Robotic Manipulation via the Assisted 3D Point Cloud of an Object in the Bin-Picking Application

Duc Minh Phan
Faculty of Mechanical Engineering
Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street
District 10, HCMC, Vietnam
Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Ha Quang Thinh Ngo
Faculty of Mechanical Engineering
Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street
District 10, HCMC, Vietnam
Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

In the field of robot control, manipulation or object handling is one of the most critical tasks. The existing techniques reveal some challenges such as the unstructured nature of objects or their random orientations within cluttered environments. Our method has emerged as a promising solution, providing detailed spatial information that enhances object detection and pose estimation in this study. Initially, several mechanical computations are carried out to indicate the user-defined tool of robotic end-effector. Then, the image processing techniques, for instance HSV filter, are deployed to identify the center of target object. After that, the coordinates of an object can be obtained using the 3D point cloud data. This information is transmitted to our embedded computer via TCP/IP communication protocol. The outcome of the proposed approach is to properly enable the grasping operation without the human intervention. From these results, it can be seen obviously that our approach is feasible and can be applied in many industrial fields.

Keywords: *3D point cloud, robotic manipulation, visual grasping, motion control, assisted technology.*

1. INTRODUCTION

The industrial application such as bin picking is one of the critical challenges in robotics, including the ability of intelligent robots to autonomously determine, grasp, and handle objects from a dense environment [1-3]. This mission imitates the role of an operator in sorting and retrieving items, and its successful implementation in different manufacturing, logistics or e-commerce.

In the conventional industry, some actions (i.e. gras–ping or placing) [4-6] are usually executed manually. Those missions require more labours prone to manual mistakes, and cause to be fatigue in respect to time. Instead of doing that, the automated industry could considerably lessen labour costs and enhance efficient operation and scalability [7, 8]. However, the bin-picking application contains highly complicated chal–lenges owing to the need for more accurate object iden–tification as well as manipulation in various working conditions.

In the real-world scenario, objects have random orientation, occlusion, or overlapping which becomes the existing barriers for robotic vision system, gripper, and control algorithm [9]. To overcome the above limi–tations, our study aims to address key aspects of robotic problems consisting of designing a robust gripper mec–hanism, integrating image processing algorithms for object detection, and achieving precise robot control [10]. This work not only exposes the technical contri–bution of our knowledge but also is a good chance to solve the scientific and practical problems in the field of robotics and automation.

Received: February 2025, Accepted: April 2025
Correspondence to: Ha Quang Thinh Ngo
Faculty of Mechanical Engineering,
Ho Chi Minh City University of Technology, Vietnam
E-mail: nhqthinh@hcmut.edu.vn
doi: 10.5937/fme2502290P

2. LITERATURE REVIEW

Robots have become indispensable in modern produc–tion, transforming industries with their ability to per–form repetitive and complex tasks with precision, speed, and reliability [11, 12]. They play a pivotal role in improving productivity, reducing costs, and enhancing workplace safety. Key applications of robots in the production industry comprise material handling [13], mechanical assembly [14], or bin picking [15]. In the domain of material handling, robot proceeds the move–ment, sorting, and goods delivery across the production lines. It is possible to manipulate a variety of items from small components to heavy loads with less risk of da–mage during operation.

In assembly tasks, high precision ensures consistent product quality and reduces human error, even if skill-enabled actions are required [16]. In the bin picking system, robot automates the retrieval of randomly ori–ented objects from containers or bins [17]. It is a prin–cipal step in logistics, assembly, and order fulfilment. The advanced control algorithm and modern vision system enable robotic platform to manipulate cluttered or overlapping objects in the practical scenario [18].

To summarize the state-of-the-art researches, Table 1 represents a list of related works in the same domains. According to the categories of applications such as automatic laundry, supportive dressing, surgery, pro–duction, ocean-based manipulation and the others, several key publications are analysed and evaluated in the type of gripper, visual tool, technique(s) and limitation(s).

Table 1. Review of the state-of-the-art publications in related domains.

Application	Author(s)	Publication year	Gripper type	Visual tool	Technique(s)	Limitation(s)
Automatic laundry	Borras, J. et al [19]	2020	Off-the-shell device with small planar fingertips	Multiple-view geometric cues	A novel framework to classify not only grasp types but also the manipulation primitives	A significant gap in the availability of standardized bench-marks and datasets
Supportive dressing	Zhang, F. et al [20]	2019	Wrist-driven parallel-jaw gripper	RGB-D camera, Point Cloud representation	Two-stage policy learning and simulation -free training use only the success/failure feedback to improve without external labels	The method relies heavily on a custom-designed wrist-driven gripper, and extreme geometries of object could reduce success rates.
Surgery	Patel, R. V. et al [21]	2022	End-effectors mimicking surgical tools (forceps, scissors)	Stereo endoscopic camera	Impedance control with haptic feedback modalities is manipulated by sensitive forces at the tool tip	Miniaturization of force sensors is difficult due to surgical tool size constraints and latency and stability challenges in haptic feedback loops
Production	Zhou, Z. et al [22]	2022	Suction cup gripper	Microsoft Kinect camera	Pixel-wise affordance maps and image matching network trained to associate observed object appearances explores multiple grasp directions	Not suitable for soft or porous items and grasping strategies are limited to top-down suction
Ocean-based manipulation	Zarebidoki, M. et al [23]	2024	A parallel-jaw gripper	Overhead RGB-D camera (Intel RealSense SR300)	A Fully Convolutional Network (FCN) fused the pushing-grasping actions which indicate that pushing is prioritized when it increases the predicted grasp success probability	It does not generalize to more complex manipulation like twisting or lifting, and limits applications in cluttered or 3D-enclosed spaces
Others	Yang, B. et al [24]	2020	Multi-functional parallel gripper	RGB-D camera	Unified neural network to perform multiple tasks produces pixel-wise predictions for different action types	Generalization to novel objects and cluttered scenes as well as scalability to more complex sequential tasks still need improvements

3. PRIMITIVE STUDY

3.1 Concept of Bin-picking

A robotic arm which is equipped with sophisticated end-effectors, is crucial for handling and manipulating objects within a bin. It can navigate the bin, determine and pick up objects, and then place them at a specific location. Because of the flexibility and dexterity of robotic manipulator, it allows to handle a wide range of both the physical dimensions and shapes of objects. The usage of industrial robots provides consistent perfor–mance, leverages moving speed, and allows continuous operation without fatigue, significantly enhancing efficient industrial settings. Fig. 1 describes the typical application of bin picking which involves one industrial manipulator, one or two digital cameras, a cluttered bin and belt conveyor.

In such a system, the configuration of a vision-based approach may be either eye-in-hand or eye-to-hand. The first type consists of the directly mounting actuator on the robotic end-effector. It permits the camera to move with the robotic platform providing a dynamic view of the workspace. For the second configuration, our camera is attached on a fixed frame outside the robotic workspace. Owing to this attachment, a stable and wide view of the entire working environment could be reac–hed.

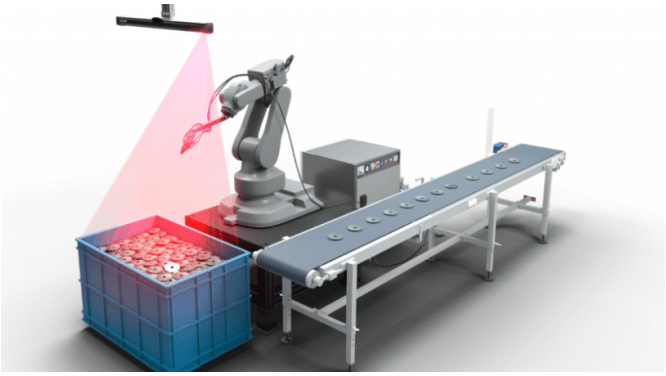


Figure 1. Description of typical concept for bin-picking industry.

3.2 Technical procedure

Commonly, there are five steps, as shown in Fig. 2, to grasp an object in a dense environment such as a bin or container via 3D point cloud data. The first step in bin picking is to identify and locate the objects within the bin/container. Using vision systems (cameras, 3D sensors), the system must distinguish each object and

obtain its positions in 3D workspace. Additionally, it must accurately detect the orientation (angle) of each object to schedule the best grasp, and it must measure the distance from the container to its position and compute its trajectories in 3D workspace.

After computing the position of object, the optimi–zation of extraction path is needed. This step is to search the most efficient and safest path for the robotic arm to reach and pick up that object while minimizing the unexpected movement.

