Integration of Object Detection, Grasp Pose Estimation and Gripping Force Compensation for a Six-DoF Robotic Arm Pick-and-Place System

Hao-En Chang, Rongshun Chen

***Abstract*—It is crucial for the research of object localization, grasp pose estimation and the secure handling of objects to prevent damage or dropping during the pick-and-place process. This study employs the Mask R-CNN algorithm to locate the object, and to obtain the mask. Then, the mask is combined with the depth image to generate a point cloud. Subsequently, an algorithm is proposed to determine whether the object is suitable for vacuum gripper grasping. For an unsuitable case, the point cloud is fed into a PointNet++ model, which utilizes geodesic distance as the loss function to predict a grasp pose for the parallel gripper in an end-to-end manner. Additionally, to achieve stable grasping, this study arranges eight FSRs in an array configuration. By analyzing contact force information, it can detect the slippage, caused by insufficient applied force, and further compensates the gripping force. To enhance the performance of the proposed object pick-and-place system, a hand-eye calibration and a motion trajectory planning are performed. Finally, the system is tested on a six-DOF robotic arm, involving objects such as ball, bottle, paper cup, box and wooden block. The proposed system achieves a success rate of 92% in object pick-and-place over 100 experiments.**

***Index Terms*—Force Sensing Resistor, Grasp Pose, Object Detection, PointNet++, Robotic Arm**

I. Introduction

THE utilization of robotic arms has primarily been confined to industrial environments, where they are pre-programmed with specific parameters to handle and process objects. However, a recent trend has emerged in which robotic arms are being incorporated into daily life scenarios. This trend necessitates the requirement for robotic arms to be capable of managing objects of diverse sizes, shapes, and materials. Therefore, it becomes crucial to address the research of object localization, grasp pose estimation, and ensuring secure object, handling to prevent damage or dropping during the picking process.

Point clouds have found widespread use in the field of robotics research as they contain geometry information of objects. However, estimating grasp poses becomes challenging when dealing with complex object shapes. Moreover, relying solely on partial geometric representations within the point cloud can introduce biases into grasp pose calculations. In order to address these challenges, several methodologies [1-5] have adopted deep learning models to learn the implicit relationship between object point clouds and grasp poses.

Wang and Lin [6] employ PointNet [7] as feature learning network, aiming to directly regress a quaternion for the grasp pose. However, their approach of using Euclidean distance as the loss function and manipulating the dataset by randomly multiplying the ground truth quaternions by -1 is not optimal for minimizing training errors. As an alternative, we propose utilizing geodesic distance and ensuring that the real part of ground truth quaternions is non-negative. Further details are explained in Section 2.

To enhance the safety of the gripping process, a force compensation system has been developed. By integrating sensors with the gripper, the system is capable of perceiving the contact information between the gripper and the object. The primary goal of the force compensation system is to detect potential slippage and adjust the force accordingly to prevent the object from dropping. Details are discussed in Section 3.

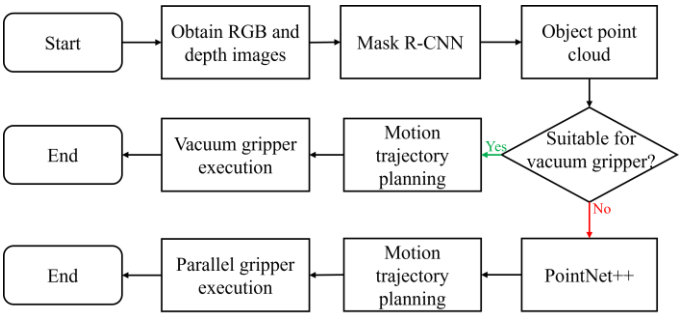


Fig. 1. Overall system flowchart

Fig. 1 elaborates on the overall system flowchart of the proposed system. Firstly, an Intel RealSense™ D435i depth camera is installed on the end-effector. Once RGB and depth images are captured, Mask R-CNN [8] is utilized to generate a mask of the desired object on the RGB image. The mask is then combined with depth information to generate a point cloud. Subsequently, an algorithm is proposed to determine whether the object is suitable for vacuum gripper grasping. If not, the object point cloud is fed into a deep learning model to predict a grasp pose for parallel gripper. Finally, motion trajectories are planned, and object pick-and-place process is conducted, where a force compensation algorithm will be involved during the process. A six-degrees-of-freedom robotic arm, Ufactory Lite 6, is used to demonstrate the experiments. Robot experiments will be discussed in Section 4.

