Automated Bin Picking With 3D Machine Vision

Thoranin Oonariya
Department of Production and Robotics Engineering
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
s6201023621026@email.kmutnb.ac.th

Kolapat Booleua
Department of Production and Robotics Engineering
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
s62010236210130@email.kmutnb.ac.th

Aran Blattler
Department of Production and Robotics Engineering
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
aran.b@eng.kmutnb.ac.th

Jiraphan Inthiam
Department of Instrumentation and Electronics Engineering
King Mongkut's University of Technology North Bangkok
Bangkok, Thailand
jiraphan.i@eng.kmutnb.ac.th

***Abstract* - Labor is used in the industrial packaging process to perform work. The utilization of labor allows for both the possibility of human error and the potential for process inconsistency. This system helps enhance process consistency and reduce mistakes by using a robotic arm manipulator with an RGBD camera. To analyze and control the bin-picking system to perform work in the industry efficiently, each process must be optimized to excel in its respective sectors: perception, planning, and control. The content of this research will be broken down into five key sections: simulation, planning, perception, calibration, and 3D modeling which important to the system. Since bin-picking requires a high-performance and repeatable system, robust development is needed in each process to maximize the performance and repeatability of the operation. As a result, the bin-picking system that has been developed in this research exhibits some biases and has a specific distribution value within pick-and-place process.**

Keywords – RGB-D camera, PCA (Principal Component Analysis), Calibration, ICP (iterative closest point), 3D Modeling, ROS (Robot Operating System)

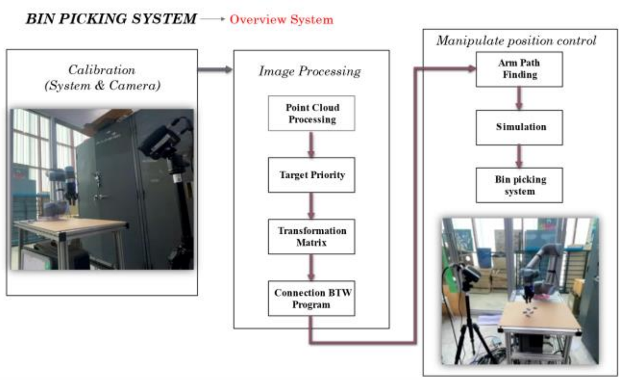

Fig. 1 Hardware system diagram

I. INTRODUCTION

The bin-picking system has been developed over many years, by combining robot manipulators and machine vision [1]. In terms of robot manipulators, there are various types, such as articulated robots, delta robots, SCARA robots, Cartesian robots, and collaborative robots. Each type of robot manipulator has different movements, joints, and workspaces. Collaborative robots have the highest degree of freedom (DOF), making them well-suited for bin-picking systems, but it is challenging to compute the inverse kinematics needed to control each joint link.

In terms of machine vision [2][3][4], there are also numerous types of cameras that can provide depth information [5], such as stereo cameras, Time-of-Flight, Structured Light, and Light Detection Ranging (LiDAR). Each type of depth camera has its own pros and cons. The bin-picking process involves using a depth camera to capture images, which can be 2D or 3D, to calculate the object's position and orientation to control the robot arm. Once the object's position is defined, the planner used to control the robot arm will find the inverse kinematics to reach the target and pick up the object, placing it in the desired destination.

This research sets up an environment that includes simulations in a Gazebo and real environments with actual hardware to perform the task of pick-and-place for right-angle aluminium pieces onto a jig. This paper contains experiments that will demonstrate the accuracy and repeatability of the overall system, using the Universal Robot UR5 as a robot arm manipulator and the Zivid-Two [5] camera as the perception sensor.

II. OVERVIEW SYSTEM

The overall architecture of the bin-picking system proposed in this paper is illustrated in Fig. 1. The Zivid-Two camera serves as a sensor, providing point cloud and RGB data for objects within the workspace. The system output is controlled by the Universal Robot 5 (UR5) with a Robotiq gripper (2F-85). This paper focuses on efficiently processing data from the Zivid-Two camera and its transformation for use by the UR5 robot manipulator.

For research and development purposes, ROS is employed to construct and integrate the bin-picking system, including nodes for obtaining 3D datasets, object positions, and orientations from the camera, as well as nodes for controlling the manipulator's end-effector using Stochastic Trajectory Optimization for Motion Planning (STOMP). Moreover, the object perception module, the input pose estimates the object's position in the workspace. The transformation matrix is calculated for the following relationships: object-to-camera, camera-to-end-effector, end-effector-to-manipulator base, and manipulator base-to-world coordinates. Once the gripper has secured the object, the manipulator can move and place the object in the designated workspace.