Consequently, robot must navigate around many obstacles in the working environment while moving to retrieve the object. This involves real-time adjustments to avoid collisions with other objects in the bin or sur–rounding areas, ensuring safe and precise movements.

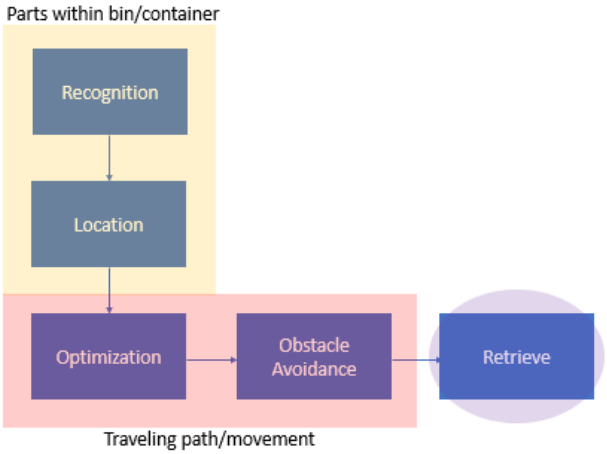


Figure 2. Flowchart of typical technique for bin picking application.

3.3 Handling gripper

In fact, robotic gripper plays an important role in the solution of bin picking application. Several structures of this mechanism are two fingers, three fingers, or human hand-inspired gripper. More degree-of-freedoms (DoFs) are, more flexible motion of the grasping action can be achieved. In Fig. 3, the conceptual design of our gripper is demonstrated. It is the bio-inspired architecture of dual-finger with linear motion, soft contact, and flexible manipulation. In this design, a one-step motor was utilized to drive two fingers, and two sensing devices were attached to the surfaces of the contacts. In the control method of motor, micro-step mode is recom–mended to maintain the highly precise driving signal. Fig. 4 depicts the block diagram of electrical compo–nents for this gripper. It comprises Arduino Uno R3, loadcells, host computer, stepper motor, stepper driver, limit switch and embedded PC. Since it requires DC power, there are two sources such as AC source and DC source.

Arduino: it would play the role of controlling the gripper and receiving feedback signals from load cells.
Embedded PC: this device is responsible for controlling the joints of the robot arm. Additionally, it exchanges signals such as open/close commands for the gripper with the Arduino and identifies whether the gripper has successfully grasped an object or not.

Limit switch: whenever one finger moves to open this gripper, it is essential to attach one sensor to limit its traveling distance.

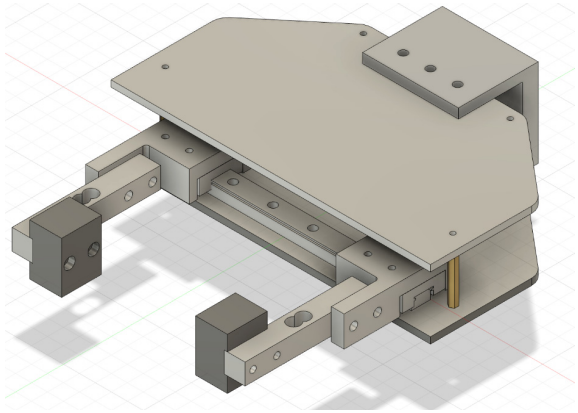


Figure 3. Illustration of the proposed gripper.

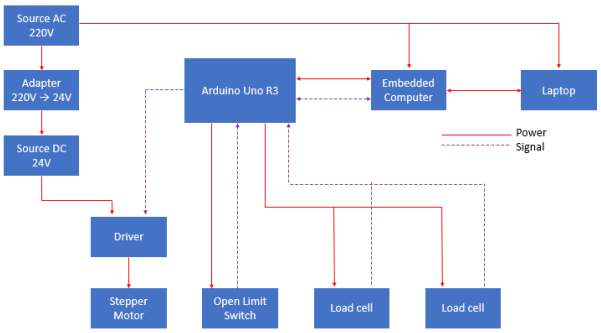


Figure 4. Block diagram of the relative connections bet–ween electrical components.

Host computer: our personal laptop receives images from the camera, then runs the image processing techniques to determine the location of grasping targets. The required positions are then transmitted to the embedded computer via the TCP/IP protocol, and the results of the actions are received back.

Step driver and stepper motor: these components drive the gripper to execute grasping task.

Total power of the electrical gripper:

$$P_{sum} = P_{driver} + P_{source24} + P_{load} \tag{1}$$

With:

P_{sum} : power of electric gripper

P_{load} : power of load

P_{driver} : power of the stepper motor driver,

$P_{driver} = 42W$

$P_{source24}$: power of the power supply 24V,

$P_{source24} = 720W$

$\rightarrow P_{sum} = 42 + 720 = 762W$

Computation of conductor cross-section for 220V supply and branch wires is as below:

Rated current:

$$I_{rated} = \frac{P_{sum}}{U_{rated}.\cos\varphi} = \frac{762}{220.0,8} = 4,33(A) \tag{2}$$

Conductor cross section:

$$S = \frac{I_{rated}}{J_{copper}} = \frac{4.33}{4} = 1.08\left(mm^2\right) \quad (3)$$

With $J_{copper} = 4$ (current density for copper conductors).

The selection of conductor sizes is as follows:

According to the standard tables for conductor cross-sections, our design chooses a 2 mm² conductor for the main power supply to the driver and a 1 mm² conductor for signal branch wires from the Arduino.

3.4 Electrical system of robotic platform

Since there are various electrical devices in our robot, two separated power systems for instance 220V-AC and 24V-DC must have to maintain the full operations. Additionally, one DoF in our robot consists of both servo driver and motor. Hence, 220V-AC power is supplied for five sets of drivers and motors. Fig. 5 describes the schematic diagram of electrical connec–tions for overall devices. In that scheme, dot line indi–cates DC wire for the sensing signal and continuous line represents AC wire for the power supply.

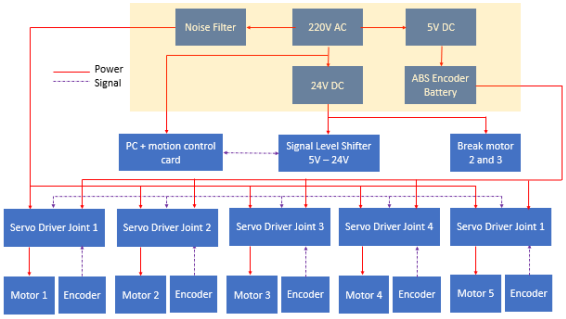


Figure 5. Schematic diagram of the electrical system in our robot.

Computer master: An industrial computer with a PCI slot for motion control cards.

Motion card: A module that generates pulse signals to control the AC servo drive system.

Digital IO card: A module for managing digital IO input and output signals.

Signal conversion circuit: Isolates and boosts signal levels from 5V to 24V.

Buffer circuit: An isolation circuit with output pins for connecting to digital IO devices.

Servopack: A driver for controlling AC servo motors.

5-degree-of-freedom robotic arm: A mechanical structure equipped with AC servo motors.

Computer slave: A computer responsible for processing data from Realsense D435 Stereo cameras.

RealSense camera: Used for human recognition.

3.5 Robotic platform

Most of the work in our investigation is to install five DoF robotic arms. In this platform, there are only rotational movements depending on the turning angles of servo motors. To provide the commanding drive, they are supplied by AC (Alternative Current) power. These servo motors need to be carefully configured to ensure accuracy and compliance with the requirements of manufacturer. Fig. 6 illustrates the architecture of our robotic platform with some parameters between the links and joints.

where:

+ Link length a_i: the length of the common normal of axis (i) and axis (i+1)

+ Link twist α_i angle between the two axes

+ Link offset d_i: distance between the two common normal lines

+ Joint angle θ_i: angle between the two common normal lines

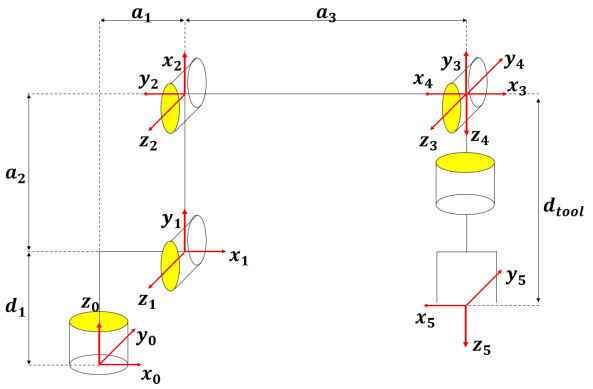


Figure 6. Diagram of the relative joints and links in our robot.