Manuscript received 08 December 2023; accepted 11 January 2024; date of publication 18 January 2024. This work was supported in part by Ministry of Sciences and Technology, TAIWAN, project under the grant number MOST 111-2221-E007-099.

Hao-En Chang is with the Department of Power Mechanical Engineering, National Tsing Hua University (NTHU), Hsinchu, Taiwan (e-mail: sean1999880304@yahoo.com.tw).

Rongshun Chen is a Distinguished Professor with the Department of Power Mechanical Engineering, National Tsing Hua University (NTHU), Hsinchu, Taiwan (corresponding author, email: rchen@pme.nthu.edu.tw).

II. Grasp Pose Estimation

A. Vacuum Gripper Grasp Pose Estimation

To detect the experiment objects, a Mask R-CNN model was trained on a dataset of 200 images using transfer learning. The trained model is then utilized to detect the mask of the object, whose point cloud is generated by combining the mask with the depth image. The proposed algorithm aims to find the most suitable plane within the point cloud. Fig. 2 illustrates the flowchart of grasp pose estimation for vacuum gripper.

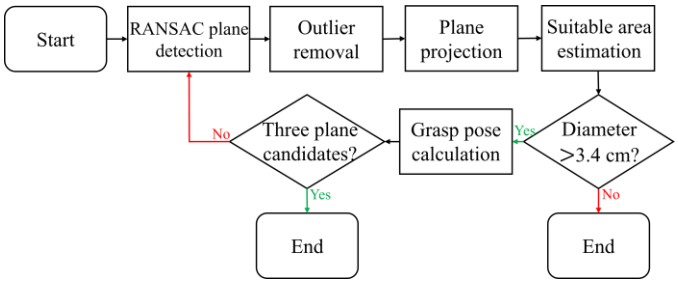


Fig. 2. Vacuum grasp pose estimation flowchart

First, we implement the random sample consensus (RANSAC) algorithm to detect a plane within the point cloud. Outlier removal is then conducted to remove points that are far from their neighboring points. Furthermore, the inliers are projected onto the plane, reducing them to 2-D points. Next, the 2-D convex hull is computed on the plane to create a polygon that encompasses all the data points. The centroid of this polygon is determined by averaging the maximum and minimum values of its vertices in both the x and y directions. Using this centroid as the center, a circle is established within the bounds of the convex polygon. Subsequently, the diameter of circle is calculated and is compared to the diameter of the suction cup. To account for potential errors of point cloud data in acquisition and localization, the diameter of suction cup (1.7 cm) is multiplied by a safety factor (2), yielding a threshold of 3.4 cm. If the calculated diameter of circle is greater than this threshold, it is considered suitable for vacuum gripper. The grasp position and pose can be calculated by back-projecting the center of the circle into 3-D space and employing principal component analysis (PCA) method, respectively. As the object point cloud might comprise multiple planes, the RANSAC plane detection is applied to the outliers following the generation of the first plane. This iterative process is repeated up to three times. Prioritization is assigned to the planes. Theoretically, the closer the gripper's z-axis aligns with the direction of gravity, the more stable the grasping. Consequently, we calculate dot products between the z-axis vector of plane and the base coordinate z-axis of robotic arm's ($\vec{v}_z = (0, 0, 1)$). A smaller value from the dot product indicates a higher priority level.

Fig. 3 demonstrates the results obtained from the algorithm applied to five different objects. For the ball, bottle, and paper cup, the calculated suitable area diameters were smaller than the threshold of 3.4 cm, indicating them as unsuitable for vacuum gripper. However, the calculations for the box and wooden block reveal suitable regions for vacuum gripper.

B. Parallel Gripper Grasp Pose Estimation

In this work, we implemented PointNet++ [9], a successor to PointNet, which enhances the ability to extract local features of point cloud by involving the addition of two steps: sampling and grouping. The model learns hierarchical features by repeatedly performing sampling and grouping, ultimately achieving higher accuracy. The proposed network architecture is illustrated in Fig. 4. The model takes object point clouds as input and extracts local features through repeated sampling and grouping. Finally, it connects to a multilayer perceptron (MLP) with a modified last layer of four neurons, tailored to predict the four quaternion values. As the model has been modified from classification to numerical regression, the softmax function should be substituted with the identity function.

