

Received February 12, 2022, accepted February 17, 2022, date of publication February 21, 2022, date of current version March 8, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3153071

Optimal Grasping Strategy for Robots With a Parallel Gripper Based on Feature Sensing of 3D Object Model

HSIN-HAN CHIANG¹, (Senior Member, IEEE), JIUN-KAI YOU²,
CHEN-CHIEN JAMES HSU^{1,2}, (Senior Member, IEEE), AND JUN JO³

¹Department of Vehicle Engineering, National Taipei University of Technology, Taipei City 106, Taiwan

²Department of Electrical Engineering, National Taiwan Normal University, Taipei City 106, Taiwan

³School of Information and Communication Technology, Griffith University, Southport, QLD 4222, Australia

Corresponding author: Chen-Chien James Hsu (jhsu@ntnu.edu.tw)

This work was supported in part by the Chinese Language and Technology Center of the National Taiwan Normal University (NTNU) through the Featured Areas Research Center Program within the Framework of the Higher Education Sprout Project by the Ministry of Education (MOE), Taiwan; and in part by the Ministry of Science and Technology, Taiwan, through the Pervasive Artificial Intelligence Research (PAIR) Laboratories, under Grant MOST 109-2634-F-003-006 and Grant MOST 109-2634-F-003-007.

ABSTRACT Grasping strategy of many different kinds of target objects is vital in a wide range of automatic robotic manipulations. Among the many potential applications for a robot arm with a parallel gripper, the key challenge for robotic grasping is to determine an optimal grasping pose relative to the target object. Previous works based on 2D grasp and 6-Dof grasp planning have been proposed to efficiently consider the physical contact between the robotic gripper and the object. However, there are still a few unsolved problems caused by partial and limited information about the detected objects due to their locations and geometries that reduce the grasping quality and reliability. In view of these problems, this paper proposes an optimal grasping strategy to deal with target objects with any poses based on their 3D model during the grasping process. Experimental results of the performance evaluation show that the proposed method outperforms the state-of-the-art in terms of grasping success rate on the YCB-Video datasets. Moreover, we further investigate the effectiveness of the proposed method in two scenarios where the robotic manipulator works in either the collaborative or bin-picking modes.

INDEX TERMS Robot grasping, point cloud, parallel gripper, 3D objects, grasp planning.

I. INTRODUCTION

With the advancement of science and technology in recent years, robot arms have been widely used in many practical fields, such as industrial manufacturing, living assistance, agriculture, medical rescue, entertainment services, military security, and even space exploration, etc. According to the International Federation of Robotics (IFR), over the past ten years, more than 2.7 million industrial robots have been introduced and operated worldwide [1]. The top three industries with the highest usage are automobile (30%), electrical and electronic industries (25%), and the metal industry (10%). Most of these traditional industrial robots operate automatically but lack self-awareness and flexibility. They

The associate editor coordinating the review of this manuscript and approving it for publication was Bidyadhar Subudhi.

generally rely on professional engineers to customize control programs to achieve various grasping tasks on the production line for different applications. As a result, most applications are based on low-mix, high-volume production models. According to the study [2], production-grade robot systems can require days or weeks of effort from highly trained robot programmers when the production lines change. This adjustment time is far from satisfying the flexible production needs of the industry, especially for small and medium-sized businesses (SMBs) that generally produce customized products in small batches and short production cycles. There is not sufficient capital or time to carry out sophisticated tuning for robots in the production line. These difficulties have prevented SMBs from adopting robot arms into production lines to achieve Low Volume Automation (LVA). Thus, robotic systems' development has gradually shifted their

focus from efficient single repetitive tasks in a controllable environment to intelligent robots with autonomous visual recognition and smart grasping strategy. As well as achieving LVA in the factory, collaborative robots have also become increasingly popular in recent years. Through human-robot interaction, a collaborative robot can work together with people to accomplish various tasks, like assembling parts, carrying items, inspecting products, oropharyngeal-swab sampling [3], etc., to further improve the working efficiency.

Since robot grasping is the crucial technique for robotic manipulation, it has long been a significant challenge for robotic systems [4], [5]. For various robots, including industrial robots, service robots, and collaborative robots, to accurately complete a task, grasping strategy, which dominates the reliability of automation, is of significant importance. As a result, related technologies have also been extensively studied for decades, particularly in recent years [4], [7]. However, traditional factory work is mainly concerned with specific grasp locations because the robot arm performs regular work that can be pre-defined. Various 6-7 Dof robot arms have been produced and utilized to accomplish different tasks. With greater flexibility in the robot arm, grasp estimation also has been widely researched. Several different methods to solve the various tasks have also emerged. The 2D planar grasp is widely used to solve the task when a target object lies on the plane. In this scenario, the plane's height is fixed, so it only needs to know the position of the object from the camera and the rotation angle which is vertical to the plane. The problem here is that the target object is constrained to lie on a plane and the gripper is constrained from one direction. With the recent advance of deep learning-based approaches, a large number of methods are used to evaluate the candidates of the oriented rectangle whether they are reliable to grasp. Although 2D planner grasp provides a more reliable estimate by using a neural network, it still has several constraints about the placement of the gripper and object under different circumstances. With the progression of the 6DoF grasp, a robot arm can grasp the target from different angles and locations in the 3D domain. Analytical methods were utilized to analyze the geometric structure of the 3D data, and thus the better points according to different grippers could be found. With the use of the 3D cameras with depth sensing such as Microsoft Kinect, Intel Realsense, etc., several methods have been developed to utilize depth information to yield a better grasping capability. Similarly, the application of deep learning is quite popular when searching for a better grasp rectangle from point cloud data. However, several problems still exist with this method since it would be restricted by the angle between objects and RGB-D camera. Another problem is the precision and constraints of the depth camera. Besides estimating the grasp from 3D information, most of the current grasping methods aim at object detection first and then estimating the pose of the object, finally using the predefined grasp pose which is relative to the estimated pose so that the robot arm can successfully grasp the object. As a result, the problems lie

in the accuracy of object pose estimation and the suitability of the predefined grasp pose.

