Bin pick-and-place is an important topic in factory automation and warehouse automation. In this paper, a bin pick-and-place system based on robot operating system (ROS) is implemented to make a six-degree-of-freedom (6-DOF) robot manipulator to complete multiple pick-and-place tasks. The proposed system uses ROS to integrate an object perception module and a pick-and-place module, where the former uses an RGB-D camera to capture images inside the bin, and the latter controls a 6-DOF robot manipulator and two self-made vacuum tools. To estimate the pose of the target object, a YOLOv4 object detector is implemented, and an object sorting method is proposed to find the target object in the image. Then, a pose estimation method based on computer aided design (CAD) is proposed to estimate the pose of target object. To perform the object pick-and-place operations, a coordinate transformation node is designed to transfer the pose of the target object into the workspace. Then, a link distance-based bin collision avoidance method is proposed to avoid collisions. Finally, the angle of the 1-DOF vacuum tool and the picking and placement poses of the robot manipulator are obtained from the result of the bin collision avoidance and the pose of the target object. In this study, a total of ten ROS nodes are designed, and the solutions that make each function easier to implement and reproduce are proposed. In the experiments, we set up four experiments with two task types and two object types to verify the effectiveness of the implemented bin pick-and-place system.

INDEX TERMS Bin pick-and-place, robot operating system, object detection, pose estimation, collision avoidance.

I. INTRODUCTION

The integrated applications of robot manipulator have been developed in automation technology for many years, and bin pick-and-place is still one of the challenging research issues. The ability of a robot manipulator to autonomously pick and place objects from a cluttered bin can be applied to many scenarios, such as pick-and-place objects in warehouse automation, or pick-and-place parts in factory automation [1]. One of the goals of research related to automation technology is to establish a system with the ability to work independently, and different types of objects also bring different challenges. The design of the bin pick-and-place system mainly includes two parts: object perception and object pick-and-place. The

purpose of the object perception part is to detect the object or estimate its pose so that the robot manipulator can correctly complete the pick-and-place task. In general, the types of common objects can be divided into textured objects and texture-less objects according to whether they have surface textures. For example, most groceries for warehouse automation are textured objects with rich surface textures, while parts for factory automation are mostly texture-less objects. Therefore, different object detection methods are usually designed under different object types and scenes. On the other hand, the object pick-and-place part converts the result of the object detection into the motion command of the robot manipulator when it performs a pick-and-place task. The types of pick-and-place tasks can be divided into orientation non-required and orientation required depending on whether the objects need to be placed in a particular pose.

The associate editor coordinating the review of this manuscript and approving it for publication was Shun-Feng Su¹.

For example, in warehousing automation, objects only need to be classified, not necessarily arranged neatly. Therefore, objects only need to be classified to complete the task. However, in factory automation, it is often necessary to place objects in a specific pose. Therefore, in addition to object detection, it is also necessary to perform pose estimation on the target object and calculate the picking pose and placing pose of the robot manipulator. The bin is an obvious obstacle during the bin pick-and-place task, and there will be many objects very close to the edge of the bin, which makes the robot manipulator prone to collisions during picking and placing. Therefore, the collision avoidance between the robot manipulator and the bin is also an important topic in the bin pick-and-place task. The realization of collision avoidance can ensure the safety of the equipment when performing tasks.

In the research of object detection, many studies have discussed the topic of feature detection [2]–[4], but most of them use feature-based object detection methods to deal with textured objects in simple scenes. If the feature-based method is used to deal with complex scenes or texture-less objects, the performance of object detection is limited. In contrast, learning-based object detection methods are more general and can handle both textured and texture-less objects. For example, Blank *et al.* used convolutional neural networks to detect texture-less metal workpiece and used the detection results for segmentation and pose estimation, and then used a robot to perform the pick-and-place task [5]. Li *et al.* proposed a modified YOLOv2 to detect objects such as cups and beverage cans, which are then grasped by a humanoid robot [6]. Schwarz and Behnke use deep learning methods to carry out detection and segmentation of various groceries, and the robot is used to perform bin pick-and-place task [7]. Lee *et al.* used YOLOv2 to detect texture-less hexagonal and circular rings, and then used Iterative Closest Point (ICP) algorithm for object pose estimation [8]. Many other studies have also discussed the topic of learning-based object detection [9]–[11], and these object detection methods can also be easily implemented in our bin pick-and-place system.

In the research of object pose estimation, Le and Lin proposed a deep learning-based instance segmentation method to extract point cloud data of planar objects. In their study, the target object is a USB flash drive with packaging, and 3D point cloud data is obtained according to the segmentation result for object pose estimation and picking [12]. Song *et al.* proposed a CAD-based pose estimation method that uses a voting scheme to identify and estimate the poses of three different industrial parts [13]. Wu *et al.* also used CAD-based methods to estimate the poses of four different objects, including three texture-less objects, and used the pose estimation results in the bin pick-and-place task [14]. Li *et al.* proposed a Partitioned Viewpoint Feature Histogram (PVFH) method, which divides the point cloud of an object into two parts for pose estimation, and achieved good results in the experiments on industrial parts [15]. Many other studies

have also investigated the topic of object pose estimation [16]–[18], and these pose estimation methods can also be easily implemented in our bin pick-and-place system.

In the research of object pick-and-place, Song *et al.* proposed a method to avoid obstacles based on 3D vision during task execution, using 3D environment monitoring cameras to track surrounding people and obstacles to avoid collisions [19]. Jeffrey and Ken proposed the Grasp Quality Convolutional Neural Network (GQ-CNN) method to train a neural network on synthetic data to have a higher quality prediction result of the grasping position, and using imitation learning to avoid collisions with adjacent objects in the bin pick-and-place task [20]. Zeng *et al.* divided the pick-and-place task into two steps: picking and classification, and adopted the method of probability maps of the affordances to select one action from four grasping primitive actions. This method can also be applied to grasping unknown objects [21]. Lin and Cong used a point cloud deep network modified from PointNet to identify the surface of an object from a partial view, and the grasping position is estimated based on the object model and grasping configuration sets [18]. Many other studies have also discussed the topic of object pick-and-place [22]–[25], and we modularize parts of the pick-and-place task into ROS nodes, which give each node the flexibility to replace different method.

Among the studies on different bin pick-and-place task types, most studies discuss different types of objects according to different scenarios, such as textured objects [6], [12], [26], [27], [28], [29], textured groceries [7], [20], [21], and texture-less objects [5], [8], [11], [14], [15]. In the complex and diverse scenarios of warehousing and factory, it is usually necessary to design many different individual methods to handle these different tasks. At present, there is also a lack of comprehensive solutions for different task types and object types. Most studies on bin pick-and-place tasks rarely discuss in detail the transformation between image data and manipulator's motion commands and the collision avoidance between the robot manipulator and neighboring objects. Based on the above discussion, this paper proposes an applicable solution for different task types and object types, and describes in detail the object perception, object pick-and-place, and system integration required for the bin pick-and-place system.

In the design of object perception, we used a learning-based approach for object detection and a CAD-based approach for object pose estimation in this study. From previous related studies, it can be found that feature-based object detection methods are mostly used to detect textured objects, while learning-based object detection methods are suitable for various types of objects, which shows the powerful performance of learning-based methods. There are also some learning-based methods for object pose estimation, but most of them require a lot of computing resources. Therefore, a CAD-based pose estimation method is adopted in this paper, which obtains the point clouds of detected objects via the outputs of the object detection,

and obtains the pose of objects via the alignment of point cloud.

In the design of object pick-and-place, a collision detection method based on the distance between robot links and the bin is used in this paper, and the collision is avoided by adjusting the picking pose. With the known bin positions and links parameters, the proposed collision avoidance method only requires simple calculations to efficiently obtain a safe picking pose. In addition, two vacuum tools are designed in this study. One of them has one degree of freedom to adjust the angle of the suction cup according to the picking pose, so that the object can be placed in a specific pose. The other has a special small suction cup that can suck smaller objects to ensure successful suction without multiple objects being sucked at the same time.