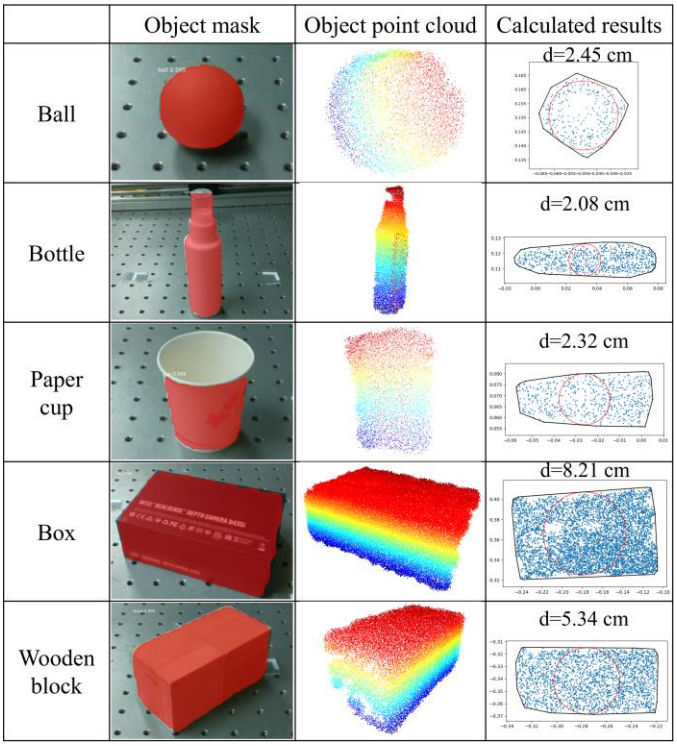


Fig. 3. Vacuum grasp pose estimation results

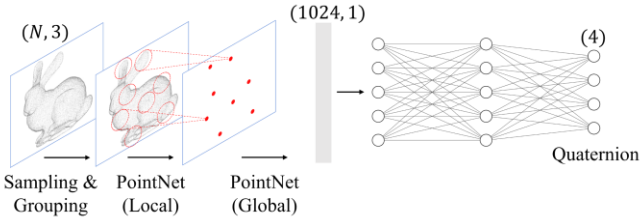
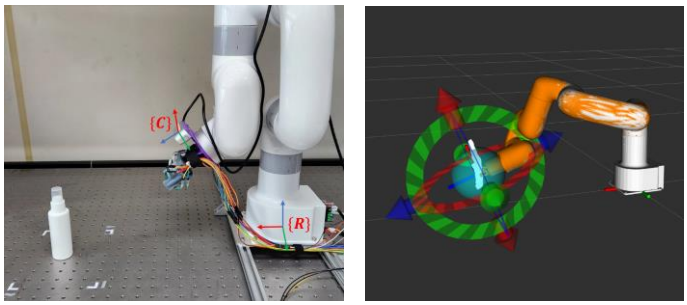


Fig. 4. Parallel gripper grasp pose prediction network architecture

During the dataset collecting process, objects were individually placed on the area as shown in Fig. 5(a). The obtained point cloud was published as PointCloud2 to ROS topic and displayed in RViz. In the virtual environment, the end-effector was dragged to annotate a desired grasp pose corresponding to the object as shown in Fig. 5(b). A unit quaternion representing the tool center point (TCP) coordinate in relation to the base coordinate was obtained by calling ROS TF package. Object point cloud was further sampled to a number of 1024 points and then normalized to the interval of [-1, 1].

Since data collecting process is time-consuming, data augmentation was employed to reduce time cost. Assuming the

original point cloud is captured when the object is placed at the 0° as shown in Fig. 6. Both point cloud and corresponding grasp pose are multiplied by a rotation around z-axis of base coordinate by an angle θ, to simulate the object, placed at different positions. The horizontal field of view of depth camera is approximately 70°, thus the object is rotated by an angle of θ degrees around the z-axis within the range of ±35° degrees to generate new data.