4. OUR APPROACH

Our concepts are to (i) propose a novel framework including two phase: offline and online phase for object detection and pose estimation, (ii) establish the vision-based system with both pre-processing and post-processing techniques and (iii) validate the efficiency and robustness of this method in the practical robotic system. In Fig. 7, there are two phases for our routine of object detection and pose estimation. In the offline phase, the features are extracted from our model. In the online phase, the features extracted from the scene are matched to the model features to finally retrieve the model pose in the scene.

4.1 Point Cloud Pre-processing

This technique is useful for enhancing the quality, efficiency, and feasibility of raw data in various appli–cations, especially visual information, or image. Any sensor like laser scanner or stereo camera regularly releases the unexpected noise because of the environ–mental factors or hardware limitations which can lead to some inaccuracies in manipulating tasks. Consequently, pre-processing procedure guarantees cleaner data while refining the computational efficiency by diminishing the size of dense point cloud, rapidly estimating the theo–retical scheme and less intensive expenditure of resource. Furthermore, it can combine with the other steps such as outlier filtering, coordinate alignment, and standardi–zation, ensuring compatibility with downs–tream tasks like segmentation, registration, or feature extraction, which often require well-structured and reliable data.

- Down sampling: this process is to reduce the rates of density in the point cloud data while retaining its geometric structure and specific features. Additi–onally, it provides two standard methods for down-sampling: voxel down-sampling and uniform down-sampling.

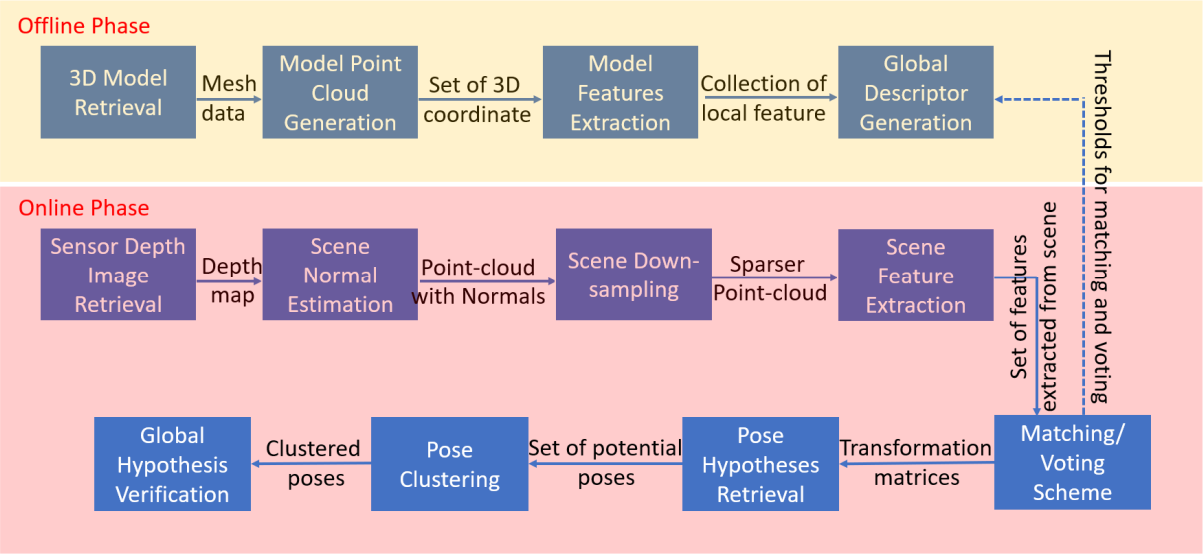


Figure 7. Description of pipeline for object detection and pose estimation.

- Noise reduction: An exception in a point cloud refers to a point that does not imitate to the desired spatial pattern or density of the nearest points. These points are usually caused by the sensing noise, envi–ronmental reflections, or faults in data acquisition. Those exceptions can degrade the quality of the point cloud, leading to inaccuracies in downstream tasks like segmentation, registration, or feature extraction. Hence, it is important to remove them in the pre-processing step to ensure data reliability.
- Evaluation of the normal vector and coordinate normalization: this procedure is to determine which manner the surface at each point is dealing and providing valuable information for the orientational and geometrical surface. Similarly, the process for normalization of a point cloud is to modify the points so that they are centered and fit within a standard size.

4.2 Point Pair Feature Scheme

The Point Pair Feature (PPF) algorithm based on the geometrical approach is powerful for object recognition and pose estimation in 3D workspace. In [25], it surpasses in scenarios involving noise, partially 3D data occlusion. As a result, it is principally valuable in robotics and computer vision applications like grasping, sorting, and scene understanding. This algorithm uses a local descriptor and is extracted from only two points obtained in the 3D data.

In fact, a point pair feature can be achieved from a pair of oriented points. It is supposed that m_1 and m_2 are points in a 3D space, n_1 and n_2 are their corres–ponding normal vectors which represent their orientational surface, and d is the vector between both points (m_1 - m_2). Finally, the feature F is defined as following:

$$F(m_1,m_2)=\left(\|d\|_2,\angle(n_1,d),\angle(n_2,d),\angle(n_1,n_2)\right) \quad (4)$$

where $\angle(n_1,d)$; $\angle(n_2,d)$ and $\angle(n_1,n_2)$: are the angle between vectors n_1 and d; vectors n_2 and d, vectors n_1 and n_2.

4.3 Global model description

In the offline phase, the global model description is constructed to utilize the point pair feature. This process starts by demonstrating the model as a set of point pair features. With similar feature vectors, they are grouped together for efficient matching. The feature vector F is computed to deploy all point pairs $m_i,m_j \in M$ on the model surface.

These computations involve the discretized distances and angles. It is noted that distance d_{dist} and angle $d_{angle}=\frac{2\pi}{n_{angle}}$ are represented while n_{angle} is the number of angles. After that, feature vectors with the identically discretized values are gathered and enabled efficiently during object recognition.

The output of this step is to map the sampled point pair feature space to the corresponding model surface pairs. This mapping is formally defined as:

$$L: Z^4 \rightarrow A \subset M^2 \quad (5)$$

where:

Z^4 : the discretized 4D feature space

A: the set of all point pairs $\left(m_i,m_j\right)\in M^2$ that cor–responds to the feature vector F.

4.4 Voting scheme

In the online phase, a voting scheme is employed to identify the optimal pose of an object by estimating potential matches between scene and model points. This method leverages local coordinates and an efficient accumulator-based strategy. Given the point pair feature F, the rotation angle α is computed as

$$\alpha=\alpha_m-\alpha_s \quad (6)$$

where α_m and α_2 are precomputed angles dependent only on model and scene point pairs, respectively.

For each reference point s_r in the scene, this scheme is proceeded as following:

- Feature matching: Each point s_i in the scene is paired with reference point s_r, and the point pair feature function $F_s(s_r, s_i)$ is com–puted. This feature is matched to the global model description, and saving all model point pairs (m_r, m_i) with similar features.
- Local coordinate computation: For each matched point pair (m_r, m_i), the respecti–vely rotational angle α is identified by using equation (6).
- Vote casting: One vote is completed for the local coordinates (m_r, α). This voting pro–cedure collects evidence for various poses of object based on the relations between scene and model point pairs.

4.5 Pose clustering

As mentioned above, the voting scheme determines potential poses of object. The occurrence of noise, sampling inconsistencies, and occlusions may be displayed in multiple approximated poses. Thus, pose clustering is presented as an additional stage to refine and validate this data.

4.6 Identification of the intrinsic parameters

Basically, digital camera is an electronic device which can capture visual data from the environment. From this captured image, colour data consists of many pixels that each storing the colour of the actual image, usually through three colour values: Red, Green, and Blue (RGB). In our study, the RealSense D435 camera is chosen as the main sensing device for visual data. This camera requires the calibration process which contains a calibration board to predefine patterns that assist our camera detect and correct any discrepancies in its intrinsic parameters for a resolution of 640 x 480 pixels with the form above.