In the system integration work, Robot Operate System (ROS) is used in this study to design and implement the proposed bin pick-and-place system. This paper describes its architecture in detail. ROS is an open source operating system specially developed for designing robot systems. Since ROS can transfer information between different computers or different robots, using ROS to develop robot systems can improve the efficiency of system development. In related applied researches, Kumra *et al.* used ROS to construct a neural network-based robot grasping system [30]. Hernandez-Mendez *et al.* used ROS to construct a 3-DOF robotic arm [31]. Wang *et al.* used ROS to construct a mobile robotic arm platform that can be used to detect and grasp the radiation sources [32]. Wong *et al.* used ROS to implement the motion planning, manipulation planning, and object grasping of the robot manipulator [33]–[35]. Zhang *et al.* used ROS to construct a conveyor belt pick-and-place system [36]. It used MoveIt to plan the robot's motion in the collision avoidance part. Although it can avoid collisions during the movement, it cannot handle the situation that the target pose given by the visual-related nodes will collide. However, the situation is more common in the bin pick-and-place system, so a collision avoidance method is proposed in this paper to solve this problem. Tavares and Sousa proposed a pick-and-place system including perception, control, and task strategy levels [37]. However, they also did not solve the collision problem of the target pose, and they did not consider the stacking of objects. Thus, an object sorting method is proposed in this paper to select feasible objects from the stacked objects for picking. We can see the application potential of ROS for the development of robotic systems. Therefore, this study uses ROS to integrate various functions to complete the design of the proposed bin pick-and-place system.

The rest of this paper is organized as follows. Section II presents the system architecture of the proposed bin pick-and-place system and describes the relationship between each module. Section III introduces the ROS nodes of the bin pick-and-place task in this paper, including task strategy node, object detection node, object sorting node, pose estimation node, coordinate transformation node, and the bin collision

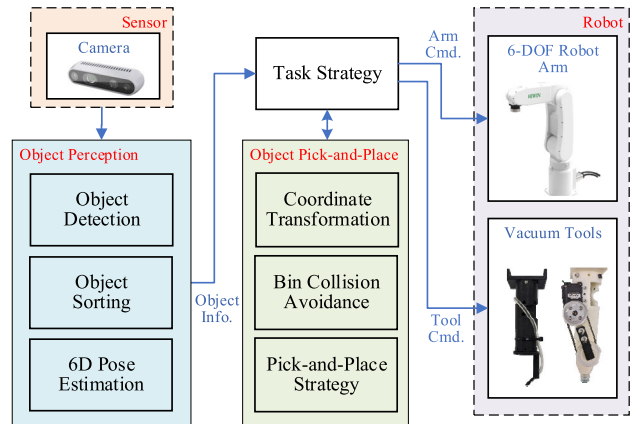


FIGURE 1. System architecture diagram of the proposed bin pick-and-place system.

avoidance node. Section IV introduces and discusses the experimental results of four pick-and-place tasks based on whether the object is texture or texture-less and whether the task is orientation non-required or required to illustrate the effectiveness of the proposed system. Finally, Section V presents the conclusions of this study.

II. SYSTEM STRUCTURE

The overall architecture diagram of the bin pick-and-place system proposed in this paper is shown in Fig. 1. The Intel RealSense D435i camera is used as the sensor of the system to provide the RGB image and point cloud data of the objects in the bin, and the outputs of the system are control commands of a 6-DOF robot manipulator and a vacuum tool. In terms of vacuum tool, two self-made vacuum tools are designed in this paper to pick and to place different types of objects. The proposed bin pick-and-place system consists of a task strategy to control the task flow, an object perception module to obtain the object information in the bin, and an object pick-and-place module to transform the object information into the commands of the robot manipulator and the vacuum tool. Through the above functions, the robot manipulator can perform bin pick-and-place tasks efficiently.

In order to achieve fast and flexible development, ROS is used to construct the proposed bin pick-and-place system. The computation graph between the designed ROS nodes of the proposed bin pick-and-place system is shown in Fig. 2, where the “camera manager node” continuously publishes RGB images and point cloud data to the “image topic” and “points topic”, respectively. The “arm manager node” continuously publishes the state of the robot manipulator to the “current state topic”. The “task strategy node” mainly communicates with service nodes of the object perception module and the object pick-and-place module, and sends commands to the “arm manager node” and the “tool manager node”.

In object perception module, the “object detection node” inputs the detection results to the “object sorting node” to select a target object. The “pose estimation node” uses

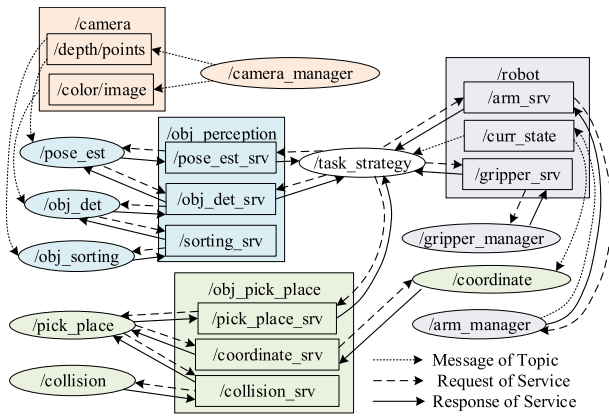


FIGURE 2. Computation graph between the designed ROS nodes.

the bounding box of this target object to capture its point cloud data, and outputs the pose estimation result of this target object. In the object pick-and-place module, the “pick-and-place strategy node” first obtains the target pose of the object in the working coordinate system via the “coordinate transformation node”, and then calculates a safe target pose via the “bin collision avoidance node”. The outputs of “pick-and-place strategy node” include the angle of suction cup, picking pose, and placement pose.

III. ROS NODES DESIGN

In the ROS architecture, a ROS node can publish or subscribe to some messages as a ROS topic, and can also request or response some messages as a ROS service. There is no hierarchical relationship between ROS nodes, and each node communicates with each other through the ROS master. Therefore, this architecture makes the design of each ROS node more simplistic and modular, and also greatly reduces the complexity of system development.

In this study, the proposed bin pick-and-place system contains ten ROS nodes, of which three hardware manager nodes are used to manage the camera, manipulator, and vacuum tool, respectively. They publish device messages to the system and provide services to control the device. The remaining seven ROS nodes provide the core functions of the bin pick-and-place system, including one task strategy node that controls the task flow, and three object perception nodes, and three object pick-and-place nodes.

A. TASK STRATEGY NODE

The task strategy node is used to control the process of bin pick-and-place tasks. According to the states of the robot manipulator and the task procedure, the robot manipulator is controlled to perform corresponding actions, and the required information is obtained through the topic or service of each node in the object perception module or object pick-and-place module. Fig. 3 shows the flow chart of the proposed bin pick-and-place system, which can handle both orientation non-required and required tasks, making the robot easy

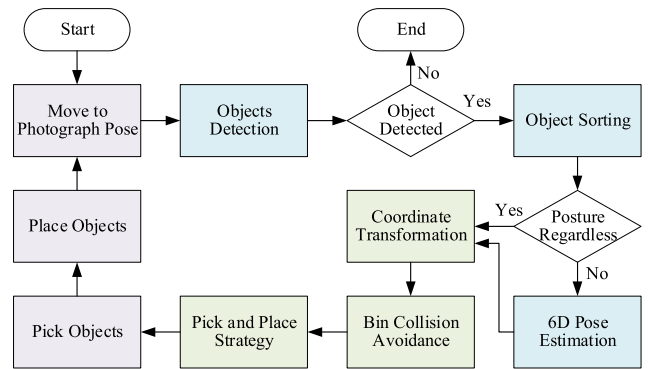


FIGURE 3. Flow chart of the proposed bin pick-and-place system.

to apply to most bin pick-and-place tasks. The detailed description of each step is described as follows:

- Step 1: Complete the initial settings of the system and start the bin pick-and-place task.
- Step 2: Move the robot manipulator above the bin to obtain RGB images and point clouds.
- Step 3: Perform object detection to detect the class and location of all objects in the RGB image.
- Step 4: Determine whether there is an object detected, if there is no object, go to Step 14.
- Step 5: Perform object sorting to select a target object.
- Step 6: Determine whether the task is orientation required task or not, if it is orientation non-required task, go to Step 8.
- Step 7: Perform pose estimation to estimate the pose of the target object.
- Step 8: Execute coordinate transformation to transform the pose of the target object from the camera coordinate system into the working coordinate system.
- Step 9: Execute bin collision avoidance method to plan a safe picking pose that can prevent the collision when picking object.
- Step 10: Execute pick-and-place strategy to calculate the angle of suction cup, picking pose and placement pose.
- Step 11: Move the robot manipulator to the picking pose and pick the object.
- Step 12: Move the robot manipulator to the placement pose and place the object.
- Step 13: Complete an object pick-and-place task and return to Step 2.
- Step 14: End the bin pick-and-place task.