A. *Calibration section*

There are 2 steps of calibration to make sure that the system will operate with precision and accuracy. First the camera calibration because the transformation matrix is highly sensitive to the camera's intrinsic matrix (1), so the system must precisely calibrate the camera matrix and camera distortion. Fig. 2 presents the results of the calibration section, the camera calibration using the industrial grade chessboard with to find the camera matrix and camera distortion. In this paper the calibrate board consist of Zivid calibrate board 7x8-30mm and calib.io calibrate board 9x12-30mm to calibrate the Zivid2 camera which provide camera matrix and camera distortion.

$$K = \begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

Second the system calibration. The bin-picking system are composed of many components or device it needs to be calibrated before operating in this case is the position and orientation of the device. To find the transformation matrix (2) that represent the translation and rotation from the RGB-D camera to the robot base link, various methods can be employed to determine the position and orientation of the end-effector.

$$T = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_x \\ R_{21} & R_{22} & R_{23} & T_y \\ R_{31} & R_{32} & R_{33} & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

The ArUco marker is one such method that can define the robot end-effector's pose. To detect the end-effector's position and rotation on the robot arm, the end-effector is attached to an ArUco marker. The system is aware of the transformation matrix controlling the robot arm's position and orientation, which is based on forward kinematics. The calibration program computes the inverse of the transformation matrix and multiplies it with the transformation matrix provided by the ArUco marker. Ultimately, the calibration program obtains the transformation matrix representing the translation and rotation from the RGB-D camera to the robot arm base link. In this paper the ArUco GridBoard consist of 10x14-21.5mm using

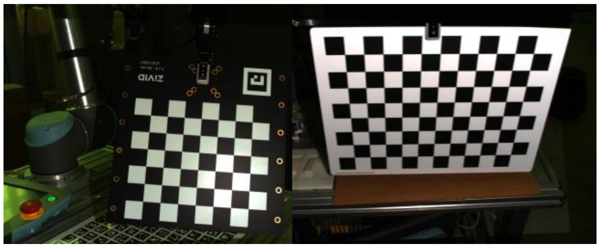

Fig. 2 Zivid calibrate board and calib.io calibrate board

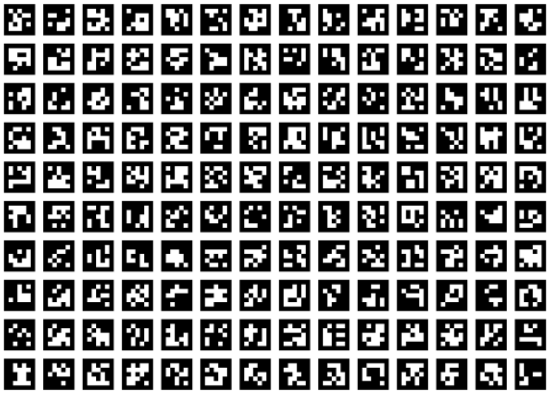

Fig. 3 ArUco 10x14-21.5mm DICT_5X5_250

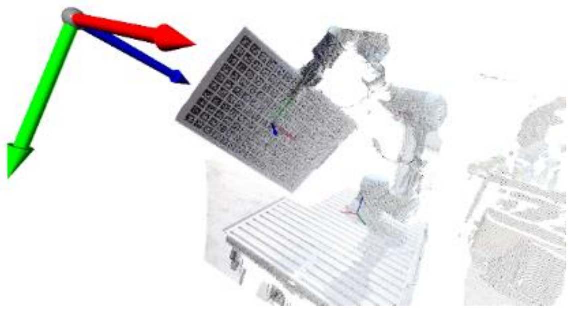

Fig. 4 Calibration section

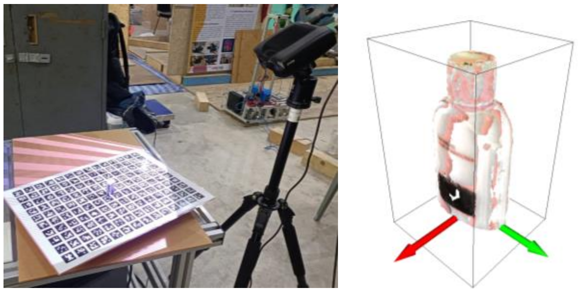

Fig. 5 3D Modeling Process & 3D Pre-define Data

DICT_5X5_250 type as shown in Fig. 3 and Fig. 4. The reason that uses 10x14 ArUco marker is to increase the precision of the coordinate that is provided by the board since the noise from the image processing eliminated using the averaged number of ArUco.