The intrinsic parameter matrix of the camera is typically represented as follows:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{7}$$

where:

f_x, f_y are the focal lengths of the camera in the x and y directions, respectively, measured in pixels.

c_x, c_y are the coordinates of the optical center (also known as the principal point) in the image, typically expressed in pixels.

Table 2. List of the intrinsic parameter for the RealSense D435 camera

Parameter	Value	Unit
f_x	611	px
f_y	610	px
c_x	313	px
c_y	231	px

The exact values for f_x, f_y and c_x, c_y would be provided by the calibration tool as Table 2. These values are critical for ensuring that our camera correctly maps 3D data points to 2D image coordinates, especially for the industrial applications like object detection and robotic navigation.

After completing the calibration, distortions would be automatically eliminated by using API commands in the RealSense library to capture images. This process ensures that any distortion caused by the lens of a camera or other optical factors is corrected in real time and provides accurate and undistorted image data for further processing.

4.7 Identification of the extrinsic parameters

Basically, a set of the extrinsic parameters is repre–sented by the transformation matrix which defines the relationship between the coordinate system of camera and the coordinate system of robot. Furthermore, it al–lows us to convert the position and orientation of objects captured by the camera into the coordinate sys–tem used by the robot for accurate location and mani–pulation. The extrinsic parameter matrix typically takes the following form:

$$P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
$$\begin{bmatrix} x_{robot} \\ y_{robot} \\ z_{robot} \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \\ 1 \end{bmatrix} \tag{8}$$

To determine this transformation matrix, several corresponding points between the coordinate system of robot and the coordinate system of camera need to be established. In the eye-to-hand calibration method, a recognizable pattern, such as a chessboard which can be easily detected by the camera, is utilized. In this rese–arch, its dimensions of 100 x 100 mm attached to a mica sheet are fixed to the end-effector of robot.

Through the known coefficients, for instance dimen–sions of the chessboard, the size of the mica sheet, and forward kinematics, the position of the central point of chessboard could be computed as below:

1) Attaching the chessboard: Locked the chessboard onto the mica sheet and fix it to the robotic end-effector.
2) Using forward kinematics: By utilizing the computational kinematics, the coordinates of the center of the chessboard are determined in the coordinate system of robot.
3) Capturing images: The camera captures multiple images of the chessboard from different positions.
4) Matching points: Identify the corresponding points on the chessboard in both the coordinate system of robot and the coordinate system camera.

The outputs of this process are to compute the rota–tion and translation between the camera and coordinate systems of robot. Certainly, our camera would be loca–ted near the robot in such a way that its field of view covers the entire working area.

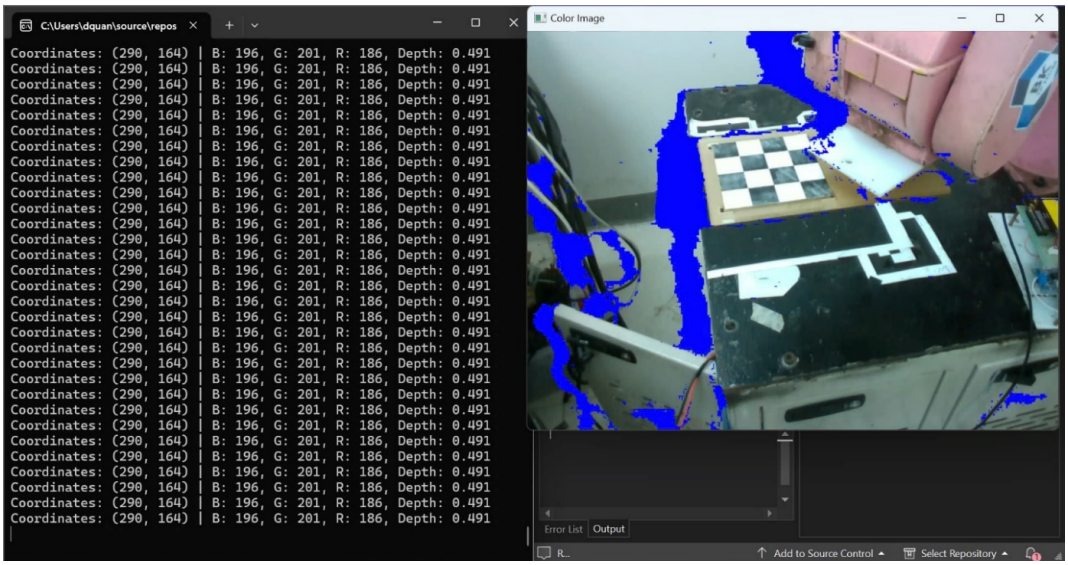


Figure 8. Detection of the central point by manual adjustment.

+ Robot Positioning: The robot is sequentially moved to predetermined positions within the working space. At each position, the camera captures an image using API functions provided by the RealSense library. Our programming language is C++ in the Microsoft Visual Studio environment.
+ Capturing Central Coordinates of Chessboard: After obtaining an image, our next step is to identify the pixel coordinates and the depth value at the center point of the chessboard. The OpenCV library [26] provides a function that can automatically detect the center of the chessboard. However, in some cases, if the image is overexposed or the shooting angle is too steep, our algorithm might fail to detect the center precisely.
+ Manual Adjustment: In case that automatic detection is unsuccessful, the central point of the chessboard must be manually determined as Fig. 8. This is done by moving the mouse pointer to the center of the chessboard in the image and extracting the pixel coordinates and depth values manually.

To enhance the accuracy of calibration data, both automatic and manual method can be combined, despite potential issues with image clarity or orientation. Once the pixel coordinates and depth values are found, they are used in conjunction with the known positions of robot to compute the extrinsic transformation matrix.

After obtaining the pixel coordinates and depth value $\begin{bmatrix} x_{pixel} \\ y_{pixel} \\ d_{depth} \end{bmatrix}$, the point cloud coordinates in the 3D space of the camera $\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \\ 1 \end{bmatrix}$ using the intrinsic parameter matrix and the depth value d_{depth} can be constructed.

$$x_{cam} = \frac{\left(x_{pixel} - c_x\right)}{f_x}.d_{depth} \tag{9}$$

$$y_{cam} = \frac{\left(y_{pixel} - c_y\right)}{f_y}.d_{depth} \tag{10}$$

To find the coefficients of the extrinsic parameter matrix, it needs to be decomposed and solve in each row.

$$P = \begin{bmatrix} P_1 \\ P_2 \\ P_3 \\ 1 \end{bmatrix} \tag{12}$$

For the first row of the extrinsic parameter matrix, we obtain as below:

$$\left[x_{robot}\right] = \left[p_{11} \quad p_{12} \quad p_{13} \quad p_{14}\right].\begin{bmatrix} x_{cam} \\ y_{cam} \\ z_{cam} \\ 1 \end{bmatrix} \tag{13}$$

$$\left[x_{robot}\right] = \left[x_{cam} \quad y_{cam} \quad z_{cam} \quad 1\right].\begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \end{bmatrix} \tag{14}$$

```
Extrinsic Matrix
-0.988166  0.166914 -0.154067  0.573011
 0.258132  0.675378 -0.665424  0.569819
0.0630075 -0.702442 -0.691218   0.51468
        0         0         0         1
Point at Camera:
-0.0926224
-0.0617163
     0.757
         1
Point at Robot Calculate
   0.537608
0.000502325
  0.0289442
          1

Point at Robot Real
0.537
0
0.03

Error
X_Error: 0.00060768
Y_Error: 0.000502325
Z_Error: 0.00105577
XYZ_Error: 0.00131767
```

Figure 9. Result of our computation for the extrinsic matrix.

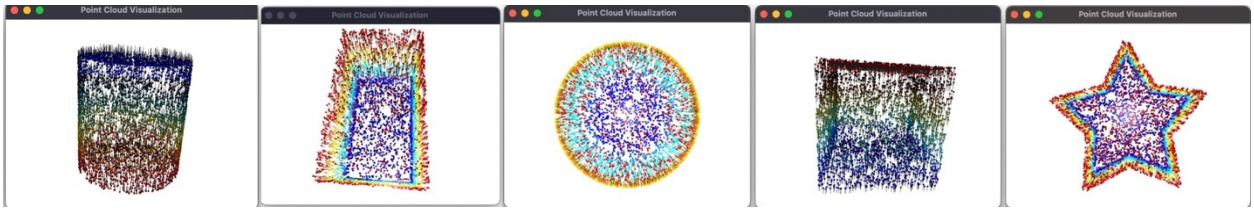


Figure 10. Simulation result of the proposed approach using triangle interpolation.