As an attempt to relieve the restricted constraints in the aforementioned research, this paper presents a system that takes advantage of the related methods to deal with the above mentioned problems so that the robotic manipulator can successfully grasp the target object. Thanks to the rapid development of the object pose estimation, we not only can estimate the correct object pose with RGB-D cameras from mature methods such as DenseFusion [8], [9] and PVN3D [10], but also directly estimate object pose with RGB cameras by employing the method such as OPEPL [11]. Compared with several similar works, our method utilizes the object pose to estimate the grasp pose directly. In this way, we propose an effective algorithm rather than using the deep learning method so that an optimal robotic grasp pose can be quickly determined to perform the reliable grasping task in real time. In this strategy, we can find the 6D grasp pose containing the location and angle in 3D space rather than 2D location and one orientation by the traditional 2D grasping planner. Moreover, our proposed grasping strategy can be utilized in different applications such as collaborative robots, production lines, and bin-picking, etc. Also, to achieve the hand-over action in the collaborative mode or production line, we separate the target object into two parts, i.e., grasping area and non-graspable area. Accordingly, the robot arm can find an optimal grasp pose based on the determined grasping area of the target object with any pose for grasping with a conventional parallel gripper. More specifically, this paper makes the following contributions:

- 1) An algorithm-based optimal grasping strategy is proposed to efficiently determine an optimal grasping pose for target objects with any poses based on their 3D model during the grasping process using only a RGB camera.
- 2) The proposed optimal grasping strategy is effective for accomplishing tasks in both collaborative and bin-picking modes to grasp objects with various shapes and sizes.
- 3) The proposed optimal grasping strategy has a better success rate for grasping objects on the YCB-Video dataset than the state-of-the-art methods.
- 4) Experimental results show that the total computation time required to obtain an optimal grasping pose is about 0.14 seconds, which reveals the feasibility to provide real-time operations for practical robot grasping applications.

II. RELATED WORKS

A. 2D GRASPING PLANNER

The grasping strategies in a 2D image plane have been widely used in circumstances where the target object is placed on a plane. In these circumstances, we only need to consider the 2D location and one orientation. Recent commonly used methods can be roughly divided into finding the grasp contact

point and using the oriented rectangle. Firstly, the method of finding the grasp contact point first generates multiple grasp points as candidate points and then evaluates the possibility of successful grasping through analysis or deep learning-based methods. The study in [12] proposed a method of estimating from depth maps, but this empirical method generally encounters difficulties in dealing with unknown objects. With the development of deep learning, [13], [14] proposed a method based on deep learning to solve this problem. First, this method generates several grasp candidates from a depth map and determines the score of each candidate from the network. The one with the highest score will be determined as the final grasp points. Through training on a huge dataset, this method can handle unknown objects to obtain a grasp point.

Secondly, [15] proposed a method using the oriented rectangle to represent the configuration of the gripper, then selecting a good grasp from the candidates of the oriented rectangle. As well as finding candidates from an image directly, [15] estimated the object contour from the depth map to find an oriented rectangle with the skeleton of the object. There are also several methods [17] using deep learning to find the oriented rectangle as candidates to determine the best candidate through another network.

Unfortunately, the methods mentioned above only deal with the situation when the objects are placed on a specific plane. As a result, they suffer from several constraints, including the accuracy of the depth camera, the detection distance of the depth camera, the situation where the object is occluded, and even the position of the depth camera.

B. 6-DOF GRASPING PLANNER

Because of the constraints of the 2D grasping configuration, 6-Dof grasping has received great attention. The difference from the previously mentioned 2D planner grasp is that the purpose of this approach is to generate a valid grasp from the 6D pose space. Through this approach, the robot arm can grasp an object which is placed at any random position and angle and is no longer restricted to objects that must be placed on a fixed plane. The methods of 6Dof grasp can be roughly divided into methods based on partial point cloud and methods based on the complete shape, according to the different information used.

1) EVALUATING THE GRASP QUALITIES OF CANDIDATE GRASPS

This method is widely used by sampling many candidate grasps to find the best grasp quality using various algorithms. Among the traditional methods, [14] proposed a supervised learning approach to find the best grasp pose. With the current rise of deep learning, several methods have been proposed. GPD [18] used ROI (Region of Interest) to find candidate grasps, and evaluated each candidate by a convolutional neural network to find the best grasp candidate which had the highest score. Besides, PointnetGPD [19] randomly samples candidate grasps from GPD [18], and evaluates the grasp quality by direct point cloud analysis with the 3D deep neural

network PointNet. In 2019, GraspNet [20] used a vibrational autoencoder and grasp evaluator model to refine the grasps. Then REGNet [21] proposed a three-stage approach. First, a score network (SN) is used to divide the point cloud into regions with high confidence. Then a grasp region network (GRN) will generate several grasp candidates, and use the refinement network to revise the grasp pose to obtain the final grasp pose. However, to obtain an optimal grasp candidate, these methods based on evaluating the grasp candidates on the partial point cloud need to detect the object and separate the point cloud of the object from the depth map first.

2) TRANSFERRING GRASPS FROM EXISTING ONES

This method focuses on transferring the existing grasp to another, which means finding the correspondences from the target to the existing ones if both are in the same category. [22] proposed a taxonomy-based approach, classifying objects into several categories so that the required grasping pattern for a certain object can be found from the associated categories. [23] also proposed a part-based grasp planning to segment objects into several categories according to their shape and volumetric information. The object parts are also labeled with semantic and grasping information. As a result, the grasp can be transformed from the same category's object. DGCM-Net [24] adopted the network to learn a reliable grasp, and then transferred the grasp to unseen objects in the same category. In order to transfer the grasp, both methods need to detect the object and separate the point cloud of the object from the depth map first. Another method, as proposed in [25], utilized a model-based registration approach for the 6-DoF pose estimation of the target object, and increased the grasping efficiency by reducing the 3D data scanning operations.