B. 3D modeling section

Since this paper's bin-picking system uses an ICP operation to provide the pose between the target point cloud object and the pre-defined 3D point cloud model, 3D model data must be collected. This can be performed by using an RGBD camera to capture objects that place on the ArUco board from various viewing angles, as shown in Fig. 5. The ArUco board can provide the position and orientation then in every capture the information consist of point cloud data and coordinate of the board that related with the object Afterwards, the pre-defined 3D model data is created by merging the point cloud data from several views into a single model [6] as shown in Fig. 5.

C. Perception section

In the perception section, there are four main steps that provide the object's position and orientation.

First, in the Point Cloud Processing step, the RGB-D image is captured, which contains both RGB and depth data for each pixel. These images are merged to build point cloud data by referencing the camera matrix. The point cloud plane is subsequently trimmed using the RANSAC method, and the DBSCAN clustering algorithm is employed to separate each object.

Second, objects are prioritized by comparing the density of point cloud data and the distance from the camera for each group of points.

Third, after clustering and prioritizing each group of point clouds, the position and orientation of each group must be computed. This is done using 2 method. First method is Principal Component Analysis (PCA) to find the mean and eigenvectors of the point cloud group, which represent the position and orientation relative to the camera. Second method is Iterative Closest Point by using this method between pre-define 3D model that assign the coordinate as same as RGBD camera coordinate and the object point clouds in the workspace which provide the transformation matrix that represent the translation and rotation between the camera and target pose.

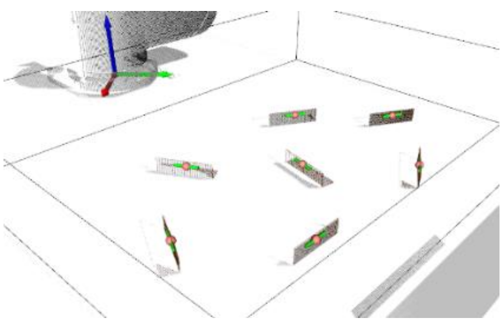

Fig. 6 Perception section

Fig. 7 Simulation in Gazebo

Finally, the pose reference is changed from the camera to the robot base link by multiplying the transformation matrix representing the distance and rotation between the camera and the robot arm. This transformation matrix can be obtained from the calibration process. In Fig. 6 shows the result of the perception section. And

D. Planning section

The manipulator's trajectory to reach the object pose is calculated using a planning method. By accessing MoveIt through the Rviz plug-in and connecting the planning process to the Stochastic Trajectory Optimization for Motion Planning (STOMP) library, a smooth, collision-free path is generated within a reasonable timeframe. STOMP relies on generating noisy trajectories and can include functions to control torque limits, energy, and tool constraints. It can also find the inverse kinematics needed to move the manipulator to the desired position and orientation.

E. Simulation section

Since bin-picking involves developing multiple components, problems not related to programming may arise if the development process is conducted using real hardware and real environments. Gazebo simulation can be used to simulate the environment for testing the bin-picking system's program. The information provided by Gazebo, such as RGB images, depth images, point clouds, and joint states of the simulated robot arm, is conveyed through ROS messages. The developed program utilizes this information to compute the object's position and orientation and the robot arm's trajectory to complete the pick-and-place process. Fig. 7 shows the simulation of the proposed system consist of Ur5 Robot Arm, table workspace, and the object.

III. Experimental Results

In the experimental result, A depth camera, specifically the Zivid-Two, was used to capture point cloud data and calculate the object's position and orientation. The planning node then calculated and controlled the robot arm to move the object, which in this case was an eraser, to the jig. The real workspace has been set as shown in Fig. 9. With the propose process the Bin-picking system can evaluate the coordinate of the target object in the workspace as shown in Fig. 8

The system's error was measured by using a graph paper grid with a size of 1 mm to define the place position offset from the target position as shown in Fig. 10. The overall design was tested 550 times, divided into six scenarios with three angles (45°, 90°, and 135°), two distances (80 cm and 100 cm), and 200 random positions, summarized in TABLE I. We rounded up the values to the nearest integers and created a histogram. From the histogram that is presented in Fig. 11, we observed that there is a bias in the final position data, with a shift towards negative values, approximately -6 mm.