(a) Environment setup (b) Grasp pose annotation

Fig. 5. Dataset collecting process

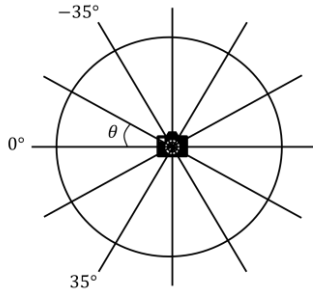


Fig. 6. Data augmentation

A quaternion is represented as $q_0 + q_1 i + q_2 j + q_3 k$, where q_0, q_1, q_2 and q_3 are real number, and i, j and k are the basis elements. Suppose a unit quaternion q ($\|q\| = 1$), q and $-q$ represent the same orientation in 3-D space. To tackle this property, Wang and Lin [6] randomly multiply the ground truth quaternions by -1, and use Euclidean distance, as defined in (1), as loss function to train on PointNet model.

$$Loss = \left\| q_{GT} - q_{pred} \right\|_2 \tag{1}$$

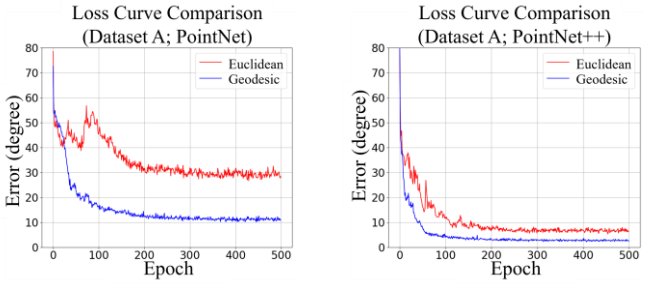
However, a problem may arise by doing so. Given two similar input data and their grasp poses q_1 and q_2. Suppose q_2 was randomly multiplied by -1, resulting in a new value denoted as q_2'. The predicted values p_1 and p_2 are estimated during the training process. If we calculate Euclidean error between the ground truths and the predicted values, we can find that one of the errors is small while the other one is relatively large. In this scenario, the model is not able to minimize the overall error. While geodesic distance measures the angle of difference between two unit quaternions, as defined in (2).

$$\theta = 2\cos^{-1}\left(\left| q_{GT} \cdot q_{pred} \right|\right) \tag{2}$$

In this case, two errors are found to be nearly the same, the model is now able to minimize overall error.

Finally, 30 samples were collected. Data augmentation was performed with θ = 5°, resulting in a total of 450 samples. To facilitate a comparison between the proposed method and Q-PointNet, the ground truth quaternions were randomly multiplied by -1, creating dataset A, which was then randomly divided into a training set of 360 samples and a testing set containing 90 samples. The training parameters employed were as follows: Adam optimizer was used with learning rate of 0.001, momentum of 0.9, weight decay of 0.0001, batch size of 45, and total of 500 epochs. The training environment was set up on Ubuntu 20.02, and a GeForce RTX 3090 GPU was utilized to accelerate the training process.

Fig. 7 shows the training loss curve of dataset A. It is evident from Fig. 7(a) that both loss curves have converged by epoch 500. However, it is obvious that the model trained using Euclidean distance as the loss function converged with a relatively large error than the one using geodesic distance. Fig. 7(b) shows the training loss of models on PointNet++. The results demonstrate that both models significantly reduce the errors in comparison to Fig. 7(a). Moreover, Fig. 7(b) reveals that the error of using Euclidean distance is still slightly larger than the one using geodesic distance.



(a) Loss function comparison of PointNet (b) Loss function comparison of PointNet++

Fig. 7. Training loss curves of dataset A

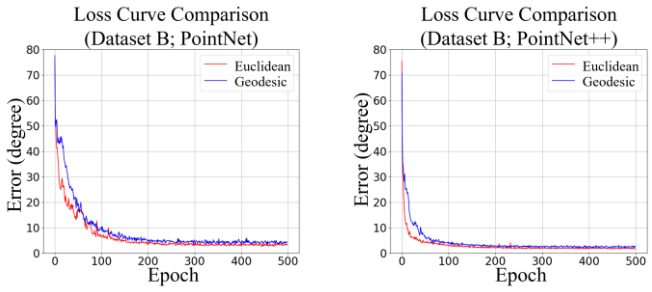
This work proposes an alternative approach for manipulating datasets. Consider that a quaternion q and $-q$ represent the same orientation, it is possible to train the model using only one representation of quaternions. This can be achieved by ensuring that the real part of the quaternion remains non-negative for all ground truth samples, as defined in (3)

$$q_n = \begin{cases} q_n, & \text{if } \mathrm{Re}\{q_n\} \geq 0 \\ -q_n, & \text{if } \mathrm{Re}\{q_n\} < 0 \end{cases} \tag{3}$$