III. OPTIMAL GRASPING STRATEGY

In this paper, we propose an optimal grasping strategy utilizing the estimated object pose obtained through an object pose estimation system for two-finger grippers. Figure 1 illustrates the architecture of the proposed grasping process, in which an optimal grasping strategy aims to find an optimal 6Dof grasp pose for a two-finger gripper based on an object pose and preloaded object's point cloud. With this optimal grasping pose, we can solve the problem of the 2D planner grasp which only can grasp objects placed on a fixed plane. In developing the optimal grasping strategy, the complete point cloud of the object is transformed by the object pose so that we would not need to consider the limitation caused by the placement angle, accuracy, or sensor noise of the depth camera. At the same time, we expect to find a grasp pose on a specific area of the object so that the collaborative robot would not grasp the area currently held by the human hand. As far as production line is concerned, we also expect robotic arms can grasp a specific area of the target object for further processing. As a result, we propose this optimal grasping strategy to accomplish these tasks, and Figure 2 shows the flowchart of the optimal



FIGURE 1. Architecture of the proposed grasping process of a robotic manipulation system.



FIGURE 2. Flowchart of the proposed optimal grasping strategy.

grasping strategy. First, the object is transformed by an object pose. Next, we determine the grasping area for the object by human assistance as indicated by a dashed rectangle. Third, the grasping area is segmented into K clusters. Fourth, each cluster will be processed to create M grasping paths. Finally, by calculating the cost function of each grasping path, the system can determine an optimal grasping pose for guiding the manipulator to complete the grasping task.

A. DETERMINE GRASPING AREA

As mentioned earlier, the proposed strategy aims to find the optimal grasp solution. Therefore, the robot arm firstly needs to detect a proper contact area of the object to guarantee a stable grasping. In order to solve this problem, we roughly separate each object into grasping and non-graspable areas. At this stage, we determine a specific area of each object as the desired grasping area suitable for a specific situation, task, or application by human assistance with the aid of Meshlab [26], an open source system for processing and editing 3D triangular meshes. Steps to determine the grasping area of a target object can be summarized as:

- 1) Use “Meshlab” to open the point cloud file of the object.
- 2) Select the non-graspable area roughly through the “Interactive Selection” function in “Meshlab”.
- 3) Delete the selected non-graspable area to keep the desired grasping area
- 4) Save the remaining point cloud as the grasping area.

For example, in human-robot collaboration, we might determine a desired grasping area of a target object according to the human’s attempt to grasp the object. Take the object ‘mug’ for example. Figure 3 shows a complete 3D point cloud of the mug, while Figure 4 shows the determined grasping area of this object in red dots according to a specific task or application.

B. SEGMENT INTO CLUSTERS

After determining the grasping area, we can roughly separate the grasping area from the original point cloud into two areas. However, we want to find more precise areas to fit the size of the gripper. As a result, K -means method [27] is

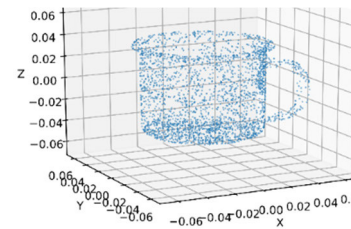


FIGURE 3. Complete 3D point cloud of a mug.

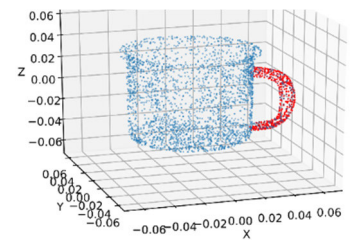


FIGURE 4. Determined grasping area for the mug in Figure 3.

used to segment the point cloud of the grasping area into K clusters. K -means is a method of vector quantization, aiming to partition n observations into K clusters in which each observation belongs to the cluster with the nearest mean. With this method, the point cloud of the grasping area can be easily segmented into K clusters, where K depends on the size of the gripper. A bigger gripper has fewer clusters and a small gripper has more. At the same time, K -means can also find the center of each cluster. As illustrated in Figure 5, the determined grasping area is segmented into 3 clusters painted with different colors, and the corresponding center point is shown by a red point.

C. CREATE GRASPING PATHS

After segmentation, we have K clusters and their centers. To further find the grasp pose, we create M grasping paths in each K cluster. By visualizing every possibility of the grasping paths when we grasp a point or object, we find that there exists a sphere surrounding the grasping target, where the connection from each point on the sphere to the

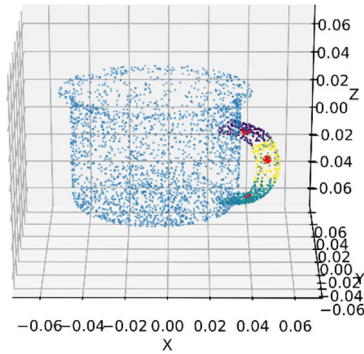


FIGURE 5. Segmentation of the grasping area.

grasping point represents a grasping path. Fibonacci Lattice is a method to create points lying on the surface of the sphere. As a result, we will use the Fibonacci sphere to create M grasping paths, where M depends on how many paths we want to create for grasping paths. The center of the Fibonacci sphere is the center of each cluster. We then create the M grasping paths from the M points scattered on the sphere. In Figure 6, red lines represent all possible grasping paths while setting the Fibonacci sphere on the center of each K cluster.

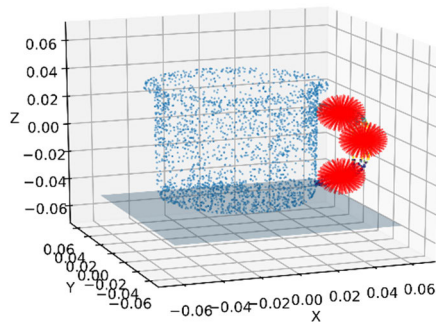


FIGURE 6. M paths created by Fibonacci sphere for each cluster.