B. OBJECT DETECTION NODE

The object detection node is used to detect the class and location of the object in the RGB image, and to select a target object through the object sorting service. The computation graph of the object detection node is shown in Fig. 4. In this paper, YOLOv4 is used to implement the object detection function. First, an empty request is used to trigger the object detection service. Then, the center position of each object is calculated using the object bounding box detected by YOLOv4, and it was used as the request of the object sorting

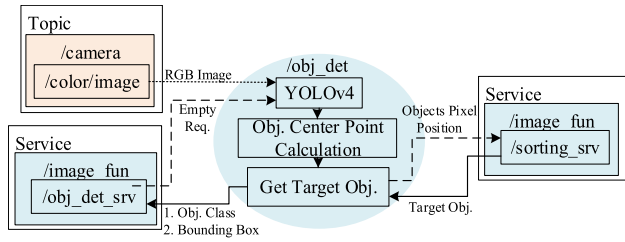


FIGURE 4. Computation graph of the object detection node.

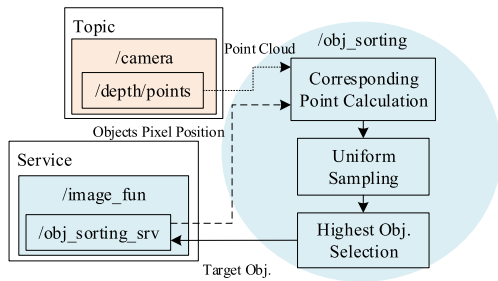


FIGURE 5. Computation graph of the object sorting node.

service to obtain a target object. The response of this node includes the class and bounding box of target object. In the proposed bin pick-and-place system, the method of object detection node can be replaced with other object detection methods according to the conditions of the objects and the environments, and can even be replaced with other semantic segmentation or instance segmentation methods.

C. OBJECT SORTING NODE

The object sorting node is used to select a target object from multiple detected objects. Since the objects are randomly stacked in the bin, the robot manipulator may collide with other objects when picking the target object, or accidentally pick up the objects that cover the target object. Therefore, the uppermost object is selected based on the height of the object in the bin, and this object is used as the target object for the bin pick-and-place task.

The computation graph of the object sorting node is shown in Fig. 5. First, the request of the object pixel position list is used to trigger the object sorting service, and the corresponding surface center point of each object is found in the point cloud data. Second, in order to avoid the influence of noise, uniform sampling is performed with a radius of 1 cm on the surface center point of the object. Third, because the camera is placed directly above the center of the bin, the highest object is the smallest z value of surface center point in the camera coordinate system. Finally, the response of the object sorting service is the index of the highest object in the requested list.

D. POSE ESTIMATION NODE

The pose estimation node is used to calculate the pose of the object to ensure that the object can be picked and placed

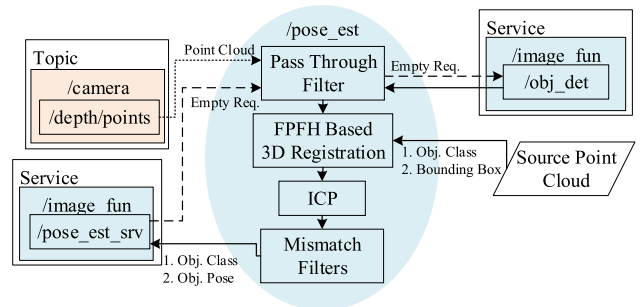


FIGURE 6. Computation graph of the proposed pose estimation node.

correctly. The proposed pose estimation method is realized by using the Point Cloud Library (PCL), and cooperates with the object detection node to capture the point cloud data of the target object in the bin.

The computation graph of the pose estimation node is shown in Fig. 6. First, it uses the bounding box of the target object provided by the object detection service to obtain the pixel position of the object in the 2D image. Second, a pass through filter is used to filter out the point cloud outside the bounding box and obtain the target point cloud of the object. Third, the object class provided by the object detection service is used to obtain the corresponding source point cloud, and uses the alignment method based on the Fast Point Feature Histograms (FPFH) [38] to align the source point cloud and the target point cloud to obtain the pose of the target object. FPFH is a feature vector to represent 3D points. Its idea is to calculate the Point Feature Histogram (PFH) of each point in the k neighbors of the query point separately, and then calculate the weighted summation of all PFHs into the final fast point feature histogram. Fourth, the Iterative Closest Point (ICP) algorithm is used to fine align the target object [39]. The key idea of ICP is to obtain the corresponding point pairs between the source point cloud and the target point cloud, a rotation and translation matrix is constructed based on the corresponding point pairs, and the source point cloud is transformed into the coordinate system of the target point cloud by using the obtained matrix. Finally, a mismatch filter is used to filter out obviously wrong alignment results, and the alignment result is the pose of the target object.

In this study, the source point clouds of the objects are generated by the CAD model sampling. In the actual implementation of the pick-and-place task, the target point cloud obtained by the camera only contains the part of the object facing the camera. Therefore, the source point clouds of the objects are divided into front and back two parts based on different sides of view. An example of dividing the source point cloud of a non-textured object into two parts is shown in Fig. 7. Although the proposed method increases the computational complexity, it can improve the accuracy of the pose estimation.

In terms of the mismatch filter, it uses the class of the object to determine whether to accept the alignment result. Since

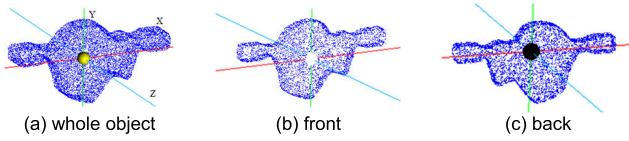


FIGURE 7. The complete point cloud of the object and the source point cloud of the two parts of the object.

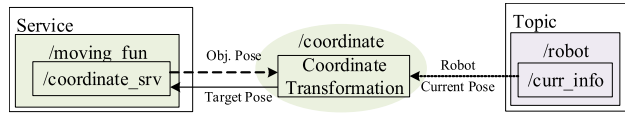


FIGURE 8. Computation graph of the proposed coordinate transformation node.

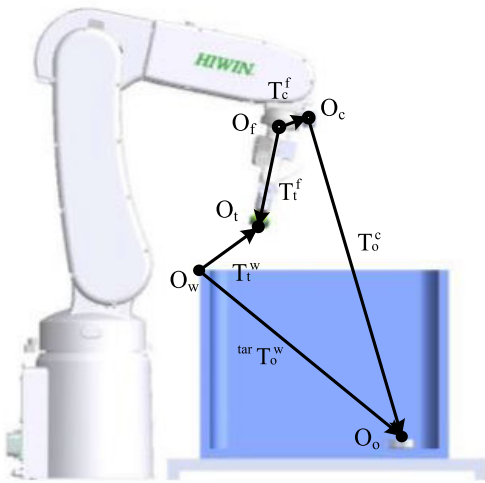


FIGURE 9. Correlation diagram of five coordinate systems of the coordinate transformation.

the shapes of the front and back of some objects are very similar, an object is divided into two classes in this study, and the result of object detection can be used to judge whether the alignment result is acceptable. For example, the z-axis of the object in Fig. 7 is perpendicular to the plane of the front and the back, and the direction is from the back to the front. Thus, when the object is detected as the front part, the alignment result with negative z-axis values is not accepted. Similarly, when the object is detected as the back part, the alignment result with positive z-axis values is not accepted. Therefore, using a mismatch filter can remove obviously wrong alignment results to improve the accuracy of the pose estimation method.

E. COORDINATE TRANSFORMATION NODE

The coordinate transformation node is used to transform the pose of target object in the camera coordinate system into the working coordinate system. The computation graph of the coordinate transformation node is shown in Fig. 8. The relationship between five coordinate systems of the bin pick-and-place system is shown in Fig. 9, where O_w , O_t , O_f , O_c , and O_o represent the origin of the working, tool,

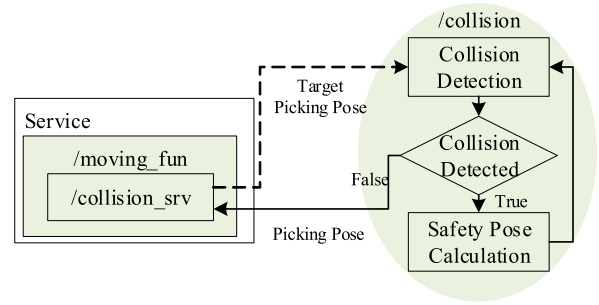


FIGURE 10. Computation graph of the proposed bin collision avoidance node.

flange, camera and object coordinate system, respectively. ${}^{tar}T_t^o$, T_w^t , T_f^t , T_c^f and T_o^c are the transformation matrices of the object in the working coordinate system, the tool in the working coordinate system, the tool in the flange coordinate system, the camera in the flange coordinate system, and the object in the camera coordinate system, respectively. Transformation matrices used in this paper are all 4-by-4 matrices including position and orientation. The direction of the arrow in Fig. 9 represents the transformation direction of the transformation matrix. Therefore, based on the transformation sequence from O_o to O_w , the equation of the transformation matrix of the object in the working coordinate system ${}^{tar}T_o^w$ can be expressed as

$${}^{tar}T_o^w = T_w^t \left(T_t^f \right)^{-1} T_c^f T_o^c \tag{1}$$

where $(T_t^f)^{-1}$ is the transformation matrix of the flange in the tool coordinate system, and ${}^{tar}T_o^w$ is the transformation matrix of the object in the working coordinate system, and also the response of the coordinate service.