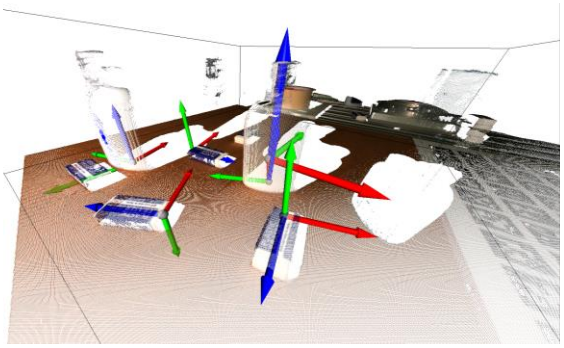

Fig. 8 Pose estimate of target objects

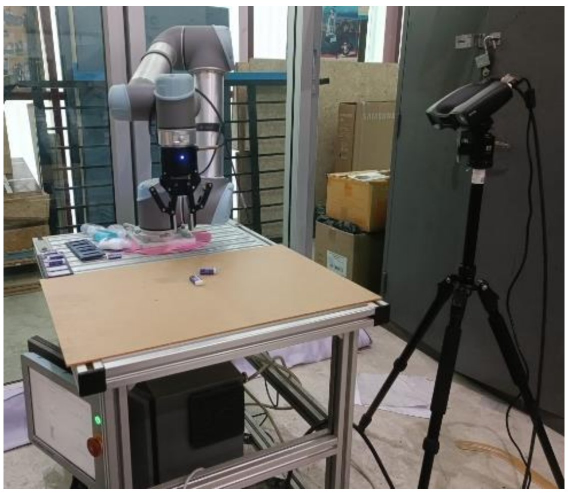

Fig. 9 Real environment of the proposed system

Fig. 10 Measurement error with a graph paper grid

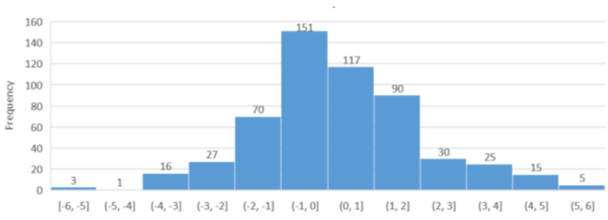

Fig. 11 Data histogram

TABLE I

Testing (Times)	Median (mm.)	Mean (mm.)	Standard Deviation (mm)
550	0.275	0.526	1.759

IV. Discussions & Conclusions

The bin-picking system is a versatile system that can employ various solutions and methods for automatic operation. In perception, the bin-picking system can utilize 2D images or 3D point clouds to compute the position and orientation of objects in the process, or even use both types of data for object analysis. In planning, numerous planners with different calculation methods were selected to be compatible with the system.

For future work, this experiment only used 3D point clouds to compute object positions and worked exclusively with objects that were placed separately from one another. Our aim is to implement multi-method approaches for computing object positions and separating objects that lay on top of each other which can use AI (Artificial intelligence) to detection or segment the object that overlay on each other before process the pose-estimation. Additionally, we plan to refine the placement solution to reduce errors. After placing the object, reducing the vacuum surface gripper's contact area could potentially decrease error values.

Acknowledgment

The authors would like to express our deepest gratitude to our advisors, colleagues, members of the iRAP robot club, and researchers of the Center of Innovative Robotics and Advanced Precision Systems: iRAPs for their invaluable guidance, support, and contributions to this research. Their expertise and insights have been crucial to the successful completion of this paper. Additionally, we appreciate the hardware system support provided by the Department of Instrumentation and Electronics Engineering, Faculty of Engineering, KMUTNB and Abiz Technology Co., Ltd., which enabled us to conduct this research.

References

[1] C.-C. Wong, C.-Y. Tsai, R.-J. Chen, S.-Y. Chien, Y.-H. Yang, S.-W. Wong, and C.-A. Yeh, "Generic Development of Bin Pick-and-Place System Based on Robot Operating System," IEEE Access, vol. 10, pp. 65257-65270, 2022.

[2] F. -L. Markley, Y. Cheng, J. -L. Crassidis and Y. Oshman,"Averaging Quaternions," Journal of Guidance, Control and Dynamics, vol. 30, No. 4, pp.1193-1196, 2007

[3] S. S. Mirkamali and P. Nagabhushan, "RGBD image segmentation," 2015 9th Iranian Conference on Machine Vision and Image Processing (MVIP), Tehran, Iran, 2015, pp. 41-44.

[4] T. Zheng, C. Chen, J. Yuan, B. Li and K. Ren, "PointCloud Saliency Maps," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019, pp. 1598-1606.

[5] Zivid, "Zivid Camera Datasheet," [Online]. Available: https://www.zivid.com/hubfs/files/SPEC/Zivid%20Two%20Datasheet.pdf

[6] N. Pudchuen and W. Jitviriya"3-D Mapping and Localization using RGB-D Camera," 2019 First International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)