D. CALCULATE GRASPING COST FUNCTION

To apply this grasping strategy for human-robot interaction such as collaborative robots or service robots, we expect the gripper to grasp the target as a human does. Through observation of humans, we find that humans are used to grasping an object orthogonally to the main axis of an object. For example, most people are used to taking a PET bottle horizontally. Inspired by this human behavior, we use Principal Components Analysis (PCA) to find the main axis of the object and grasping area, respectively. PCA is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much as possible of the data's variation. Through this dimensionality reduction method, we can get the main axis by reducing the dimensionality of the object's point cloud. The red lines in Figure 7 show the main axis of the grasping area of the object

obtained by PCA. Besides, the M paths generated through the Fibonacci sphere contain different directions in the sphere. Some of the paths will pass through the desktop, depicted by a blue plane in Figure 8, which is infeasible in reality. As a result, we can directly delete the paths that will pass through the desktop by removing them from the candidates. Figure 8 shows some of the paths that have been removed from the candidate sets.

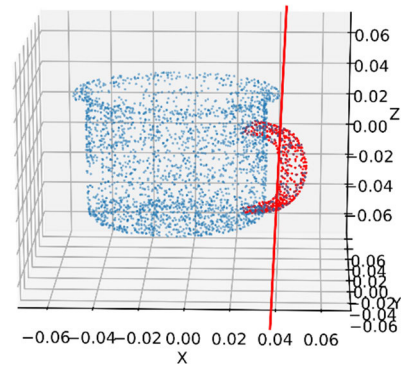


FIGURE 7. Main axis of the grasping area.

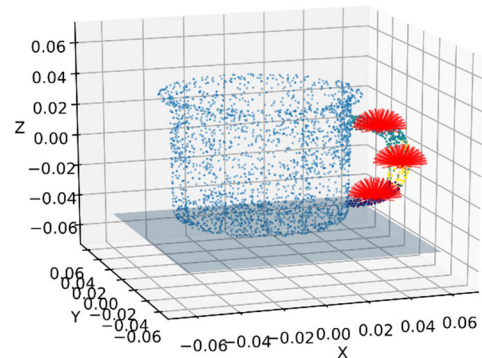


FIGURE 8. Some infeasible candidate paths are removed.

After creating M paths of each cluster as candidates, an optimal grasping path is required from among all candidates as the best grasp. Thus, the other objective of this work is to estimate an optimal grasping pose for robotic manipulation to successfully grasp the object. The best grasp of the target object can be reasonably assumed to contain the following features: 1) The grasping path should be as orthogonal to the main axis as possible. 2) There are more points within the radius of the gripper for the gripper to grasp and fewer points outside the radius to prevent the gripper from colliding with object. To achieve these objectives, a cost function is defined to evaluate the cost for each generated grasping path, which is given by

$$J_p = \frac{0.1 * |\alpha_p - 90| + 0.5}{8} * \sum_{i=1}^N \min \left(1, \frac{20}{(Y_{ip} - r_k)^2 + 1} \right),$$

$$p = 1, \dots, M,$$

where α_p is the angle between path p and the main axis, Y_{ip} represents the distance from points i to grasp path p , r_k represents the radius of the k th cluster, and N represents the object's point cloud. The constants in (1) are empirically obtained. In order to find the grasping path with the two characteristics, Equation (1) contains 2 terms multiplied together, where the first term $\frac{0.1 * |\alpha_p - 90| + 0.5}{8}$ calculates the cost of the intersection angle α_p between path p and the main axis of the object, and the second term

$$\sum_{i=1}^N \min \left(1, \frac{20}{(Y_{ip} - r_k)^2 + 1} \right)$$

calculates the cost of the distance Y_{ip} from all points i to path p in the object's point cloud. Through Equation (1), we can find that the path resulting in a lower value from the cost function implies that the path is more orthogonal to the main axis and it has a higher success rate within the radius of the gripper to complete the grasping.

The design of Equation (1) is to find the optimal grasp path suitable for the collaborative mode. Furthermore, we might also utilize this strategy for the bin picking mode when the target object is on a flat table. To this end, we need to be concerned about whether the gripper would collide with the table or not when the object is lying on the table. Taking this into consideration, Equation (1) can be modified as Equation (2) below:

$$J_p = \frac{0.1 * |\alpha_p - 90| + 0.5}{8} * \sum_{i=1}^N \min \left(1, \frac{20}{(Y_{ip} - r_k)^2 + 1} \right) * T(d_k, \beta_p), \quad p = 1, \dots, M \quad (2)$$

$$T(d_k, \beta_p) = \begin{cases} \frac{(\beta_p - \pi/2)^2}{2} + 0.05, & \text{if } d_k \leq g \\ 1, & \text{otherwise} \end{cases}, \quad p = 1, \dots, M, \quad (3)$$

where d_k is the distance from the center of the k th cluster to the table, β_p is the angle between path p and the table, and g represents the maximum open range of the gripper. The other values are empirically obtained. To prevent the gripper from colliding with the table when grasping the object on the table, there is an extra third term expressed in Equation (3) to multiply with Equation (1), where d_k is used to judge whether the grasping point is too close to the table or not. Once the grasping point is too close to the table, the term $(\beta_p - \pi/2)^2/2 + 0.05$ in Equation (3) will yield the value of the intersecting angle β_j between path p and the table. Through Equation (2), we find that a path resulting in a lower value from the cost function implies that this path is not only more orthogonal to the main axis with a higher success rate within the radius of the gripper to complete the grasping, but also prevents the gripper from colliding with the table.

Besides finding the optimal grasping path, we still want to find the grasp pose for the two-finger gripper. As a result, we find a vertical vector of the grasping path and the main

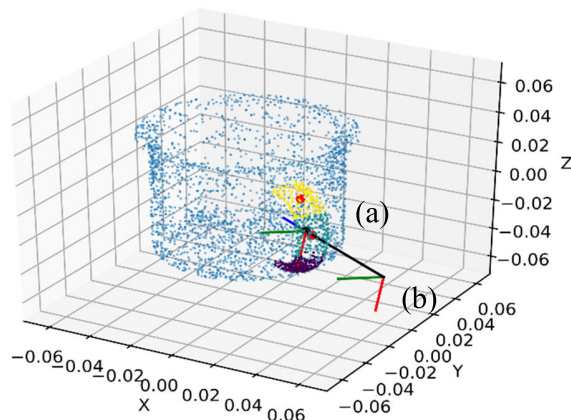


FIGURE 9. Optimal grasp pose produced by the proposed grasping strategy.

axis of the grasping area which is produced by PCA to find the 6DoF grasp pose. As illustrated in Figure 9, the coordinate system (a) shows the optimal grasp pose of the object, where the blue axis represents the best grasping path, which is the Z-axis of the gripper, the red axis is the vector of the main axis found by PCA which is the X-axis of the gripper, and the last green axis is a vertical vector calculated through the grasping path and the main axis, which is used as the Y-axis of the gripper. The coordinate system (b) shows the position and angle of the previous axis of the gripper in the robot arm. On the other hand, we can directly determine the suitable open range of the gripper from r_k representing the radius of the k th cluster.