F. BIN COLLISION AVOIDANCE NODE

The bin collision avoidance node is used to calculate the picking pose for the robot manipulator to safely pick the object. In the bin pick-and-place task, in addition to being able to accurately pick the object in the bin, the robot manipulator also needs to avoid collision with the bin. Since the wrist joint of the robot manipulator has a certain width, if the object is located on the edge of the bin, it will easily collide during the picking process. Therefore, the design of bin collision avoidance is essential. The computation graph of the bin collision avoidance node is shown in Fig. 10. First, the collision detection is performed on the target pose. Then, if a collision is detected, the picking pose of the robot manipulator is adjusted to avoid the collision. Therefore, the design of the proposed bin collision avoidance node includes two parts: 1) collision detection, and 2) collision avoidance. They are described in the following subsections.

1) COLLISION DETECTION

Since the link of tool is easier to collide with the bin than other links of robot, we simplified the robot model for collision

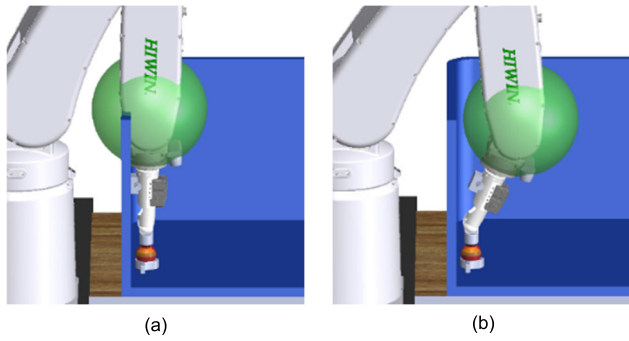


FIGURE 11. Schematic diagram of the simplified sphere of wrist and TCP used for collision detection and the adjusted collision-free pose of robot.

detection into two spheres centered at the wrist joint and the Tool Center Point (TCP) in this paper. As shown in Fig. 11, the green and red spheres, respectively, are used to simplify the representation of the wrist joint and TCP. It indicates that avoiding the collision of the two spheres with the bin can also avoid the tool and the forearm link colliding with the bin. When a bin is too deep or a tool is too short, objects in certain positions will not be able to pick. But this problem can be avoided in the configuration of mechanism. Then, the collision detection can be further simplified as the calculation of the distance from the wrist joint or TCP to the bin, and the threshold of the distance is the radius of spheres.

The internal size of the bin used in this study is 585*385*300 (W*D*H mm), and the bin's relative position to the robot manipulator is fixed. Therefore, the origin of the working coordinate system O_w is set at a corner of the bin, and the x, y axis parallel are set to the two sides of the bin. As shown in Fig. 12, the planes s1, s2, s3, and s4 are the four sides of the bin. In this way, the position of the bin in the workspace can be easily determined, and the working coordinate system can be set at the same time. The system can directly calculate the TCP of the robot manipulator and the shortest distance between the wrist joint and the four sides of the bin. If the distance is less than the preset safety threshold, the system will consider it as a collision. In this paper, the safety threshold of the TCP is set as the radius of the smallest enclosing circle, which depends on the suction cup of the vacuum tool. The safety threshold of the wrist joint is the radius of the wrist joint (about 10 cm) plus a safety distance of 2 cm. When either one of the distances is less than the preset safety threshold, the system will perform collision avoidance to calculate the safe picking pose.

2) COLLISION AVOIDANCE

When a collision is detected, the system needs to adjust the pose of the arm to avoid the collision and ensure the safety of task execution. In this paper, we discuss the collision avoidance of the TCP and the collision of wrist joint separately. At the TCP, if the position of target picking pose collides with the bin, the x or y value will be adjusted directly into the safe area. In the part of the wrist joint, the position

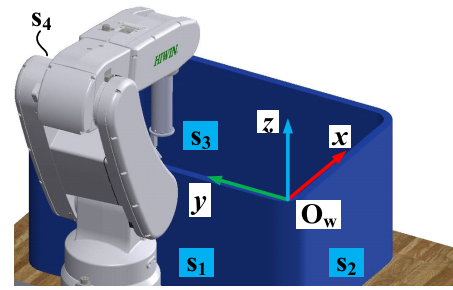


FIGURE 12. Schematic diagram of the location of bin and working coordinate system.

of the wrist joint is changed by adjusting the orientation of the tool, which keeps the position of the target picking pose unchanged. The adjustment of the tool orientation is achieved by rotating along the 'x-axis' or 'y-axis' of the working coordinate system. For example, if the wrist joint collides with the four sides of the bin, s1, s2, s3, and s4, the tool needs to rotate along the 'y-axis', '-x-axis', '-y-axis', and 'x-axis' directions of the working coordinate system, respectively. For example, Fig. 11 shows the difference before and after the pose adjustment, while the position of TCP remains the same. Figs. 11(a) and 11(b), respectively, show the wrist is on a collision situation and the pose of the tool after rotating along the 'y-axis' of the working coordinate system. When the pose adjustment makes the wrist move towards the center of the bin, we can see that it makes the wrist from a collision situation to a collision-free situation. Such adjustment will make the wrist simplified sphere adjust to a safe situation. Furthermore, because the bin is wider than the wrist sphere, the collision between the sphere and the bin can be avoided after the pose adjustment. Among them, the picking pose ${}^{pick}T_t^w$ after adjusting the orientation of the tool by T_{t1}^t is given by

$${}^{pick}T_t^w = {}^{tar}T_t^w T_{t1}^t \quad (2)$$

where

$$T_{t1}^t = \begin{bmatrix} R & 0 \\ 0 & 1 \end{bmatrix}$$

where ${}^{tar}T_t^w$ is the transformation matrix of the TCP in the working coordinate system, that is, the transformation matrix of the requested target picking pose. R is the rotation matrix of the tool rotates θ degree along a vector. Taking the rotation along the x-axis of the working coordinate system with θ degree as an example, the rotation matrix R is given by (3), as shown at the bottom of the next page. It is calculated by the conversion formula from the axis-angle representation to the rotation matrix representation [40]. Among them, the $c\theta$ and $s\theta$ denote $\cos\theta$ and $\sin\theta$, respectively. (n_x, n_y, n_z) is the vector of the x-axis vector of transformation $(T_t^w)^{-1}$, which is described by

$$(T_t^w)^{-1} = \begin{bmatrix} n_x & O_{x_x} & a_x & W_x \\ n_y & O_{y_y} & a_y & W_y \\ n_z & O_{z_z} & a_z & W_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4)$$

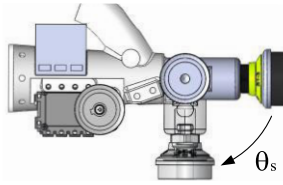


FIGURE 13. Schematic diagram of a vacuum tool with 1-DOF.

Among them, the vectors n , o , a , w represent the x , y , and z axis and origin of the working coordinate system in the tool coordinate system, respectively. Each rotation angle θ is defined as 5 degrees in this paper. After completing the adjustment of the orientation of the tool, the collision detection will be performed again. If a collision still occurs, the orientation adjustment will be carried out again. This service repeats the iteration until there is no collision, and outputs the final result as the picking pose.

G. PICK-AND-PLACE STRATEGY NODE

The pick-and-place strategy node is used to calculate the angle of suction cup, the picking pose, and the placement pose when the robot manipulator performs bin pick-and-place tasks. Due to the consideration of the collision between the robot manipulator and the bin, the picking pose of the robot manipulator during the picking of the object in the bin is limited. Therefore, for the task of orientation requirement, it is necessary to calculate the corresponding placement pose to place the object. In this study, a vacuum tool with 1-DOF is designed to solve this problem. As shown in Fig. 13, the suction cup at the end of the tool can be changed from 0 to 90 degrees, so there is one more degree of freedom: suction cup's angle θ_s at the output to effectively complete the pick-and-place task of the object. For orientation non-requirement task, the angle of the vacuum tool is maintained at 0 degrees, and the picking and placement pose only needs to calculate the position and does not need to calculate the orientation.