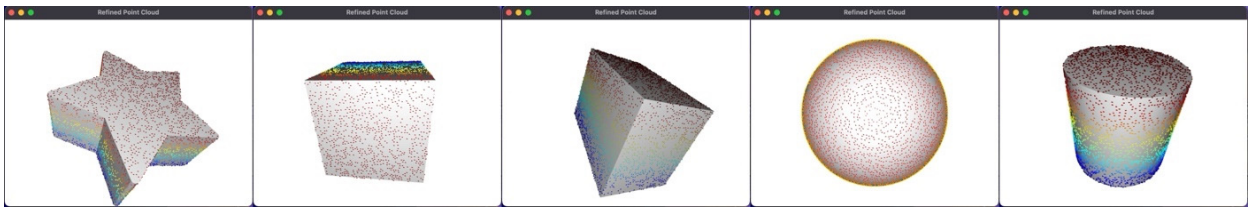


Figure 11. Simulation result of the proposed approach using Poisson disk sampling.

From equation (13) and (14), it consists of four unknowns such as x_{robot}, x_{cam}, y_{com} and z_{cam} thus it is essential to regulate at least four arbitrary points to solve it. However, it might lead to the extrinsic parameter matrix being overfitted if too few points are detected. To prevent this phenomenon, 40 points within the working area are captured.

$$\begin{bmatrix} x_{robot_1} \\ x_{robot_2} \\ \dots \\ x_{robot_{40}} \end{bmatrix} = \begin{bmatrix} x_{cam_1} & y_{cam_1} & z_{cam_1} & 1 \\ x_{cam_2} & y_{cam_2} & z_{cam_2} & 1 \\ \dots & \dots & \dots & 1 \\ x_{cam_{40}} & y_{cam_{40}} & z_{cam_{40}} & 1 \end{bmatrix} . \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \end{bmatrix} \tag{15}$$

$$X_{robot} = X_{cam} . P_1 \tag{16}$$

To solve the system of equations, the pseudoinverse matrix method of X_{cam} would be deployed. In our case, the system is over-identified, and using the pseudo–inverse allows us to find the best-fitting solution. The equation becomes:

$$X_{cam}^{+} . X_{robot} = P_1 \tag{17}$$

The formula for determining the pseudoinverse X_{cam}^{+} is:

$$X_{cam}^{+} = X_{cam}^{T} . \left(X_{cam} . X_{cam}^{T} + \lambda . I \right)^{-1} \tag{18}$$

where λ =0.0001.

x_{robot_i} : location i^{th} of robot.

x_{cam_i} : location i^{th} of camera.

X_{cam}^{T} : pseudoinverse of matrix X_{cam}

After identifying P_1, similar steps to solve for P_2 and P_3 could be performed. The resulting extrinsic para–meter matrix after calibration is:

$$P = \begin{bmatrix} -0.988166 & 0.166914 & -0.154067 & 0.573011 \\ 0.258132 & 0.675378 & -0.665424 & 0.569819 \\ 0.0630075 & -0.702442 & -0.691218 & 0.51468 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19}$$

This matrix represents the calibrated transformation between the coordinate system of camera and the coordinate system of robot, including both rotation and translation. After testing with 10 different points, the maximum average error along the three axes was found to be 1.5 mm.

5. RESULTS OF OUR STUDY

5.1 Model feature extraction

To estimate the pose of an object, the point pair features of the target object need to be known. For this reason, its representation as a computer-based model in 3D space is necessary. In our case, this program can be adapted to work with polygon mesh models.

5.1.1 Point Sampling with Triangle Interpolation

Firstly, a good point cloud representation of a 3D model would be obtained by generating points on the triangle surfaces of the mesh. To distribute all the points over the mesh, a set of n random indices of the triangular surfaces is required, while n is the size of the point cloud that needs to be generated.

Additionally, the quantity of points on a surface must depend on the area of the triangular shape. As a result, these indices are weighted such that smaller triangles have a minor probability. Otherwise, a bad distribution could occur since large or fewer triangles end with an equivalent quantity of generated points.

5.1.2 Poisson Disk Sampling

The Poisson Disk Sampling approach offers a filtered distribution of the points or plays a role as a blue noise distribution. This technology certifies that there are no two points to be closer than a specified minimum distance, resulting in a well-spaced and uniform point cloud. Our implementation utilizes an efficient iterative process to reach this distribution, matching quality, and computational efficiency.

5.1.3 Scene Feature Extraction

To obtain the features from the target object, it is required to have the point cloud of the scene. In our stu–

dy, the Intel Realsense Stereo Camera D435 is deployed to capture the color images to save the RGB information of the scene, and store the depth information in a few array forms.

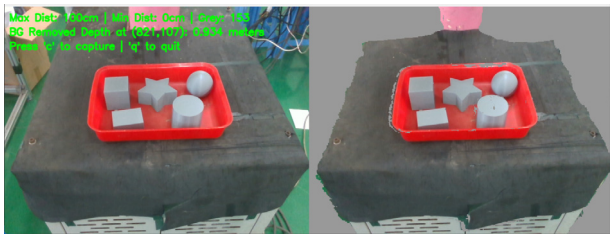


Figure 12. Result of the captured image using Intel Realsense D435.

Figure 13. Result of the 3D point cloud using our approach.

After taking the images as Fig. 12, we have two arrays: one contains RGB colour, and the other image comprises depth information on every pixel. It has been known that in the computational process of extrinsic parameters, the point cloud coordinate in the 3D space of the camera can be identified. Later, normalization of the colour by dividing its value of pixel to 255 is needed because many 3D processing libraries, including Open3D, output the color values to be in the range [0, 1] rather than the standard 8-bit [0, 255] range. To visu–alize the point cloud that was created, these images are saved in a PLY file form which stores the information of x, y, and z coordinates of each point, and information of each point in normal form nx, ny, nz.

5.2 Robotic simulation

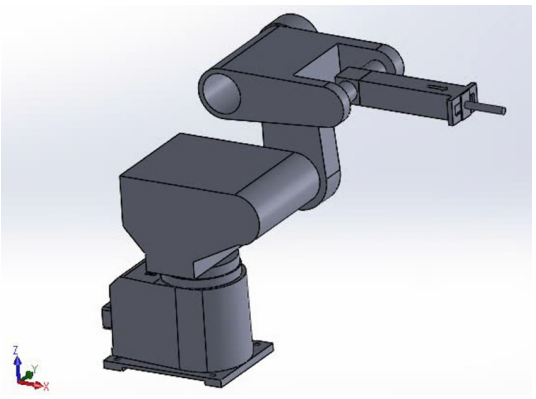
In Fig. 14a, platform of our robot is established in SolidWorks for 3D space. Later, with Simscape add-in, our model is transferred to Matlab for initial settings as Fig. 14b. To simulate its motion, robot model is built in Matlab/Simulink as Fig. 14c for estimating the traveling trajectory. Some illustrations of numerical simulations are completed as Fig. 14d and Fig. 14e respectively.

5.3 Mechanical calibration

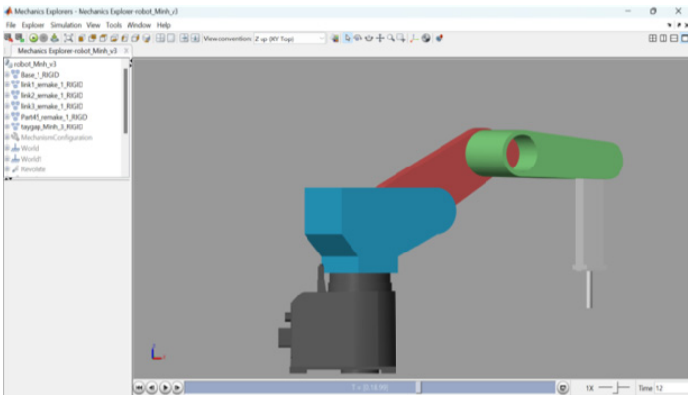
To manipulate the robotic manipulator accurately, mechanical structure of this platform should be consi–dered in both theory and experiment. There are many mechanisms in the body of robot such as gearbox, hard coupling, or mechanical bearing. Therefore, the calib–ration process should be validated to estimate the tracking errors. Typically, target location is represented in x, y, and z coordinate. Several steps for mechanical calibration should be completed. Firstly, hardware configuration is set as Fig. 16 in order to measure error in X direction. This test is repeated 20 times with the same motion command. Each time, data is collected from mechanical indicators. A list of our measurements in the X-axis is gathered in Table 3.