IV. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

In this paper, we have proposed an optimal grasping strategy to find an optimal grasp pose for a robotic arm in different scenarios. In order to evaluate our proposed optimal grasping strategy, we build a simulation environment to verify the grasping success rate in various tasks.

V-REP (simulator CoppeliaSim) [28], an integrated development environment, is based on a distributed control architecture: each object/model can be individually controlled via an embedded script, a plugin, an ROS, or BlueZero node, a remote API client, or a custom solution. This makes CoppeliaSim very versatile and ideal for multi-robot applications. Controllers can be written in C/C++, Python, Java, Lua, Matlab, or Octave. As a result, we build a simulation environment on V-REP to verify the capability of the proposed optimal grasping strategy.

We create a simulation environment shown in Figure 10 (a) that uses only a gripper (RG2), considering only the end-effector pose (x, y, z , roll, pitch, yaw) to simplify the complexity and concentrate on evaluating the proposed approach. We call this simulation environment the collaborative mode, which uses a box to support the target. The box would not be used to calculate the collision with the

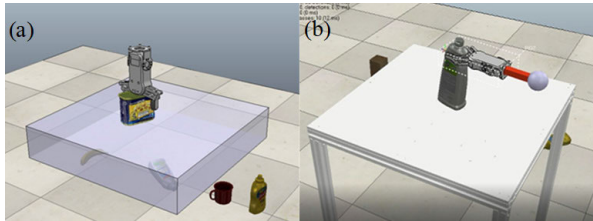


FIGURE 10. Simulation environment with a parallel gripper in (a) the collaborative mode and (b) the bin picking mode.

gripper after setting it in the simulation environment. Besides, we build another simulation environment to simulate grasping the target object placed on the table with a gripper. We call this simulation environment the bin picking mode, which is shown in Figure 10 (b).

B. DATASETS

To evaluate the performance of the proposed approach, we conduct experiments of grasping strategy on two well-known object datasets, i.e., the LINEMOD dataset [29] and the YCB-Video dataset [30]. LINEMOD dataset not only contains the pose of the object but also a 3D model of each object. There are 13 different objects in the LINEMOD dataset; ape, bench vise, cam, can, cat, driller, duck, egg box, glue, hole puncher, iron, lamp, and phone, as shown in Figure 11. The advantage of adopting the LINEMOD dataset as our experimental dataset is that each object has a different shape and size. However, there are several objects which are much bigger than the others, such as bench vise, egg box, hole puncher, and lamp. Thus, this will cause challenges in grasping. Figure 12 shows the grasping area and original point cloud depicted in red points and blue points, respectively. Besides the LINEMOD dataset, we also choose nine objects from the YCB-Video dataset as PointnetGPD [19] selected, including cleanser bottle, mug, meat can, tomato soup can, banana, power drill, mustard bottle, wood block, and screwdriver, as shown in Figure 13. Each object also has its particular shape and size. However, several objects are much bigger than the others, which would cause more challenges during the grasping experiment. For example,



FIGURE 11. 3D model of objects in the LINEMOD dataset.

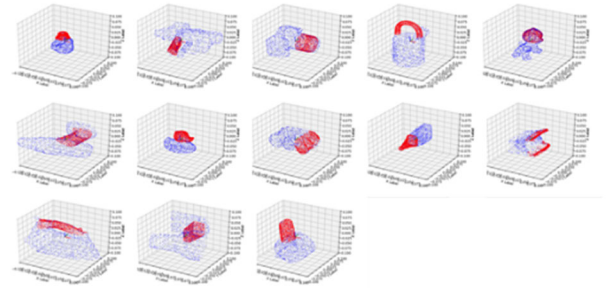


FIGURE 12. Determined grasping area (in red points) of the objects in the LINEMOD dataset.

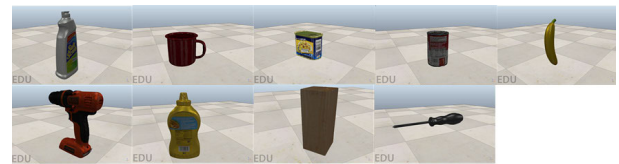


FIGURE 13. 3D model of objects in the YCB-Video dataset.

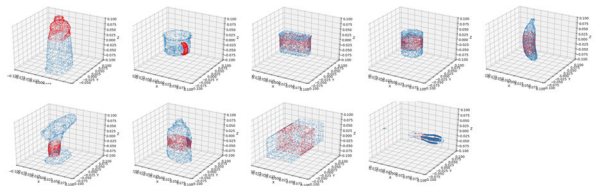


FIGURE 14. Determined grasping area (in red points) of the objects in the YCB-Video dataset.

the wood block may be too big for some parallel grippers. In Figure 14, the determined grasping area and original point cloud for the selected objects are depicted in red points and blue points, respectively. Note that the grasping areas shown in Figs. 12 and 14 are determined for illustration purpose only, and can be altered to suit the needs of different applications.

C. EVALUATION METRICS

To evaluate the grasping success rate of the proposed optimal grasping strategy, two different grasping scenarios were conducted in our experiments, called the collaborative mode and the bin picking mode, respectively. In the collaborative mode, we aimed to make this experiment as similar as possible to the real world. First, we let the object randomly rotate and fall on the box. After moving the gripper to the grasping pose, which is estimated from the proposed grasping strategy, the gripper will close its fingers and try to capture the target. Then the box is removed to simulate that a human is not holding the object. We call it a grasping success if the object was still held by the gripper. In the bin picking mode, we built up a table in the simulation environment, which is shown in Figure 10 (b). First, we randomly rotate the object and let it fall on the table. Unlike the collaborative mode experiment, the gripper would grasp the object and pull it back from the

table to simulate picking objects up from the table in the real world. If the gripper collides with the table or the object falls from the gripper, we call it a grasping failure. On the contrary, if the gripper still held the object after pulling it back, we call it a grasping success.