In the orientation requirement task, we first obtain the target pose of the object $^{tar}T_o^w$ in the working coordinate system through the coordinate service, then the target picking pose $^{tar}T_t^w$ is calculated by

$$^{tar}T_t^w = ^{tar}T_o^w (T_s^t)^{-1} \tag{5}$$

where T_s^t is the transformation matrix when the angle θ_s is 0 degree. After obtaining the target picking pose $^{tar}T_t^w$, it is necessary to confirm whether the pose will collide. Therefore, the collision service is used to obtain the safe picking pose $^{pick}T_t^w$. After obtaining the picking pose, the correlation

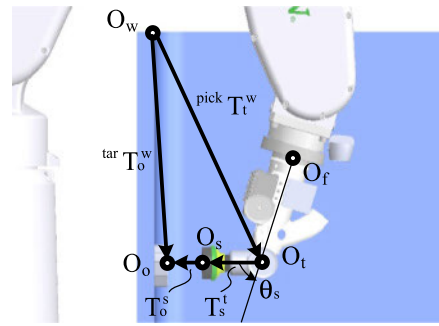


FIGURE 14. Correlation diagram of the angle of suction cup and the pose of robot manipulator.

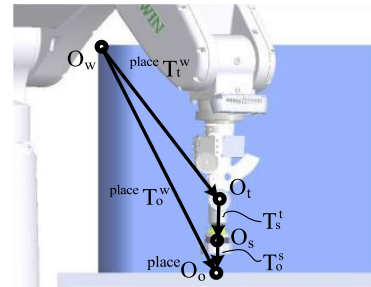


FIGURE 15. Correlation diagram of each coordinate system and the placement pose.

between the angle of suction cup and robot manipulator is shown in Fig. 14, where O_s is the origin of the suction cup coordinate system. Next, the calculation of θ_s can be expressed as

$$\theta_s = \cos^{-1} \left(\frac{|\vec{O_f O_t} \cdot \vec{O_t O_o}|}{|\vec{O_f O_t}| |\vec{O_t O_o}|} \right) \tag{6}$$

Among them, O_f and O_t are the origin of flange and tool coordinate system of the picking pose obtained through the collision service. After the bin collision avoidance adjustment, the picking pose and the target picking pose may be different. In order to place the object in a specific pose, it is necessary to calculate the transformation T_o^s between the suction cup and the object when the object is picked. The transformation can be expressed as

$$T_o^s = (T_s^t)^{-1} (^{pick}T_t^w)^{-1} ^{tar}T_o^w \tag{7}$$

Finally, the placement pose of the robot manipulator to place the object is calculated. The correlation diagram of each coordinate system when placing the object is shown in Fig. 15, where $^{place}T_o^w$ and $^{place}T_t^w$ are the placement

$$R = \begin{bmatrix} c\theta & 0 & 0 \\ 0 & c\theta & 0 \\ 0 & 0 & c\theta \end{bmatrix} + s\theta \begin{bmatrix} 0 & -n_z & n_y \\ n_z & 0 & -n_x \\ -n_y & n_x & 0 \end{bmatrix} + (1 - c\theta) \begin{bmatrix} n_x^2 & n_x n_y & n_x n_z \\ n_y n_x & n_y^2 & n_y n_z \\ n_z n_x & n_z n_y & n_z^2 \end{bmatrix} \tag{3}$$

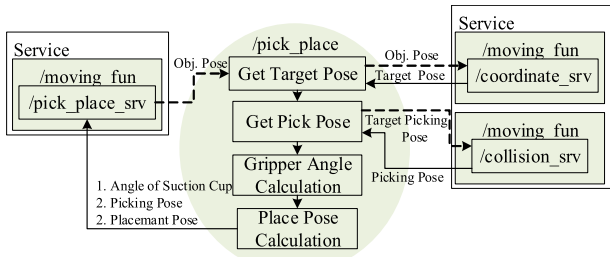


FIGURE 16. Computation graph of the proposed pick-and-place strategy node.

TABLE 1. Summary of the four experiments.

EXP	Task Type	Object Type	Number of Experiments	End Effector
A	Orientation non-required	Textured objects (Sauce packets)	7 times a total of 210 packets	1-DOF vacuum tool
B	Orientation non-required	Texture-less objects (Pushpins)	5 times a total of 200 pushpins	Small nozzle vacuum tool
C	Orientation required	Textured objects (Cracker boxes)	20 times a total of 200 boxes	1-DOF vacuum tool
D	Orientation required	Texture-less objects (Metal workpieces)	20 times a total of 200 objects	1-DOF vacuum tool



FIGURE 17. Two self-made vacuum tools used in the experiment: (a) 1-DOF vacuum tool, (b) small nozzle vacuum tool.

pose of the object and the robot manipulator, respectively. ${}^{place}O_o$ is the coordinate origin of the specific pose of the object. According to ${}^{place}T_o^w$ and T_s^o , ${}^{place}T_t^w$ can be expressed as

$${}^{place}T_t^w = {}^{place}T_o^w (T_s^o)^{-1} (T_t^o)^{-1} \quad (8)$$

Based on the above discussion, the computation graph of the proposed pick-and-place strategy node is shown in Fig. 16. The processing steps are as follows: (a) obtain a target pose ${}^{tar}T_o^w$ of the object through the coordinate service, (b) obtain a picking pose ${}^{pick}T_t^w$ through the collision service, (c) use (6) to calculate an angle θ_s (d) use (8) to calculate a placement pose ${}^{place}T_t^w$ and (e) generate a response of the pick-and-place service θ_s , ${}^{pick}T_t^w$ and ${}^{place}T_t^w$.

IV. EXPERIMENTAL RESULTS

In the experimental results, four practical test results are presented to verify the effectiveness of the proposed bin pick-and-place system. Two self-made vacuum tools used in the experiments are shown in Fig. 17. Four experiments named A, B, C, and D are shown in Table 1 based on two task types (orientation non-required or required) and two object types (textured or texture-less). Among them, since

TABLE 2. Training information of YOLOV4 for the four experiments.

Exp.	Data	Training Data	Testing Data	Learning Rate	Input Size	Classes
A	580	522	58	0.001	608×608	5
B	50	45	5			1
C	50	45	5			6
D	235	211	24			2

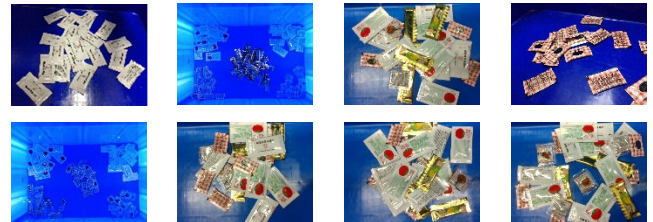


FIGURE 18. Some image data used to train YOLOv4 in experiment A.

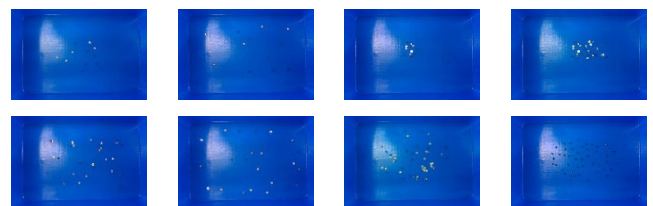


FIGURE 19. Some image data used to train YOLOv4 in experiment B.

the pushpins used in Experiment B are difficult to vacuum with a general vacuum tool, a special vacuum tool that has a small nozzle is designed for vacuuming pushpins effectively. In the proposed system, we used the object detection node to detect objects in the bin. Different types of textured and texture-less objects can be detected, such as three types of sauce packets, pushpins, three types of boxes including front and back, as well as the front and back side of the metal workpiece. In object perception, some training information is described in Table 2, and some image data used to train YOLOv4 in the four experiments are shown in Figs. 18-21. The object detection results of the four experiments are shown in Fig. 22. The object pose estimation results of two orientation required experiments are shown in Fig. 23, where the represented coordinate axes are the object pose obtained by the pose estimation node. In the actual test, the accuracy of object detection and pose estimation, as well as the success rate of the pick-and-place task are counted by executing the task multiple times. The four experiments are described as follows:

A. TEXTURED OBJECTS WITH ORIENTATION NON-REQUIREMENT TASK

The task of this experiment requires the robot to pick up some of the sauce packets that are randomly stacked in the bin. The objects used in Experiment A are three kinds of sauce packets (ketchup, salted egg sauce, pepper). The sizes of the three kinds of packets and the state in the box, as well as the



FIGURE 20. Some image data used to train YOLOv4 in experiment C.

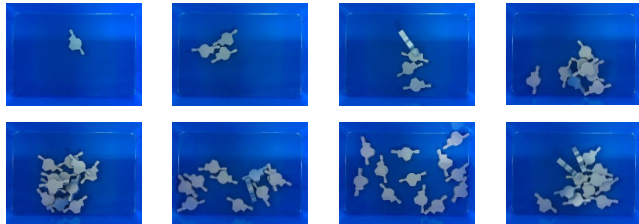


FIGURE 21. Some image data used to train YOLOv4 in experiment D.