The processed dataset is referred to as dataset B. Fig. 8 shows the training results of dataset B that four trained models have converged by epoch 500 with small error. Additionally, in Fig. 8(b), models trained on PointNet++ reach slightly smaller errors compared to those trained on PointNet, as shown in Fig. 8(a).

Finally, all eight trained models were tested on the testing set as shown in Table 1. The results show that by using Q-PointNet's proposed method (dataset A trained on PointNet

with Euclidean distance), the model achieves a test error of 29.6°. On the other hand, dataset B trained on PointNet++ with both Euclidean and geodesic distances achieve test errors of 5.8° which are significantly smaller.



(a) Loss function comparison of PointNet

(b) Loss function comparison of PointNet++

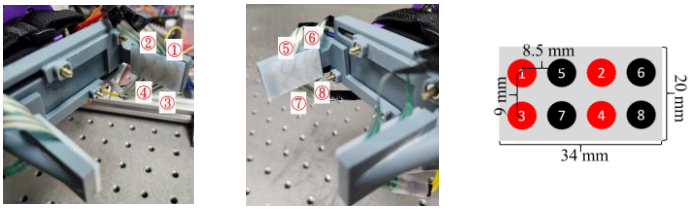
Fig. 8. Training loss curves of dataset B

TABLE I
TEST ERRORS OF EIGHT MODELS

Loss function \ Model	Dataset A		Dataset B	
	PointNet	PointNet++	PointNet	PointNet++
Euclidean	29.6°	13.9°	5.9°	5.8°
Geodesic	13.2°	6.9°	7.4°	5.8°

III. FORCE COMPENSATION SYSTEM

The proposed force compensation system consists of an actuator, MG-995 servo motor, 3D-printed gripper parts created by papabravo [10] and eight FSR 400s manufactured by Interlink Electronics. Force calibration was performed on all eight sensors to convert the output voltage to force values.



(a) Left side of gripper

(b) Right side of gripper

(c) Configuration of FSRs projected onto the right-side plane

Fig. 9. Arrangement of FSRs on parallel gripper

Four FSRs were installed on each side of gripper as shown in Fig. 9(a) and Fig. 9(b). It appears an array-like configuration as shown in Fig. 9(c) by projecting left-side of FSRs onto the right-side. Two silicone films (2 mm thick) are placed on each side of contact surface to provide sufficient friction and ensure the forces are evenly distributed.

Our proposed method is inspired by the grasping mechanism of human hands. When humans attempt to grasp an object, an initial contact grasping force is determined by factor such as the volume of object. As the hands make contact with the object, tactile receptors beneath the skin are used to determine whether the current force is sufficient to lift the object without it slipping off. Once the receptors detect the onset of slippage, hands are prompted to increase the gripping force, thereby ensuring a more stable grasping state.

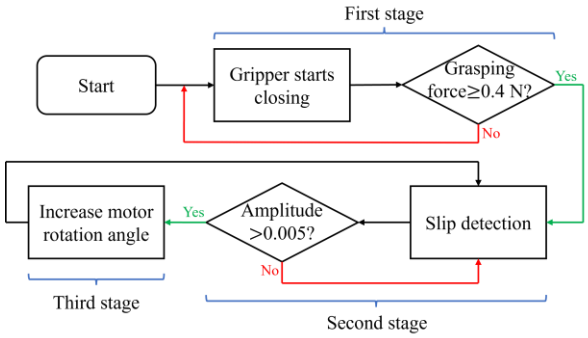


Fig. 10. Force compensation system flowchart

Fig. 10 elaborates on the flowchart of force compensation. In the first stage, the system is initiated and the gripper starts closing until one of the sensors detects a force larger than or equal to 0.4 N, which is determined through fine-tuning for five different objects used in experiments, indicating that the object has been held. In the second stage, eight sensors are employed to detect whether slippage occurs. The method for detecting the slippage proposed by Cheng et al. [11] is implemented. This involves calculating the correlation coefficient of the sensor array as defined in (4).