D. IMPLEMENTATION DETAILS

Besides designing two different scenarios, we also set up two different experiments in these two scenarios. First, we randomly rotate the object along its Z-axis and drop it onto the box or table. Second, we randomly rotate the object along its X-axis, Y-axis, and Z-axis, then drop the object onto the box or table. We recorded the success rate through our evaluation method in collaborative mode and bin picking mode. In each of the experiments, we tested ten rounds on each of the objects in the LINEMOD dataset and YCB-Video dataset.

We have pre-determined each object’s grasping area in two different datasets, which are shown in Figure 11 and Figure 13. The cluster K depends on the gripper size and the size of the grasping area. Besides, we set the parameter M as 256 to create grasp paths of each cluster in these experiments.

Figure 15 shows the experiment process of the object “bench vise” from the LINEMOD dataset in the collaborative mode. First, an optimal grasp pose is determined from the proposed method and the gripper opens according to the diameter of the grasping area found by the method too, which is shown in Figure 15 (a). Then the gripper moves to the grasp pose and closes the gripper to grasp the object. Finally, the box is removed to simulate that the object is not held by hand. If the object is still grasped by the gripper, we call it a grasping success in this experiment, as Figure 15 (c)

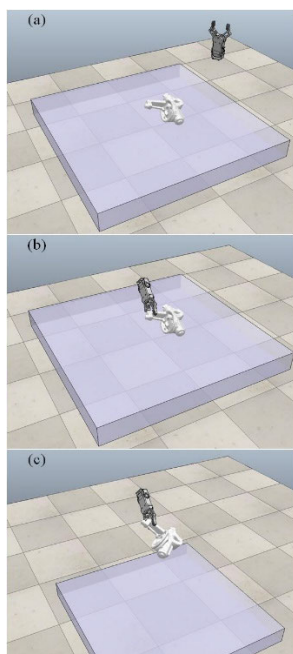


FIGURE 15. Experimental results of the object “bench vise” in the collaborative mode.

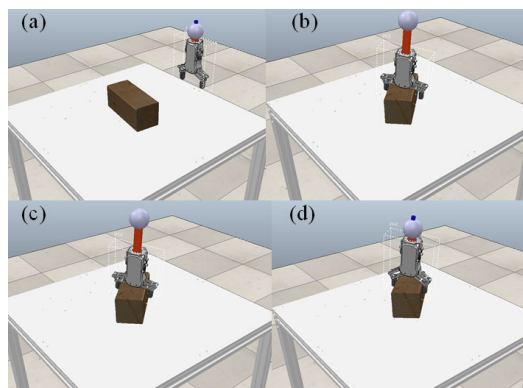


FIGURE 16. Experimental results of the object “wood block” in the bin picking mode.

shows. Figure 16 shows another experiment process using the object “wood block” from the YCB-Video dataset in bin picking mode. Different from the collaborative mode, the gripper performs the actions “extend” and “pull back” to simulate picking an object with a robotic arm, as shown in Figure 16 (b) and (d), respectively. Besides, if the gripper does not collide with the table in the process and still grasps the object after pulling it back from the table, we call it a success in this experiment which is shown in Figure 16 (d). Through our proposed optimal grasping strategy, we not only find the optimal grasp pose but also determine a suitable open range of the gripper according to the diameter of the grasping area. Figure 17 shows the experiment on the objects “mug” and “woodblock” in the YCB-Video dataset. We can see that the open range of the gripper in Figure 17 (b) when it grasps the object “wood block” is much bigger than grasping the object “mug” shown in Figure 17 (a). Because a suitable open range of the gripper can be used, collision with the other parts of the target object can be avoided during the grasping process. As a result, better performance can be obtained.

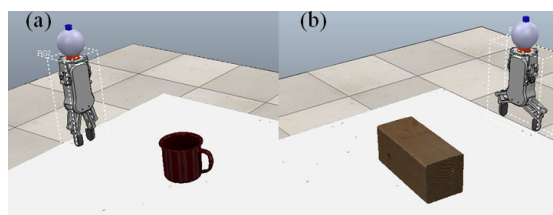


FIGURE 17. Open range of the parallel gripper for different objects.

E. COMPARISON RESULTS AGAINST THE STATE-OF-THE-ART METHODS

For better performance demonstration, the comparison of the proposed optimal grasping strategy with respect to GPD and PointnetGPD [19] in grasping is conducted to evaluate the success rate of different scenarios on the YCB-Video. As illustrated in Table 1, the proposed optimal grasping strategy in different scenarios manifests a higher average grasping

TABLE 1. Comparison between the proposed method and state-of-the-art approaches in the grasping success rate on the YCB-Video dataset.

YCB-Video	GPD [18]	PointnetGPD 2-class [19]	PointnetGPD 3-class [19]	Ours-Z (Collaborative robots)	Ours-XYZ (Collaborative robots)	Ours-Z (Bin picking)	Ours-XYZ (Bin picking)
	Success rate (%)	Success rate (%)	Success rate (%)	Success rate (%)	Success rate (%)	Success rate (%)	Success rate (%)
Cleanser B.	100	100	90	100	90	100	100
Mug	30	50	70	100	100	80	60
Meat Can	60	80	70	90	90	100	100
Tomato Soup	90	100	100	100	100	70	100
Banana	20	90	90	100	100	100	100
Power Drill	80	70	90	100	100	100	100
Mustard B.	90	100	90	90	100	100	100
Wood Block	90	90	90	50	70	100	100
Screwdriver	20	70	80	100	100	100	100
Average	64.44	83.33	85.56	92.22	94.44	94.44	95.56