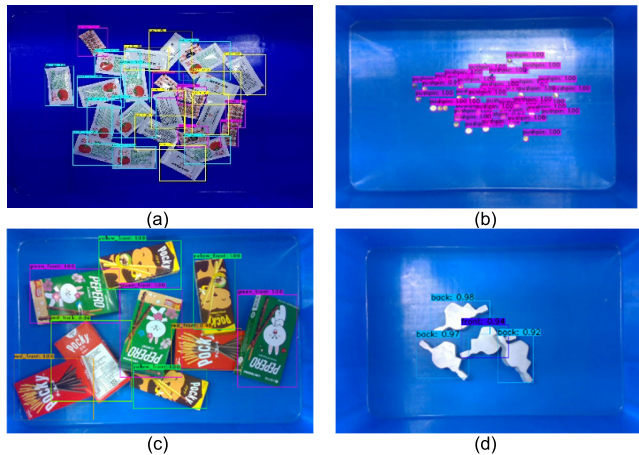


FIGURE 22. Object detection results of four experiments: (a) Experiment A, (b) Experiment B, (c) experiment C, and (d) Experiment D.



FIGURE 23. Object pose estimation results of two orientation required experiments: (a) Experiment C, and (b) Experiment D.

target state of the task are shown in Fig. 24. In the pick-and-place experiment, the main steps of the experiment process are shown in Fig. 25. These states are described as follows: (a) move the arm to the position where the camera takes pictures, (b) pick the target object, and (c) place the object into the assigned grid. The video of this experiment is available on the website: <https://youtu.be/n5ASKhCI24Y/>.



FIGURE 24. Objects used in experiment A.

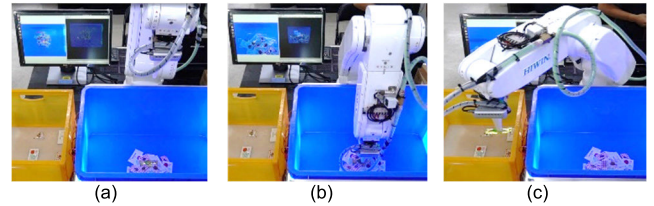


FIGURE 25. Practical test of the sauce packets pick-and-place task.



FIGURE 26. Objects used in experiment B.

In Experiment A, 7 times bin pick-and-place tasks with a total of 210 sauce packets were performed to count the accuracy rate of object detection and the success rate of pick-and-place task. With the proposed bin pick-and-place system, the correct detection rates of the 3 kinds of sauce packets are 97.1%, 85.7%, and 100%, and the success rates of pick-and-place tasks are 97.1%, 84.3%, and 100% respectively. In addition, the bounding box of a sauce packet is easily affected by other sauce packets because of stacking each other, which leads to the situation that the arm may accidentally pick other sauce packets. From the statistical data, it can be found that part of the sauce packet is correctly detected. However, the success rate of pick-and-place tasks is lower than the correct rate of object detection due to picking failures.

B. TEXTURE-LESS OBJECTS WITH ORIENTATION NON-REQUIREMENT TASK

The task of this experiment requires the robot to pick up some metal pushpins randomly stacked in the bin, and only pick and place one of them at a time. The metal pushpins are texture-less objects. The sizes of the metal pushpin, the initial state in the bin, and the target state of the task are shown in Fig. 26. The main steps of the experiment process are shown in Fig. 27. These states are described as follows: (a) move the arm to the position where the camera takes pictures, (b) pick up the target object, (c) place the object. The video of this experiment is available on the website: <https://youtu.be/YDtaj-DCmL8/>.

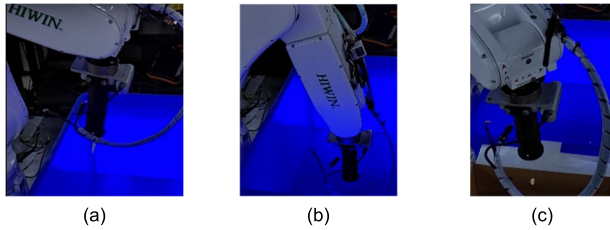


FIGURE 27. Practical test of pushpins bin pick-and-place task.



FIGURE 28. Objects used in experiment C.

In Experiment B, 5 times bin pick-and-place tasks with a total of 200 pushpins were performed to count the accuracy of object detection and the success rate of pick-and-place task. With the proposed bin pick-and-place system, the accuracy rate of pushpin object detection is 100%, and the success rate of pick-and-place tasks is 90.5%. The difficulty of the mental pushpin in Experiment B is that the pushpin lies in the easy rolling characteristics, and because the small nozzle weakens the airflow. It may be touched by the edge of the nozzle and made it roll which led to the failure. Therefore, it can be found from the statistical data that although some of the pushpins are correctly detected, the success rate of the pick-and-place task is lower than the correct rate of the object detection due to the rolling situation.

C. TEXTURED OBJECTS WITH ORIENTATION REQUIREMENT TASK

The task of this experiment requires the robot to pick up some randomly stacked cracker boxes from the bin, and place the objects neatly on the table. The objects used in Experiment C are three kinds of cracker boxes. The appearance size, the state in the bin, and the target state of the task are shown in Fig. 28. The task of picking and placing the cracker box in Experiment C is the textured objects with orientation requirement task, and the main steps of the experiment process are as shown in Fig. 29. These states are described as follows: (a) move the arm to the position where the camera takes pictures, (b) pick the target object, and (c) place the object in the specified pose. The video of this experiment is available on the website: <https://youtu.be/EGaGz50G1fQ/>.

In Experiment C, 20 times bin pick-and-place tasks with a total of 200 cracker boxes were performed to count the accuracy of object detection and pose estimation, and the success rate of the pick-and-place task. With the proposed bin pick-and-place system, the accuracy rate of the object detection of the 3 kinds of cracker box is 96.6%, 100%, and 100%, the accuracy rate of pose estimation is 96.6%, 96.3%, and 93.3%, and the success rate of pick-and-place task is

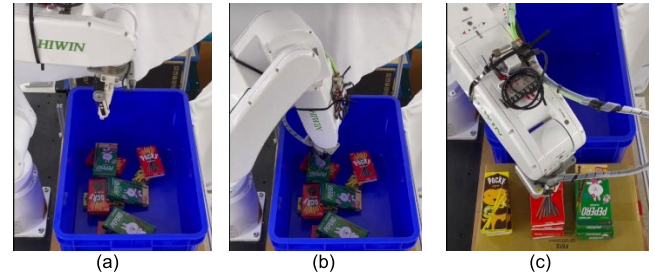


FIGURE 29. Practical test of cracker boxes bin pick-and-place task.



FIGURE 30. Objects used in experiment D.

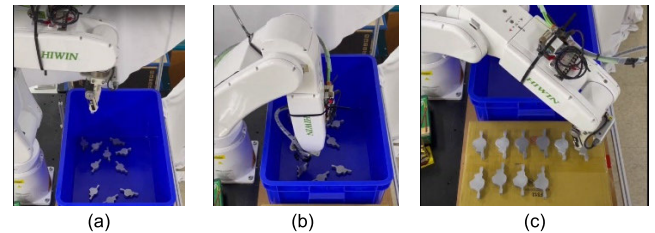


FIGURE 31. Practical test of texture-less object bin pick-and-place task.

93.3%, 96.3%, and 93.3%. In the pose estimation part, if the point cloud of the entire object is used for pose estimation, the average accuracy rate was only 74%, which shows the effectiveness of the pose estimation method proposed in this paper.

D. TEXTURE-LESS OBJECTS WITH ORIENTATION REQUIREMENT TASK

The task of this experiment requires the robot to pick up some randomly stacked objects from the bin, and place the objects neatly on the table. The objects used in Experiment D are metal workpieces, which are texture-less objects with similar front and back sides. The appearance size, the state in the bin, and the target state of the task are shown Fig. 30. The main steps of the experiment process are shown in Fig. 31. These states are described as follows: (a) move the arm to the position where the camera takes pictures, (b) pick the target object, and (c) place the object in the specified pose. The video of this experiment is available on the website: <https://youtu.be/uniHtmOMGNY/>.

In Experiment D, 20 times bin pick-and-place tasks with a total of 200 metal workpiece were performed to count the accuracy of object detection and pose estimation, and the success rate of the pick-and-place task. With the proposed bin pick-and-place system, the accuracy rate of object detection is 100%, the accuracy rate of pose estimation is 89.5%, and

TABLE 3. Experimental data Statistics.