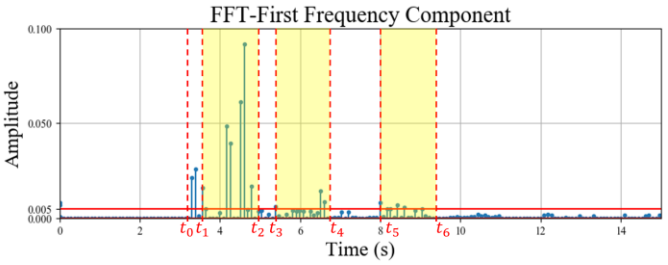
$$correlation = \frac{(X-\bar{X})(Y-\bar{Y})^T}{\sqrt{(X-\bar{X})(X-\bar{X})^T}\sqrt{(Y-\bar{Y})(Y-\bar{Y})^T}} \quad (4)$$

where X and Y represent 1-D arrays of force obtained at the end of the first stage and for each sample of sensor readings during the second stage, respectively. Next, Fast Fourier transform (FFT) is applied, and the amplitude of first frequency component is referred to as the determinant of slippage. If the amplitude exceeds a predefined threshold of 0.005, where the number is also determined through fine-tuning, it is suggested that a slip has occurred or the contact area is not stable.

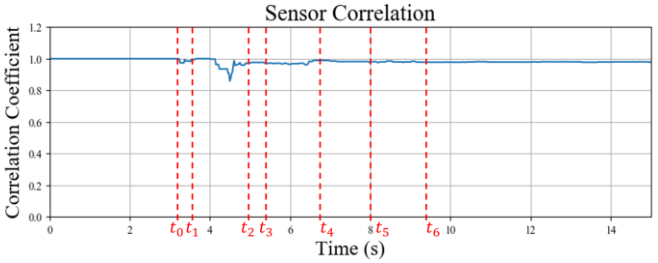
In the third stage, a force compensation is defined as a 5° rotation increment of the motor. After the force is compensated, the system will return to the second stage to detect any further slippages. Note that during the second stage, while the object is being lifted, there could potentially be a false positive slip signal. This might occur due to the change in sensor readings at the contact area, caused by the absence of force being applied from the table. The solution to this is to ignore the first three determinants of slippage once the lift-up command is sent. Additionally, during the third stage, a false positive could happen while the gripper is closing (for 0.2 seconds). There is a time delay of approximately 1 second between when the compensation command is sent and when it is executed. Therefore, determinants of slippage within 1.5 seconds after the compensation command is sent will be ignored in this study.

Fig. 11 demonstrates the results of force compensation experiments conducted on the object of bottle. At time t_0, the bottle is lifted, and the following three amplitudes are ignored.

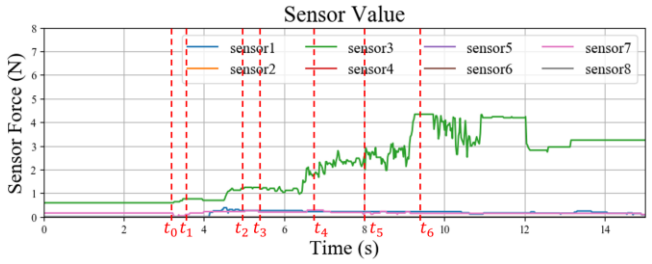
At time t_1, the first slip signal is detected, as shown in Fig. 11(a). The period during $t_1 \sim t_2$ represents the first force compensation. Due to the instability of the contact area, two further slip signals are detected, and compensations take place during $t_3 \sim t_4$ and $t_5 \sim t_6$. Fig. 11(c) illustrates that the sensor readings increase during the force compensation stages.