success rate than those of the state-of-the-art methods. The proposed optimal grasping strategy reaches a 95.56% success rate in bin picking mode, which is better than PointnetGPD [19] and GPD [18], and means that our proposed grasping strategy has a better grasping ability. The proposed method also reaches a 94.44% success rate in the collaborative mode, which shows the feasible grasping performance for handing the objects. Figure 17 illustrates the advantage of the proposed method, where a suitable open range for the gripper is determined to grasp different objects “mug” and “wood block” in the YCB-Video dataset. As shown in Figure 17, we can see that the open range of the gripper is much bigger when it grasps the object “wood block” in Figure 17 (b) than the case when it grasps the object “mug” in Figure 17 (a). Because a suitable open range of the gripper can be used, collision with the other parts of the target object can be avoided during the grasping process. As a result, better performance can be obtained as shown in Tables 1 and 2. In addition to the comparison with the state-of-the-art methods on the YCB-Video dataset, we also use other objects from the LINEMOD dataset to evaluate our methods. As illustrated in Table 2, our proposed optimal grasping strategy also has good grasping ability in different scenarios. Our methods can reach a 95.38% success rate and 94.62% success rate in collaborative mode and bin picking mode, respectively.

F. COMPUTATIONAL EFFICIENCY OF THE PROPOSED METHOD

To show the running speed of the proposed method to obtain an optimal grasp pose, we need to execute both object pose estimation and the proposed grasping strategy. The experiment of this paper is conducted on a personal computer with Intel (R) Core (TM) i7-9700 @ 3.0GHz, an NVIDIA GeForce RTX 2070 graphic card, and a Logitech C920 webcam.

TABLE 2. Grasping success rate on LINEMOD dataset.

LINEMOD	Ours-Z (Collaborative mode)	Ours-XYZ (Collaborative mode)	Ours-Z (Bin picking)	Ours-XYZ (Bin picking)
	Success rate (%)	Success rate (%)	Success rate (%)	Success rate (%)
Ape	100	100	90	80
Bench vise	100	100	90	100
Cam	100	100	100	100
Can	100	100	100	90
Cat	100	100	100	100
Driller	100	100	100	100
Duck	100	100	100	90
Egg box	60	60	70	90
Glue	100	100	90	100
Hole p.	80	80	100	70
Iron	100	90	100	80
Lamp	100	100	90	100
Phone	100	100	100	100
Average	95.38	94.62	94.62	92.31

According to the experimental results, the method to estimate the object pose [10] for an object has a running time of 0.04 seconds, while the proposed grasping strategy, including the steps of ‘Segment into Clusters’, ‘Create Grasping Paths’, and ‘Cost Function’, has a running time of 0.1 seconds on the above-mentioned platform. Thus, the total computation time required to obtain an optimal grasping pose is about

0.14 seconds, which reveals the feasibility to determine an optimal grasp pose in real time for practical robot grasping applications.

V. CONCLUSION

This paper has proposed an effective and reliable grasping strategy for determining the optimal grasp pose for widely used parallel grippers to deal with target objects with any pose. To evaluate the performance of the proposed method, we compare our method with the state-of-the-art GPD [18] and PointNetGPD [19]. The experimental results show that our proposed method reaches 94.44% and 95.56% success rate in collaborative and bin picking mode on the YCB-Video dataset, respectively, outperforming the state-of-the-art methods. Besides, experimental results of our evaluation using the LINEMOD dataset also show that our method is still feasible in both collaborative and bin-picking modes. Furthermore, since the proposed method only requires the object pose estimation from a RGB camera, it can also be applicable to other adaptive grippers to develop an optimal grasping strategy.

ACKNOWLEDGMENT

The authors would like to thank the National Center for High-Performance Computing for computer time and facilities to conduct this research.