Table 3. List of the measured error in X direction

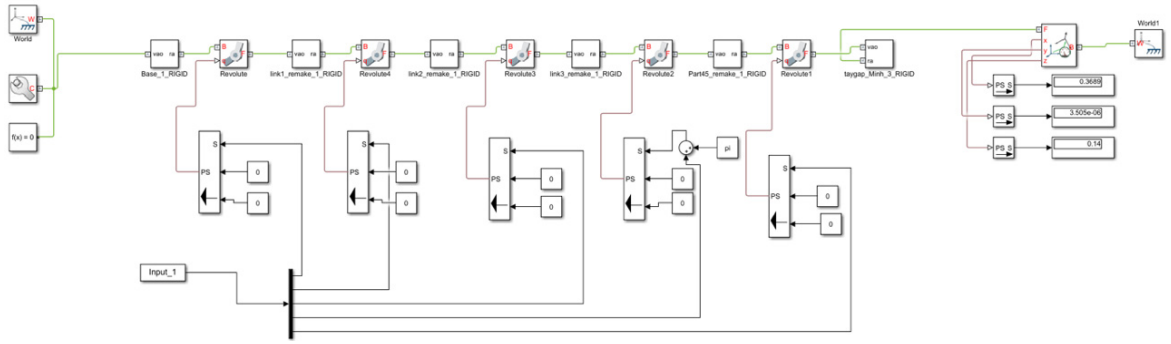
No.	1	2	3	4	5	6	7
Error	0.08	0.05	0.05	0.06	0.05	0.05	0.05
No.	8	9	10	11	12	13	14
Error	0.06	0.04	0.08	0.05	0.07	0.05	0.03
No.	15	16	17	18	19	20	
Error	0.05	0.05	0.03	0.03	0.04	0.04	



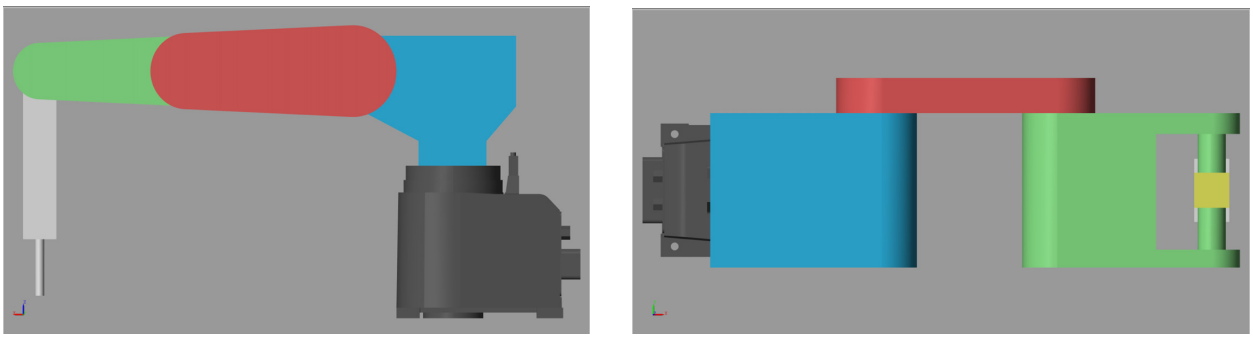
a)



b)

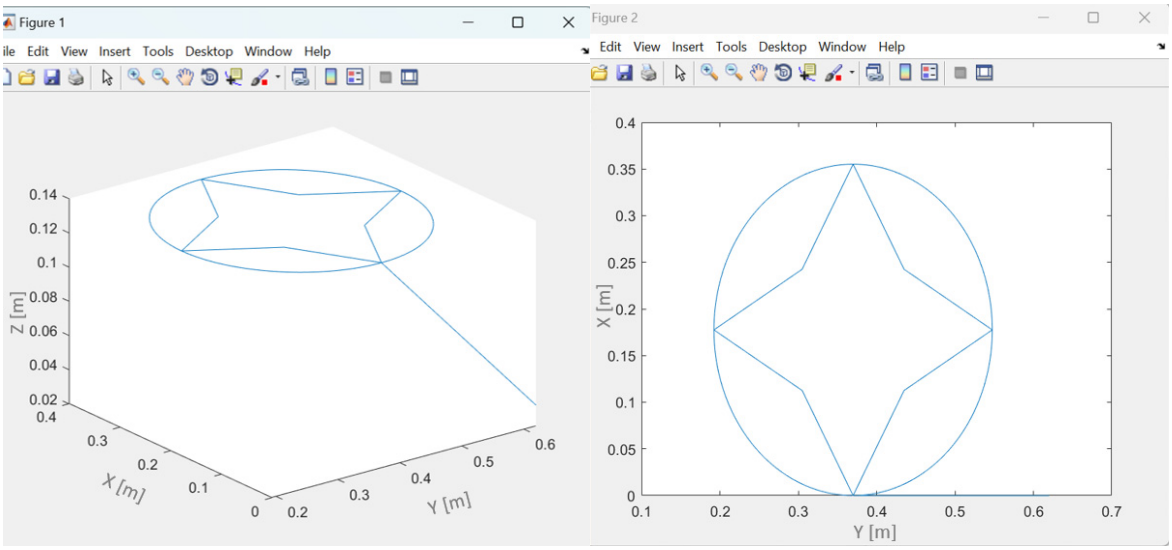


c)



d) e)

Figure 14. Robot 3D model in SolidWorks (a) and in Matlab/Simscape (b), its function blocks in Matlab/Simulink (c), results of simulation in side view (d) and top view (e).



a) b)

Figure 15. Simulation result of the tracking performance for the pre-defined trajectory, (a) 3D view and (b) top view.

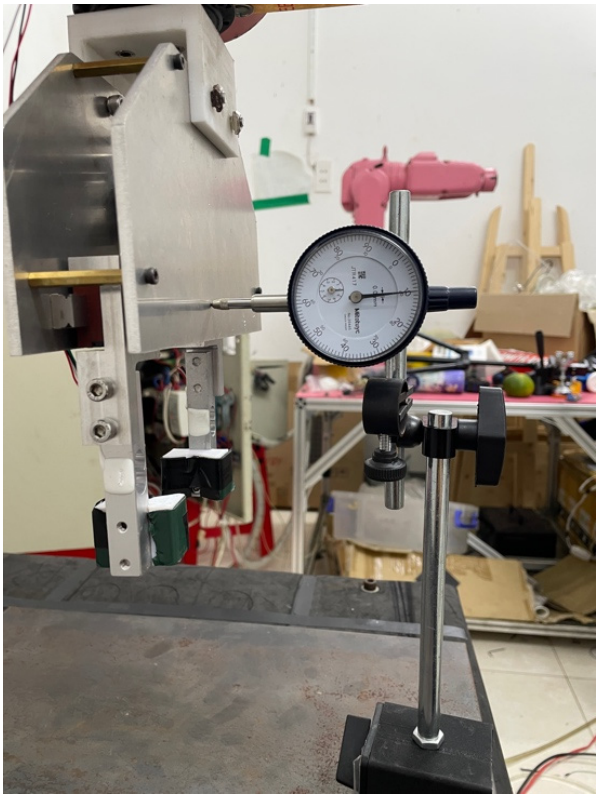


Figure 16. Experimental result of the measured error in X axis.

Similarly, the calibration process in both Y and Z axis is entirely done. The hardware setups for these tests are illustrated as Fig. 17 and Fig. 18 while Table 4 and Table 5 synthesize its results correspondingly. Due to these validations, target object can be evaluated precisely by our control program.

Figure 17. Experimental result of the measured error in Y axis.

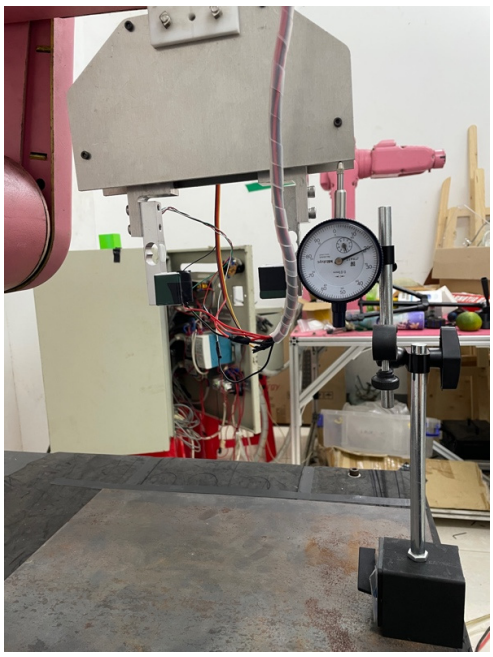


Figure 18. Experimental result of the measured error in Z axis.