(a) Slip signal



(b) Correlation coefficient



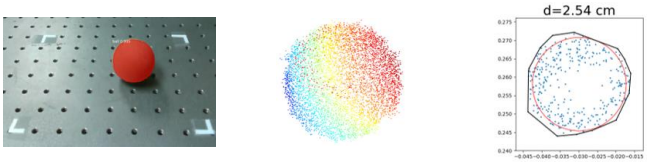
(c) Sensor value

Fig. 11. Force compensation experiment (bottle)

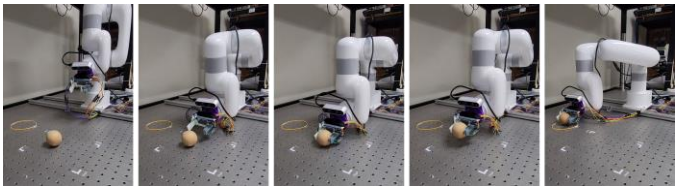
IV. Experiments of Object Pickup

In this section, grasp pose estimation algorithms and force compensation system are integrated and tested on the Lite 6. Fig. 12 demonstrates the grasping experiment of the ball in which Figs. 12(a), 12(b), and 12(c) show the ball mask, object point cloud, and the calculated suitable area for vacuum gripper, respectively. And Fig. 12(d) displays the processes to pick up the ball that is individually placed at random position inside the white rectangle area and move it to the yellow circle area. All objects used in this study follow the same process. One success is defined as the robot completing the entire operation without objects slipping off or being damaged.

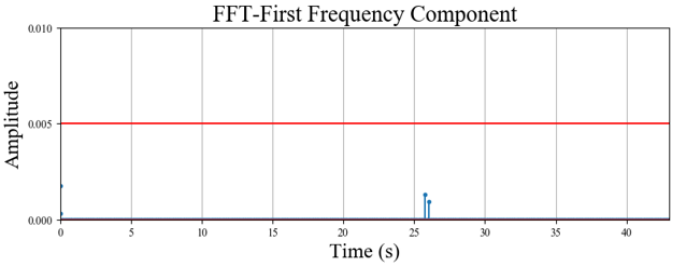
The estimated area has a diameter of 2.54 cm, suggesting using parallel gripper. Hence, a grasp pose was predicted by PointNet++. Fig. 12(d) also demonstrates the five states of the robotic arm at initial, waypoint, grasping, lift-up and final drop-down positions, respectively. It can be seen in Fig. 12(f) that the forces remain stable throughout the entire operation. Therefore, no slip signals are detected, as shown in Fig. 12(e).



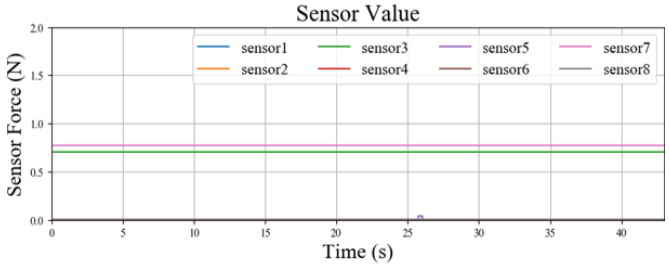
(a) Mask (b) Point cloud (c) Suitable area



(d) States of the robotic arm



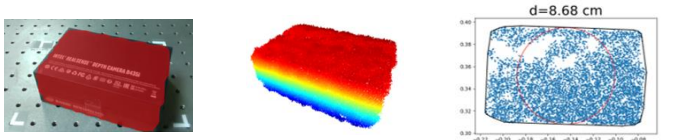
(e) Slip signal



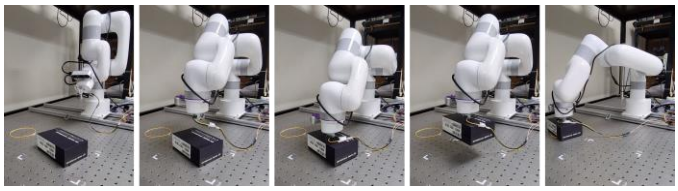
(f) sensor value

Fig. 12. Robot grasping experiment (ball)