REFERENCES

- [1] *International Federation of Robotics (IFR)*. Accessed: Apr. 3, 2021. [Online]. Available: <https://ifr.org/>
- [2] B. Sjøbakk, M. K. Thomassen, and E. Alfnes, "Implications of automation in engineer-to-order production: A case study," *Adv. Manuf.*, vol. 2, no. 2, pp. 141–149, Jun. 2014.
- [3] Y. Hu, J. Li, Y. Chen, Q. Wang, C. Chi, H. Zhang, Q. Gao, Y. Lan, Z. Li, Z. Mu, Z. Sun, and A. Knoll, "Design and control of a highly redundant rigid-flexible coupling robot to assist the COVID-19 oropharyngeal-swab sampling," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1856–1863, Apr. 2022.
- [4] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proc. Mach. Learn. Res.*, vol. 87, 2018, pp. 651–673.
- [5] Y. Hu, X. Wu, P. Geng, and Z. Li, "Evolution strategies learning with variable impedance control for grasping under uncertainty," *IEEE Trans. Ind. Electron.*, vol. 66, no. 10, pp. 7788–7799, Oct. 2019.
- [6] Q. Bai, S. Li, J. Yang, Q. Song, Z. Li, and X. Zhang, "Object detection recognition and robot grasping based on machine learning: A survey," *IEEE Access*, vol. 8, pp. 181855–181879, 2020.
- [7] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—A survey," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 289–309, Apr. 2014.
- [8] C. Wang, D. Xu, Y. Zhu, R. Martin-Martin, C. Lu, L. Fei-Fei, and S. Savarese, "DenseFusion: 6D object pose estimation by iterative dense fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3338–3347.
- [9] S.-K. Huang, C.-C. Hsu, W.-Y. Wang, and C.-H. Lin, "Iterative pose refinement for object pose estimation based on RGBD data," *Sensors*, vol. 20, no. 15, p. 4114, Jul. 2020.
- [10] Y. He, W. Sun, H. Huang, J. Liu, H. Fan, and J. Sun, "PVN3D: A deep point-wise 3D keypoints voting network for 6DoF pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 11629–11638.
- [11] J.-K. You, C.-C.-J. Hsu, W.-Y. Wang, and S.-K. Huang, "Object pose estimation incorporating projection loss and discriminative refinement," *IEEE Access*, vol. 9, pp. 18597–18606, 2021.
- [12] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Hong Kong, May/Jun. 2014, pp. 1997–2004.
- [13] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. Aparicio Ojea, and K. Goldberg, "Dex-Net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," 2017, *arXiv:1703.09312*.
- [14] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robot. Auton. Syst.*, vol. 58, no. 4, pp. 362–377, Apr. 2010.
- [15] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from RGBD images: Learning using a new rectangle representation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Shanghai, China, May 2011, pp. 3304–3311.
- [16] M. Vohra, R. Prakash, and L. Behera, "Real-time grasp pose estimation for novel objects in densely cluttered environment," in *Proc. 28th IEEE Int. Conf. Robot Hum. Interact. Commun. (RO-MAN)*, New Delhi, India, Oct. 2019, pp. 1–6.
- [17] Y. Yu, Z. Cao, S. Liang, W. Geng, and J. Yu, "A novel vision-based grasping method under occlusion for manipulating robotic system," *IEEE Sensors J.*, vol. 20, no. 18, pp. 10996–11006, Sep. 2020.
- [18] A. T. 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.
- [19] 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.
- [20] A. Mousavian, C. Eppner, and D. Fox, "6-DOF GraspNet: Variational grasp generation for object manipulation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct./Nov. 2019, pp. 2901–2910.
- [21] B. Zhao, H. Zhang, X. Lan, H. Wang, Z. Tian, and N. Zheng, "REG-Net: Region-based grasp network for end-to-end grasp detection in point clouds," 2020, *arXiv:2002.12647*.
- [22] A. T. Miller, S. Knoop, P. K. Allen, and H. I. Christensen, "Automatic grasp planning using shape primitives," in *Proc. IEEE Int. Conf. Robot. Autom.*, Taipei, Taiwan, Sep. 2003, pp. 1824–1829.
- [23] N. Vahrenkamp, L. Westkamp, N. Yamanobe, E. E. Aksoy, and T. Asfour, "Part-based grasp planning for familiar objects," in *Proc. IEEE-RAS 16th Int. Conf. Humanoid Robots (Humanoids)*, Cancun, Mexico, Nov. 2016, pp. 919–925.
- [24] T. Patten, K. Park, and M. Vincze, "DGCM-Net: Dense geometrical correspondence matching network for incremental experience-based robotic grasping," 2020, *arXiv:2001.05279*.
- [25] H.-Y. Lin, S.-C. Liang, and Y.-K. Chen, "Robotic grasping with multi-view image acquisition and model-based pose estimation," *IEEE Sensors J.*, vol. 21, no. 10, pp. 11870–11878, May 2021.
- [26] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, "MeshLab: An open-source mesh processing tool," in *Proc. 6th Eurograph. Italian Chapter Conf.*, Salerno, Italy, Jul. 2008, pp. 129–136.
- [27] X. Jin and J. Han, "K-means clustering," in *Encyclopedia of Machine Learning*, C. Sammut and G. I. Webb, Eds. Boston, MA, USA: Springer, 2011.
- [28] *The Robotics Simulator CoppeliaSim*. Accessed: Apr. 3, 2021. [Online]. Available: <https://www.coppeliarobotics.com/>
- [29] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," in *Computer Vision—ACCV (Lecture Notes in Computer Science)*, vol. 7724. Berlin, Germany: Springer, 2012.
- [30] B. Calli, A. Singh, A. Walsman, S. S. Srinivasa, P. Abbeel, and A. M. Dollár, "The YCB object and model set: Towards common benchmarks for manipulation research," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Seattle, WA, USA, Jul. 2015, pp. 510–517.



HSIN-HAN CHIANG (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical and control engineering from the National Chiao-Tung University, Hsinchu, Taiwan, in 2001 and 2007, respectively. He was a Postdoctoral Researcher in electrical engineering from the National Taipei University of Technology, Taipei City, Taiwan, in 2008 and 2009. From August 2009 to January 2017, he was an Assistant Professor and an Associate Professor with the Department of Electrical Engineering, Fu Jen Catholic University, New Taipei City, Taiwan. From February 2017 to January 2022, he was an Associate Professor with the Department of Electrical Engineering, National Taiwan Normal University, Taipei City. Since February 2022, he has been an Associate Professor with the Department of Vehicle Engineering, National Taipei University of Technology. His research interests include the area of intelligent systems and control, autonomous driving systems, vehicle dynamics and control, and autonomous mobile robots.



JIUN-KAI YOU received the M.S. degree in electrical engineering from the National Taiwan Normal University, Taipei City, Taiwan, where he is currently pursuing the master's degree with the Department of Electrical Engineering. His research interests include computer vision and robotics.



CHEN-CHIEN JAMES HSU (Senior Member, IEEE) was born in Hsinchu, Taiwan. He received the B.S. degree in electronic engineering from the National Taiwan University of Science and Technology, Taipei City, Taiwan, in 1987, the M.S. degree in control engineering from the National Chiao-Tung University, Hsinchu, in 1989, and the Ph.D. degree from the School of Microelectronic Engineering, Griffith University, Brisbane, QLD, Australia, in 1997. He was a Systems Engineer with IBM Corporation, Taipei City, for three years, where he was responsible for information systems planning and application development, before commencing his Ph.D. studies. He joined the Department of Electronic Engineering, St. John's University, New Taipei City, as an Assistant Professor, in 1997, and was appointed as an Associate Professor, in 2004. From 2006 to 2009, he was with the Department of Electrical Engineering, Tamkang University, New Taipei City. He is currently a Professor with the Department of Electrical Engineering, National Taiwan Normal University, Taipei City. He is the author or coauthor of more than 200 refereed journals and conference papers. His current research interests include digital control systems, evolutionary computation, vision-based measuring systems, sensor applications, and mobile robot navigation. He is a fellow of IET.



JUN JO received the Ph.D. degree from the University of Sydney, in 1994. He worked as a Postdoctoral Research Fellow with the Key Centre of Design Computing, University of Sydney, until he joined Griffith University, in 1996. He has been working on various research projects, including computer vision and machine learning, and their applications in various areas including robotics, autonomous cars, drones, the IoTs, satellite data analysis, and medical image analysis. He is also the Program Director of Bachelor of Intelligent Digital Technologies (BIDT) Degree Program at Griffith University, Australia.

...