Table 4. List of the measured error in Y direction

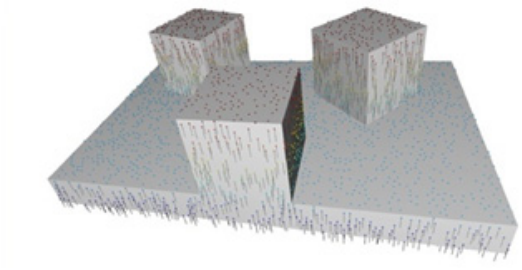
No.	1	2	3	4	5	6	7
Error	0,04	0,03	0,02	0,02	0,05	0,01	0,02
No.	8	9	10	11	12	13	14
Error	0,06	0,06	0,05	0,05	0,05	0,01	0,01
No.	15	16	17	18	19	20	
Error	0,01	0,02	0,01	0,05	0,04	0,04	

Table 5. List of the measured error in Z direction

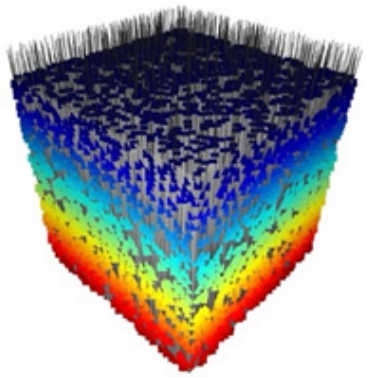
No.	1	2	3	4	5	6	7
Error	0,01	0,01	0,01	0,02	0,02	0,01	0,01
No.	8	9	10	11	12	13	14
Error	0,01	0,01	0,02	0,03	0,01	0,01	0,01
No.	15	16	17	18	19	20	
Error	0,02	0,02	0,01	0,01	0,01	0,01	

5.4 Experimental validation

In this stage, data from digital camera is analysed deeply. In Fig. 19, the practical result is established using our model. Initially, some simple objects, i.e. sha–pes of rectangular or cube object. Owing to the mat–ching scheme as Fig. 20, a virtual model is launched and measured accurately. For more details, location of each point belonged to these objects could be identified and ensured to grasp its in 3D workspace. In the following verifications, more complicated objects with various shapes are suggested to deploy. In Fig. 21, there are five objects including cube, rectangle, cone, star and sphere.



a)



b)

Figure 19. Experimental result of (a) model and (b) scene in 3D point cloud.

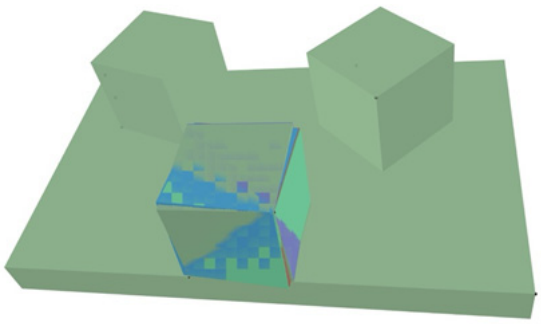
```
Returning 4 clustered and averaged poses
Clustering took 2.2s
Score 5.196152422706632
[[-1.e+00  6.e-02 -3.e-02  5.e+01]
 [ 6.e-02  1.e+00 -6.e-02  0.e+00]
 [ 3.e-02 -6.e-02 -1.e+00  5.e+01]
 [ 0.e+00  0.e+00  0.e+00  1.e+00]]

Score 5.196152422706632
[[ 1.   -0.    0.    0.08]
 [ 0.    1.   -0.    0.32]
 [-0.    0.    1.    0.08]
 [ 0.    0.    0.    1.  ]]

Score 5.196152422706632
[[-1.000e+00 -6.000e-02  3.000e-02  5.085e+01]
 [ 6.000e-02 -1.000e+00  6.000e-02  4.645e+01]
 [ 3.000e-02  6.000e-02  1.000e+00 -2.630e+00]
 [ 0.000e+00  0.000e+00  0.000e+00  1.000e+00]]

Score 3.4641016151377544
[[ 1.00e+00 -7.00e-02 -4.00e-02  3.65e+00]
 [-7.00e-02 -1.00e+00 -7.00e-02  5.31e+01]
 [-3.00e-02  7.00e-02 -1.00e+00  4.79e+01]
 [ 0.00e+00  0.00e+00  0.00e+00  1.00e+00]]
```

(a)



(b)

Figure 20. Experimental result of the matching scheme.

For more details as Fig. 20a, result of pose esti–mation using the proposed algorithm is depicted. The output shows four clustered and averaged 6D poses of the matched object in the scene. Each pose is associated with a confidence score and a 4×4 transformation matrix (both rotation and translation), representing the position and orientation of an object in the camera coordinate of this system. Additionally, the score specifies the confidence level or the matching quality for that pose, based on the accumulated PPF votes. Also, transfor–mation matrix describes the estimated pose: In our result, the top-left 3×3 submatrix characterizes the rota–tion matrix R. The rightmost 3×1 column (excluding the last row) is the translational vector T, given in units corresponding to the scene. And the bottom row [0, 0, 0, 1] is standard in homogeneous transformation matrices.

Figure 21. Illustration of the target objects in our test.

The purpose of our experiment involves picking up an object from bin 1, where each bin contains a single object of varying sizes and shapes, placed randomly. A stereo camera is used to capture an image of the bin. As described in Fig. 22, the point cloud is generated from the image, pre-processed, and analyzed to extract relevant features. Later, our approach can provide robust pose estimation and optimize the handling works for dynamic environments with high accuracy and reliability.

Figure 22. Experimental result of the target objects using our approach.

6. CONCLUSION

Our research introduced a novel concept for robotic manipulation using assisted 3D point cloud techniques. Several preliminary studies were conducted to develop both the hardware platform and technical procedure. The calibration process of the mechanical structure was discussed in detail to ensure the driving manipulation of our system is both accurate and reliable. Vision-based processing technologies were also exemplified throughout the work. To verify the feasibility of our approach, real-world tests were conducted. The results demonstrate that our method is effective and particularly applicable in bin picking tasks, where complex object shapes and cluttered environments pose significant challenges.

The practical applications of this investigation might extend to different scenarios of industrial automation, including warehouse logistics, smart manufacturing, and precise assembly lines, where intelligent robotic manipulation is essential for handling productivity and flexibility. Furthermore, this system can be adapted for service robots operating in unstructured environments (such as in healthcare or domestic settings) where object recognition and manipulation are crucial.

Future research would focus on enhancing the learning capabilities of this system through integrating with reinforcement learning and improving the robustness of object detection in dynamic lighting and occlusion conditions. Additionally, expanding this framework to assist multi-robot collaboration and motion planning in shared workspaces could explore new directions in both industrial and human-robot interactive environments.

ACKNOWLEDGMENT

We acknowledge Ho Chi Minh City University of Technology, (VNU-HCM) for supporting this study.

REFERENCES

[1] Xu, Y., Arai, S., Liu, D., Lin, F., & Kosuge, K. (2022). FPCC: Fast point cloud clustering-based instance segmentation for industrial bin-picking. *Neurocomputing*, *494*, 255-268.

[2] Ngo, H. Q. T. Using an HSV-based Approach for Detecting and Grasping an Object by the Industrial Manipulator System. *FME Transactions*, *Vol. 51, No.* 4, pp. 512-520, 2023.

[3] Cordeiro, A., Rocha, L. F., Costa, C., Costa, P., & Silva, M. F. (2022, April). Bin picking approaches based on deep learning techniques: A state-of-the-art survey. In *2022 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)* (pp. 110-117). IEEE.

[4] Squizzato, S.: *Robot bin picking: 3D pose retrieval based on Point Cloud Library*, Master thesis, Faculty of Engineering, University of Padua, Italia, 2012.