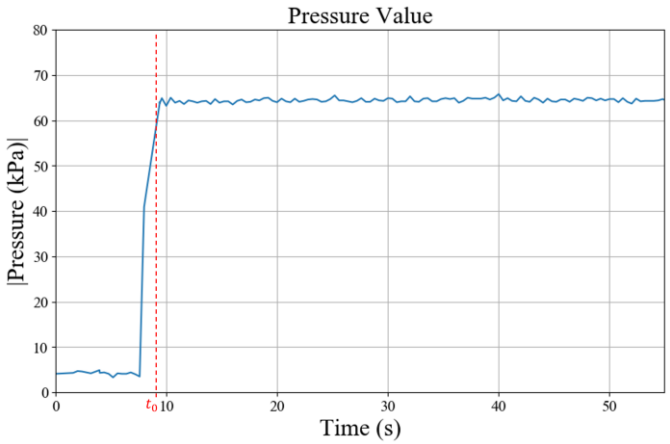
Fig. 13 demonstrates the grasping experiment of a rectangle box, in which Figs. 13(a), 13(b), and 13(c) show the box mask, object point cloud, and the calculated suitable area for vacuum gripper, respectively. As shown in Fig. 13(c), the estimated area of a diameter is 8.68 cm, suitable for the vacuum gripper according to the evaluation in previous section. The states of the robotic arm during the operation are shown in Fig. 13(d). Consequently, a grasp pose was generated by conducting PCA. A vacuum pressure sensor, CFSensor XGZP6847, is integrated onto the vacuum gripper to measure the pressure. As illustrated in Fig. 13(e), when the pressure exceeds -60 kPa as the robotic arm approaching from the way point to the grasping position, it indicates that the gripper has contacted the object. Therefore, the robot will halt its approach and proceed to lift the object. In the experiments, each object was executed on 20 times. The system achieved a success rate of 92% over 100 experiments.



(a) Mask (b) Point cloud (c) Suitable area



(d) States of the robotic arm



(e) Pressure value

Fig. 13. Robot grasping experiment (box)

V. Conclusions

In this work, we have successfully developed an object pick-and-place system for a six-DOF robotic arm that integrates algorithms for object detection, grasp pose estimation, and gripping force compensation. By implementing geodesic distance as a loss function on PointNet and PointNet++ with some dataset, we are able to converge the model with smaller errors of 13.2° and 6.9°, compared to those using Euclidean distance. Furthermore, we propose an approach for manipulating datasets using quaternions to ensure a non-negative real part. This method achieved convergence for both PointNet and PointNet++ with minimal errors of 5.8°, regardless of the employed loss function. Additionally, by implementing slip detection and the proposed force compensation mechanism, the system demonstrates the capability to detect slippages or unstable states and to compensate for gripping force. Lastly, we present the results of grasping experiments on various objects, showing that the system successfully picks up objects without dropping or damage, achieving a success rate of 92% in 100 experiments.

References

[1] A. Mousavian, C. Eppner, and D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Seoul, Korea(South), 2019. pp. 2901-2910.

[2] A. Ten Pas *et al.*, "Grasp pose detection in point clouds," *The International Journal of Robotics Research.*, vol. 36, no. 13-14, pp. 1455-1473, Oct. 2017.

[3] P. Schmidt *et al.*, "Grasping of unknown objects using deep convolutional neural networks based on depth images," *IEEE international conference on robotics and automation*, pp. 6831-6838, 2018.

[4] D. Yang *et al.*, "Robotic grasping through combined image-based grasp proposal and 3d reconstruction," *IEEE International Conference on Robotics and Automation*, pp. 6350-6356, 2021.

[5] J. Mahler *et al.*, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[6] C.H. Wang, and P.C. Lin, "Q-pointnet: Intelligent stacked-objects grasping using a rgbd sensor and a dexterous hand," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pp. 601-606, 2020.

[7] C.R. Qi *et al.*, "PointNet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Honolulu, HI, USA, 2017, pp. 652-660.

[8] K. He *et al.*, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*. Venice, Italy, 2017, pp. 2961-2969.

[9] C.R. Qi *et al.*, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, 2017.

[10] papabravo. Rack & Pinion Robotic Gripper Jaw. [Online]. Available: https://www.thingiverse.com/thing:2661755

[11] Y. Cheng *et al.*, "Data correlation approach for slippage detection in robotic manipulations using tactile sensor array," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2717-2722, 2015.

Hao-En Chang received the M.S. in Power Mechanical Engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2023. His current research interests and publications are in the areas of Robotics, Computer Vision, and Deep learning.

Rongshun Chen received the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, Michigan, USA, in 1992. Dr. Chen is currently a Distinguished Professor in the Department of Power Mechanical Engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan. He is a member of the IEEE Control Systems, Circuit and Systems, and Robotics and Automation Societies. His current research interests are Control Systems, the Applications of Artificial Intelligent, Autonomous Mobile Robots and Multi-robot Systems, Server System Cooling of Data Center, and MEMS.