Experiment	Experimental Object	Object Detection	Pose Estimation		Collision Avoid.	Pick& place
			5°	15°		
A	Sauce 1	97.1%	/	/	100%	97.1%
	Sauce 2	85.7%				84.3%
	Sauce 3	100%				100%
B	Pushpin	100%	/	/	100%	90.5%
C	Box 1	96.6%	5°	88.3%	100%	93.3%
			15°	96.6%		
	Box 2	100%	5°	88.8%		96.3%
			15°	96.3%		
	Box 3	100%	5°	86.6%		93.3%
			15°	93.3%		
D	Metal Workpiece	100%	5°	87.5%	100%	89.5%
			15°	89.5%		

the success rate of pick-and-place tasks is 89.5%. In the pose estimation part, if the source point cloud of the entire object is used for pose estimation, the correct rate is only 67%, which shows the effectiveness of the proposed pose estimation method.

The experimental data of the four experiments are shown in Table 3. Among them, the column of object detection records the detection accuracy of each object class, and the column of pose estimation records the probability of each object class that the angle error of the placed object is within 5° and 15°. The column of collision avoidance records the success rate of the robot manipulator not colliding with the bin during each pick-and-place task. The last column records the success rate of the pick-and-place task where the object is successfully picked and placed in the correct position or pose. For the object detection results of the four experiments, we observed that the detection accuracy of most of the experimental objects is 100%. Some false detection of Sauce 1, Sauce 2, and Box1 may be caused by unclear image features due to light reflections on the packaging. For the pose estimation of Experiments C and D, the average probability of angle error within 15° is 93.925%. Most of the estimation errors are due to the fact that the point cloud data obtained by the RealSense D435 camera contains noise. Furthermore, since the size of the metal workpiece is smaller than that of the biscuit box, its pose estimation becomes more difficult. For the collision avoidance of the four experiments, the collision between the wrist joint and the TCP were 100% avoided in each experimental task during the pick-and-place operation. In the four experiments, the average pick-and-place success rate of pick-and-place reached 93.04%. Some errors are caused by false detections or object dropped during picking.

V. CONCLUSION

We implemented a ROS-based bin pick-and-place system, where ROS is used to integrate RGB-D camera, robot manipulator, vacuum tool, object perception module and object pick-and-place module. Through the ROS framework,

the function of each node can be more simplistic and modular to make the process of system development more flexible and efficient. Also, we provided a complete bin pick-and-place system framework that can be applied to a variety of bin pick-and-place scenarios, such as textured or texture-less objects, pose requirement or not, different tools and different cameras, and each module can be easily changed to suit a specific application. For the implementation of each module, in the object detection part, we used the YOLOv4 to detect various types of objects, and obtained high detection accuracy. In the object sorting part, the uniform sampling is used to find the position of the object in the bin, which can accurately select the highest object. In the pose estimation part, a CAD-based pose estimation method that splits the object model into two parts is adopted, which can effectively improve the accuracy of the pose estimation. In the coordinate transformation part, the pose of the object in the camera coordinate system is accurately transform into the working coordinate system, so that the robot manipulator can accurately pick small objects such as pushpins. In the bin collision avoidance part, the proposed link-distance based method can effectively prevent the robot manipulator from colliding with the bin, so that it can safely perform the bin pick-and-place task. In the pick-and-place strategy part, by calculating the angle of suction cup and the placement pose, the robot manipulator can accurately place the object in the specified pose. Some experimental results show that the proposed bin pick-and-place system achieves good performance in four experiments with two task types and two object types.

REFERENCES

- [1] M. Fujita, Y. Domae, A. Noda, G. A. Garcia Ricardez, T. Nagatani, A. Zeng, S. Song, A. Rodriguez, A. Causo, I. M. Chen, and T. Ogasawara, "What are the important technologies for bin picking? Technology analysis of robots in competitions based on a set of performance metrics," *Adv. Robot.*, vol. 34, nos. 7–8, pp. 560–574, 2020.
- [2] W. Budiharto, "Robust vision-based detection and grasping object for manipulator using SIFT keypoint detector," in *Proc. Int. Conf. Adv. Mech. Syst.*, Kumamoto, Japan, Aug. 2014, pp. 448–452.
- [3] V. Vijayan and P. Kp, "FLANN based matching with SIFT descriptors for drowsy features extraction," in *Proc. 5th Int. Conf. Image Inf. Process. (ICIIP)*, Shimla, India, Nov. 2019, pp. 600–605.
- [4] Z. S. Ni, "B-SIFT: A binary SIFT based local image feature descriptor," in *Proc. 4th Int. Conf. Digital Home*, Guangzhou, China, Nov. 2012, pp. 117–121.
- [5] A. Blank, M. Hiller, S. Zhang, A. Leser, M. Metzner, M. Lieret, J. Thielecke, and J. Franke, "6DoF pose-estimation pipeline for texture-less industrial components in bin picking applications," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Prague, Czech Republic, Sep. 2019, pp. 1–7.
- [6] L. Tian, N. M. Thalmann, D. Thalmann, Z. Fang, and J. Zheng, "Object grasping of humanoid robot based on YOLO," in *Proc. Comput. Graph. Int. Conf.*, 2019, pp. 476–482.
- [7] M. Schwarz and S. Behnke, "Data-efficient deep learning for RGB-D object perception in cluttered bin picking," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, Singapore, May 2017, pp. 2–4.
- [8] J. Lee, S. Kang, and S.-Y. Park, "3D pose estimation of bin picking object using deep learning and 3D matching," in *Proc. 15th Int. Conf. Informat. Control, Autom. Robot.*, 2018, pp. 318–324.
- [9] Z. Pan, Z. Jia, K. Jing, Y. Ding, and Q. Liang, "Manipulator package sorting and placing system based on computer vision," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Hefei, China, Aug. 2020, pp. 409–414.
- [10] K. Du, J. Song, X. Wang, X. Li, and J. Lin, "A multi-object grasping detection based on the improvement of YOLOv3 algorithm," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Hefei, China, Aug. 2020, pp. 1027–1033.