[5] Zhuang, C., Li, S., & Ding, H. (2023). Instance segmentation based 6D pose estimation of industrial objects using point clouds for robotic bin-picking. *Robotics and Computer-Integrated Manufacturing*, *82*, 102541.

[6] Nguyen, T. T., Nguyen, T. H., & Ngo, H. Q. T. Investigation on the Mechanical Design of Robot Gripper for Intelligent Control Using the Low-cost Sensor. *FME Transactions*, *Vol. 52, No. 1*, pp. 12-28, 2024.

[7] Bui, T. H., Son, Y. G., Moon, S. J., Nguyen, Q. H., Rhee, I., Hong, J. Y., & Choi, H. R.: Deep learning based 6-DoF antipodal grasp planning from point cloud in random bin-picking task using single-view, *IEEE Robotics and Automation Letters*, Vol. 8, No. 8, pp. 5196-5203, 2023.

[8] Nguyen, L. P., & Ngo, H. Q. T. Framework Design using the Robotic Augmented Reality for the CyberPhysical System. *FME Transactions*, *Vol. 52, No. 3*, pp. 506-516, 2024.

[9] Yan, W., Xu, Z., Zhou, X., Su, Q., Li, S., & Wu, H. (2020). Fast object pose estimation using adaptive threshold for bin-picking. *IEEE Access*, *8*, 63055-63064.

[10] Zhuang, C., Wang, Z., Zhao, H., & Ding, H. (2021). Semantic part segmentation method based 3D object pose estimation with RGB-D images for

bin-picking. *Robotics and Computer-Integrated Manufacturing*, *68*, 102086.

[11] Fernandes, D. et al. (2021). Point-cloud based 3D object detection and classification methods for self-driving applications: A survey and taxonomy. *Information Fusion*, *68*, 161-191.

[12] Alonso, M., Izaguirre, A., & Graña, M. (2019). Current research trends in robot grasping and bin picking. In *International Joint Conference SOCO'18-CISIS'18-ICEUTE'18: San Sebastián, Spain, June 6-8, 2018 Proceedings 13* (pp. 367-376). Springer International Publishing.

[13] Ding, I. J., & Su, J. L. (2023). Designs of human–robot interaction using depth sensor-based hand gesture communication for smart material-handling robot operations. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, *237*(3), 392-413.

[14] Bashir, A., Moktafi, R., Giangreco, M., & Mueller, R. (2024, November). State Space Exploration with Large Language Models for Human-Robot Cooperation in Mechanical Assembly. In *2024 7th Iberian Robotics Conference (ROBOT)* (pp. 1-6). IEEE.

[15] Kan, Y., Li, H., Chen, Z., Sun, C., Wang, H., & Seidelmann, J. (2024). Recognition and pose estimation method for random bin picking considering incomplete point cloud scene. *Robotic Intelligence and Automation*, *44*(5), 668-680.

[16] Raessa, M., Chen, J. C. Y., Wan, W., & Harada, K. (2020). Human-in-the-loop robotic manipulation planning for collaborative assembly. *IEEE Transactions on Automation Science and Engineering*, *17*(4), 1800-1813.

[17] He, T., Aslam, S., Tong, Z., & Seo, J. (2021). Scooping manipulation via motion control with a two-fingered gripper and its application to bin picking. *IEEE Robotics and Automation Letters*, *6*(4), 6394-6401.

[18] Zhu, G. N., Zeng, Y., Teoh, Y. S., Toh, E., Wong, C. Y., & Chen, I. M. (2022). A bin-picking benchmark for systematic evaluation of robotic-assisted food handling for line production. *IEEE/ASME Transactions on Mechatronics*, *28*(3), 1778-1788.

[19] Borras, J., Alenya, G., & Torras, C. (2020). A gras–ping-centered analysis for cloth manipulation. *IEEE Transactions on Robotics*, *36*(3), 924-936.

[20] Zhang, F., Cully, A., & Demiris, Y. (2019). Probabilistic real-time user posture tracking for personalized robot-assisted dressing. *IEEE Transactions on Robotics*, *35*(4), 873-888.

[21] Patel, R. V., Atashzar, S. F., & Tavakoli, M. (2022). Haptic feedback and force-based teleoperation in surgical robotics. *Proceedings of the IEEE*, *110*(7), 1012-1027.

[22] Zhou, Z., Li, L., Fürsterling, A., Durocher, H. J., Mouridsen, J., & Zhang, X. (2022). Learning-based object detection and localization for a mobile robot manipulator in SME production. *Robotics and Computer-Integrated Manufacturing*, *73*, 102229.

[23] Zarebidoki, M., Dhupia, J. S., Liarokapis, M., & Xu, W. (2024). A cable□driven underwater robotic system for delicate manipulation of marine biology samples. *Journal of Field Robotics*, *41*(8), 2615-2629.

[24] Yang, B., Lancaster, P. E., Srinivasa, S. S., & Smith, J. R. (2020). Benchmarking robot manipulation with the rubik's cube. *IEEE Robotics and Automation Letters*, *5*(2), 2094-2099.

[25] Drost, B., Ulrich, M., Navab, N., & Ilic, S. (2010, June). Model globally, match locally: Efficient and robust 3D object recognition. In *2010 IEEE computer society conference on computer vision and pattern recognition* (pp. 998-1005). Ieee.

[26] Bradski, G. (2008). Learning OpenCV: Computer vision with the OpenCV library. *O'REILLY google schola*, *2*, 334-352.

APPENDIX 1

From above parameters, we can establish the trans–formation matrix at the (i) joint:

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & -\sin\theta_i\sin\alpha & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A1.1)} $$

Hence, the transformation matrices of our robotic system are

$$ {}^{0}_{1}T = \begin{bmatrix} \cos\theta_1 & 0 & -\sin\theta_1 & a_1\cos\theta_1 \\ \sin\theta_1 & 0 & -\cos\theta_1 & a_1\sin\theta_1 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A1.2)} $$

$$ {}^{1}_{2}T = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A1.3)} $$

$$ {}^{2}_{3}T = \begin{bmatrix} \cos\theta_3 & -\sin\theta_3 & 0 & a_3\cos\theta_3 \\ \sin\theta_3 & \cos\theta_3 & 0 & a_3\sin\theta_3 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A1.4)} $$

$$ {}^{3}_{4}T = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 & a_4\cos\theta_4 \\ \sin\theta_4 & 0 & \cos\theta_4 & a_4\sin\theta_4 \\ 0 & -1 & 0 & d_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A1.5)} $$

and

$$ {}^{4}_{5}T = \begin{bmatrix} \cos\theta_5 & -\sin\theta_5 & 0 & 0 \\ \sin\theta_5 & \cos\theta_5 & 0 & 0 \\ 0 & 0 & 1 & d_{tool} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{(A1.6)} $$

And,

Consequently, we have

$$ {}^{0}_{tool}T = {}^{0}_{5}T = {}^{0}_{1}T\,{}^{1}_{2}T\,{}^{2}_{3}T\,{}^{3}_{4}T\,{}^{4}_{5}T \tag{A1.7} $$

where ${}^{0}_{tool}T$: transformation matrix from link 0 to robotic tool

${}^{i-1}_{i}T$: transformation matrix from link i to link $i+1$

РОБОТСКА МАНИПУЛАЦИЈА ПУТЕМ ПОТПОМОГНУТОГ 3Д ОБЛАКА ТАЧАКА ОБЈЕКТА У АПЛИКАЦИЈИ ЗА САКУПЉАЊЕ ИЗ КОНТЕЈНЕРА

Д.М. Фан, Х.К.Т. Нго

У области управљања роботима, манипулација или руковање објектима један је од најкритичнијих задатака. Постојеће технике откривају неке изазове као што су неструктурирана природа објеката или њихове случајне оријентације у претрпаним окружењима. Наша метода се показала као обећавајуће решење, пружајући детаљне просторне информације које побољшавају детекцију објеката и процену положаја у овој студији. У почетку се врши неколико механичких прорачуна како би се назначио кориснички дефинисани алат роботског крајњег ефектора. Затим се примењују технике обраде слика, на пример HSV филтер, да би се идентификовао центар циљног објекта. Након тога, координате објекта могу се добити коришћењем 3Д података облака тачака. Ове информације се преносе на наш уграђени рачунар путем TCP/IP комуникационог протокола. Резултат предложеног приступа је правилно омогућавање операције хватања без људске интервенције. Из ових резултата се јасно види да је наш приступ изводљив и да се може применити у многим индустријским областима.