Robotics HW4[1]

R12921008 Che-Jung Chuang
R12922135 Cheng-Yen Yu
R11631045 Wei-Hsuan Cheng
B10901042 Wen Perng

1 Problem Design Rationale

The procedure of the solution is as follows: after camera calibration, the robot arm first take a picture of the three blocks and find their centroid and principle axis. Then, the arm goes down to grasp the blocks one by one, hence completing the task (not the bonus though).

Some things related to coordinate transformations are shown as below.

1.1 Image to Base Relation

To obtain the camera calibration from images of known block placements, we need the following regression program.

First we extend camera matrix to homogeneous form 4×4 and denote it as $\mathsf{CameraMat}$. The position of a block in base coordinates, gripper coordinates and image coordinates are denoted as $\mathsf{obj\_base}$, $\mathsf{obj\_gripper}$ and $\mathsf{obj\_img}$, respectively. Under the pinhole camera model, we can write the projection relation from base coordinates to image coordinates as:

$$\begin{bmatrix} \mathsf{obj\_img\_w} \times \mathsf{obj\_img\_d} \\ \mathsf{obj\_img\_h} \times \mathsf{obj\_img\_d} \\ \mathsf{obj\_img\_d} \\ 1 \end{bmatrix} = \mathsf{CameraMat} \times T_{\text{gripper to camera}} \times T_{\text{base to gripper}} \times \begin{bmatrix} \mathsf{obj\_base\_x} \\ \mathsf{obj\_base\_y} \\ \mathsf{obj\_base\_z} \\ 1 \end{bmatrix},$$

where $\mathsf{obj\_img\_w}$ and $\mathsf{obj\_img\_h}$ are the position of the block object in image coordinates. $\mathsf{obj\_img\_d}$ is the unknown depth of the image with respect to the camera. But we know that the area of the object in image is disproportional to the square of the image depth, i.e., we have

$$\mathsf{obj\_area} \propto \mathsf{obj\_img\_d}^{-2} \Longrightarrow \mathsf{obj\_img\_d} = \alpha \cdot \mathsf{obj\_area}^{-0.5}$$

where α is the constant of proportionality. Note that α is block dependent. Then

$$\begin{aligned} \begin{bmatrix} \mathsf{obj\_img\_w} \times \alpha \cdot \mathsf{obj\_area}^{-0.5} \\ \mathsf{obj\_img\_h} \times \alpha \cdot \mathsf{obj\_area}^{-0.5} \\ \alpha \cdot \mathsf{obj\_area}^{-0.5} \\ 1 \end{bmatrix} &= \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mathsf{obj\_img\_w} \times \mathsf{obj\_area}^{-0.5} \\ \mathsf{obj\_img\_h} \times \mathsf{obj\_area}^{-0.5} \\ \mathsf{obj\_area}^{-0.5} \\ 1 \end{bmatrix} \\ &= \mathsf{CameraMat} \times T_{\text{gripper to camera}} \times T_{\text{base to gripper}} \times \begin{bmatrix} \mathsf{obj\_base\_x} \\ \mathsf{obj\_base\_y} \\ \mathsf{obj\_base\_z} \\ 1 \end{bmatrix}. \end{aligned}$$

[1]Code provided in github: eps46656/Robotics-AssignmentIV, and wei-hsuan-cheng/assignments_robotics_2023-24.

Next, merge α's, $\mathsf{CameraMat}$ and $T_{\text{gripper to camera}}$ into a new matrix $\mathsf{obj\_M}$. Then we have

$$T_{\text{base to gripper}} \times \begin{bmatrix} \mathsf{obj\_base\_x} \\ \mathsf{obj\_base\_y} \\ \mathsf{obj\_base\_z} \\ 1 \end{bmatrix} = \mathsf{obj\_M} \begin{bmatrix} \mathsf{obj\_img\_w} \times \text{obj\_area}^{-0.5} \\ \mathsf{obj\_img\_h} \times \text{obj\_area}^{-0.5} \\ \mathsf{obj\_area}^{-0.5} \\ 1 \end{bmatrix},$$

where

$$\mathsf{obj\_M} = T^{-1}_{\text{gripper to camera}} \times \text{CameraMat}^{-1} \times \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \alpha & 0 & 0 \\ 0 & 0 & \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Regress to acquire $\mathsf{obj\_M}$. The regression process needs $N \geq 4$ cases to work. Each case contains: 1. a captured image, 2. the transformation from base coordinates to gripper coordinates when the image is captured, 3. object locations in base coordinates.

When inferring the pose needed for the robot arm to move to, $T_{\text{base to gripper}}$ and $\mathsf{obj\_M}$ shall be known. While $\mathsf{obj\_img}$ and $\mathsf{obj\_area}$ can also be detected from the image taken. Then:

$$\begin{bmatrix} \mathsf{obj\_base\_x} \\ \mathsf{obj\_base\_y} \\ \mathsf{obj\_base\_z} \\ 1 \end{bmatrix} = T^{-1}_{\text{base to gripper}} \times \mathsf{obj\_M} \times \begin{bmatrix} \mathsf{obj\_img\_w} \times \mathsf{obj\_area}^{-0.5} \\ \mathsf{obj\_img\_h} \times \mathsf{obj\_area}^{-0.5} \\ \mathsf{obj\_area}^{-0.5} \\ 1 \end{bmatrix}$$

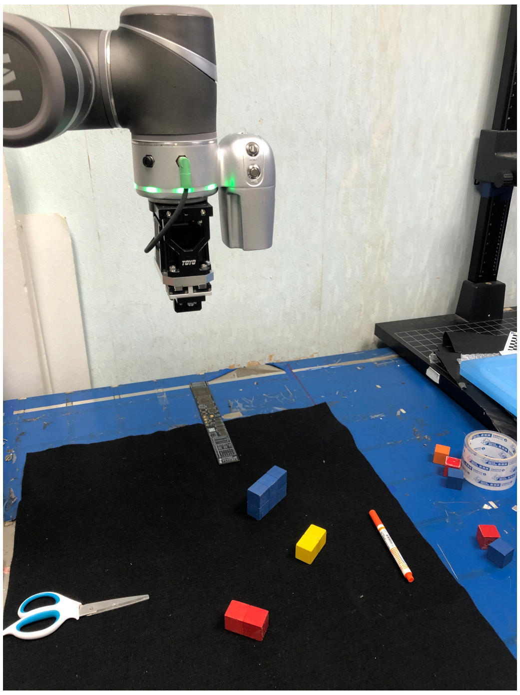

Figure 1: Brick Grasping Experiment using TM5-900 Robot Arm. Video HERE.

2 Challenges Faced and Solutions

1. *The camera were not able to capture all the blocks in a single picture initially.*

   We hence opted for a higher initial pose for the camera, giving it a wider view to capture all the blocks.

2. *The movement interpolation provided by the TM robot arm made it easy to nudge bricks away from their initial position.*

   Instead of going point-to-point for the whole trajectory, we set up waiting points above the bricks and use linear motion downwards to where the bricks are at. We were able to make the trajectory of the robot arm not collide with the bricks.

3. *Even though all the codes seem to be correct, the tasks always fail midway due to minor errors in distance.*

   The solution is by the good ol' trial and error.

Division of Work

Che-Jung Chuang:	Trajectory Planning, ROS2 System Setup
Cheng-Yen Yu:	Robot Vision
Wei-Hsuan Cheng:	Trajectory Planning
Wen Perng:	Report