- [11] P. K. Chemelil, J. G. Njiri, and J. K. Kimotho, "Real time object detection using single shot multibox detector network for autonomous robotic arm," *J. Sustain. Res. Eng.*, vol. 6, no. 1, pp. 11–23, 2020.
- [12] T. T. Le and C. Y. Lin, "Bin-picking for planar objects based on a deep learning network: A case study of USB packs," *Sensors*, vol. 19, no. 16, pp. 3602–3632, 2019.
- [13] K. T. Song, C. H. Wu, and S. Y. Jiang, "CAD-based pose estimation design for random bin picking using a RGB-D camera," *J. Intell. Robot. Syst.*, vol. 87, nos. 3–4, pp. 455–470, 2017.
- [14] W. Yan, Z. Xu, X. Zhou, Q. Su, S. Li, and H. Wu, "Fast object pose estimation using adaptive threshold for bin-picking," *IEEE Access*, vol. 8, pp. 63055–63064, 2020.
- [15] D. Li, N. Liu, Y. Guo, X. Wang, and J. Xu, "3D object recognition and pose estimation for random bin-picking using partition viewpoint feature histograms," *Pattern Recognit. Lett.*, vol. 128, pp. 148–154, Dec. 2019.
- [16] J.-H. Bae, H. Jo, D.-W. Kim, and J.-B. Song, "Grasping system for industrial application using point cloud-based clustering," in *Proc. 20th Int. Conf. Control, Autom. Syst. (ICCAS)*, Busan, South Korea, Oct. 2020, pp. 608–611.
- [17] C. Yang, Z. Li, Z. Cai, Y. Gao, T. Xu, G. He, F. Yan, and C. Shao, "Target position and posture recognition based on RGB-D images for autonomous grasping robot arm manipulation," in *Proc. 10th Int. Conf. Inf. Sci. Technol. (ICIST)*, London, U.K., Sep. 2020, pp. 65–70.
- [18] H.-I. Lin and M. N. Cong, "Inference of 6-DOF robot grasps using point cloud data," in *Proc. 19th Int. Conf. Control, Autom. Syst. (ICCAS)*, Jeju, South Korea, Oct. 2019, pp. 944–948.
- [19] K.-T. Song, Y.-H. Chang, and J.-H. Chen, "3D vision for object grasp and obstacle avoidance of a collaborative robot," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Hong Kong, Jul. 2019, pp. 254–258.
- [20] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Proc. 1st Annu. Conf. Robot Learn. (PMLR)*, vol. 78, 2017, pp. 515–524.
- [21] A. Zeng, S. Song, and K.-T. Yu, "Robotic pick-and-place of novel objects in clutter with multi-affordance grasping and cross-domain image matching," in *Proc. IEEE Int. Conf. Robot. Automat., Brisbane, QLD, Australia, Aug. 2018*, pp. 3750–3757.
- [22] R. Matsumura, K. Harada, Y. Domae, and W. Wan, "Learning based industrial bin-picking trained with approximate physics simulator," in *Proc. Int. Conf. Intell. Auton. Syst.*, Dec. 2018, pp. 786–798.
- [23] Y. Xu, L. Wang, A. Yang, and L. Chen, "GraspCNN: Real-time grasp detection using a new oriented diameter circle representation," *IEEE Access*, vol. 7, pp. 159322–159331, 2019.
- [24] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *Int. J. Robot. Res.*, vol. 36, nos. 13–14, pp. 1455–1473, Dec. 2017.
- [25] H. Liang, X. Ma, S. Li, M. Gornier, S. Tang, B. Fang, F. Sun, and J. Zhang, "PointNetGPD: Detecting grasp configurations from point sets," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, Montreal, QC, Canada, May 2019, pp. 3629–3635.
- [26] M. Wei and P. Xiwei, "WLIB-SIFT: A distinctive local image feature descriptor," in *Proc. IEEE 2nd Int. Conf. Inf. Commun. Signal Process. (ICICSP)*, Weihai, China, Sep. 2019, pp. 379–383.
- [27] B. Zhong and Y. Li, "Image feature point matching based on improved SIFT algorithm," in *Proc. IEEE 4th Int. Conf. Image, Vis. Comput. (ICIVC)*, Xiamen, China, Jul. 2019, pp. 489–493.
- [28] K. Wang and S. Liu, "RGBD object recognition and flat area analysis method for manipulator grasping," in *Proc. IEEE 9th Data Driven Control Learn. Syst. Conf. (DDCLS)*, Liuzhou, China, Nov. 2020, pp. 729–734.
- [29] C. Kaymak and A. Ucar, "Implementation of object detection and recognition algorithms on a robotic arm platform using raspberry pi," in *Proc. Int. Conf. Artif. Intell. Data Process. (IDAP)*, Malatya, Turkey, Sep. 2018, pp. 1–8.
- [30] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," 2019, *arXiv:1909.04810*.
- [31] S. Hernandez-Mendez, C. Maldonado-Mendez, A. Marin-Hernandez, H. V. Rios-Figueroa, H. Vazquez-Leal, and E. R. Palacios-Hernandez, "Design and implementation of a robotic arm using ROS and MoveIt!" in *Proc. IEEE Int. Autumn Meeting Power, Electron. Comput. (ROPEC)*, Nov. 2017, pp. 1–6.
- [32] T. Wang, Y. Zhao, L. Zhu, G. Liu, Z. Ma, and J. Zheng, "Design of robot system for radioactive source detection based on ROS," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2020, pp. 1397–1400.
- [33] C.-C. Wong, S.-Y. Chien, H.-M. Feng, and H. Aoyama, "Motion planning for dual-arm robot based on soft actor-critic," *IEEE Access*, vol. 9, pp. 26871–26885, 2021.
- [34] C.-C. Wong, L.-Y. Yeh, C.-C. Liu, C.-Y. Tsai, and H. Aoyama, "Manipulation planning for object re-orientation based on semantic segmentation keypoint detection," *Sensors*, vol. 21, no. 7, p. 2280, Mar. 2021.
- [35] C.-C. Wong, M.-Y. Chien, R.-J. Chen, H. Aoyama, and K.-Y. Wong, "Moving object prediction and grasping system of robot manipulator," *IEEE Access*, vol. 10, pp. 20159–20172, 2022.
- [36] Y. Zhang, L. Li, M. Ripperger, J. Nicho, M. Veeraraghavan, and A. Fumagalli, "Gilbreth: A conveyor-belt based pick- and-sort industrial robotics application," in *Proc. 2nd IEEE Int. Conf. Robotic Comput. (IRC)*, Jan. 2018, pp. 17–24.
- [37] P. Tavares and A. Sousa, "Flexible pick and place architecture using ROS framework," in *Proc. 10th Iberian Conf. Inf. Syst. Technol. (CISTI)*, Jun. 2015, pp. 1–6.
- [38] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (FPFH) for 3D registration," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2009, pp. 3212–3217.
- [39] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.
- [40] C. J. Taylor and D. J. Kriegman, "Minimization on the lie group SO(3) and related manifolds," Dept. Elect. Eng., Center Syst. Sci., Yale Univ., New Haven, CT, USA, Tech. Rep. 9405, Apr. 1994.



CHING-CHANG WONG received the B.S. degree from the Department of Electronic Engineering, Tamkang University (TKU), Taiwan, in 1984, and the M.S. and Ph.D. degrees from the Department of Electrical Engineering, Tatung Institute of Technology, Taiwan, in 1986 and 1989, respectively. In 1989, he joined the Department of Electrical and Computer Engineering, TKU, where he worked as the Department Chair, from 2006 to 2010. In 2007, he established the Robotics Engineering Institute. In 2011, he established the Intelligent Automation and Robotics Center. He is currently a Distinguished Professor. He has published and coauthored over 300 technical articles and 20 patents. His current research interests include intelligent robot, mobile robot manipulator, and deep reinforcement learning for robotic applications. He was elevated as a fellow of the Institution of Engineering and Technology (IET), the Chinese Automatic Control Society (CACS), and the Robotics Society of Taiwan (RST), in 2009, 2015, and 2019, respectively. He was a recipient of the Outstanding Automatic Control Award from the CACS, Taiwan, in 2009, and the Outstanding Robotics Engineering Award from the RST in 2018. He received the Outstanding and Premium Research Awards (2011–2021) from the Ministry of Science and Technology (MOST), Taiwan. From 2009 to 2010, he worked as the Chair of the IEEE Robotics and Automation Taipei Chapter.



CHI-YI TSAI (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from the National Yunlin University of Science and Technology, Yunlin, Taiwan, in 2000 and 2002, respectively, and the Ph.D. degree in electrical and control engineering from the National Chiao Tung University, Hsinchu, Taiwan, in 2008. He joined the Department of Electrical Engineering, Tamkang University, New Taipei City, Taiwan, in 2010, where he is currently a Professor.

His research interests include image processing, color image enhancement processing, visual tracking, visual servoing, computer vision, and deep learning.



REN-JIE CHEN was born in Taoyuan, Taiwan, in 1993. He received the B.S. and M.S. degrees from the Department of Electrical and Computer Engineering, Tamkang University, Taiwan, in 2015 and 2017, respectively. He is currently pursuing the Ph.D. degree with the Doctoral Program in Robotics, College of Engineering. He participated in the Short Exchange Student Program, University of Electro-Communications (UEC), Tokyo, Japan, from November to December 2019.

His research interests include robot manipulator, robotic applications, reinforcement learning, and machine learning.



SHAO-YU CHIEN was born in Keelung, Taiwan, in 1995. He received the B.S. and M.S. degrees from the Department of Electrical and Computer Engineering, Tamkang University, Taiwan, in 2017 and 2019, respectively. He is currently pursuing the Ph.D. degree with the Doctoral Program in Robotics, College of Engineering. His research interests include robot manipulator, robotic applications, and machine learning.



YI-HE YANG was born in Taipei, Taiwan, in 1996. He received the B.S. and master's degrees from the Department of Electrical and Computer Engineering, Tamkang University, Taiwan, in 2018. He is currently pursuing the Ph.D. degree with the Doctoral Program in Robotics, College of Engineering. His research interests include robot manipulator, robotic applications, and machine learning.



SHANG-WEN WONG was born in Kaohsiung, Taiwan, in 1991. She received the B.S. degree in horticulture and landscape architecture from the National Taiwan University, in 2013, and the M.S. degree in electrical engineering from the National Tsing Hua University, in 2017. She is currently pursuing the Ph.D. degree with the Doctoral Program in Robotics, College of Engineering. She participated in the Exchange Student Program, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, from August to December 2016. She is also a Teaching Assistant with the Department of Electrical and Computer Engineering, Tamkang University. Her research interests include machine learning, deep learning, robot vision, robotic application, and robotic grasping.



CHUN-AN YEH was born in New Taipei, Taiwan, in 1996. He received the Bachelor of Science degree from the Department of Electronic Engineering, Longhua University of Science and Technology, Taiwan, in 2019, and the master's degree from the Department of Electrical and Computer Engineering, Tamkang University, Taiwan. He is currently pursuing the Ph.D. degree with the Doctoral Program in Robotics, College of Engineering. His main research interests include robotic arms, applications of robotic arms, deep learning, machine vision, and machine learning.

...