Robotics: Assignment IV

Robot Arm Manipulation

Biomechatronics Engineering

Li-Wei Yang (B07611001), Cheng-Yen Chung (B07611048), Andrew Tay Jian An (B07611024), Hua-Ta Liang (B07611038)

Division of work:

Transform matrix calculation:

Li-Wei, Cheng-Yen.

Gripping strategy:

Andrew, Hua-Ta.

Instructions

So far you have learnt the kinematics of robot arm and some basic image processing techniques to detect objects from images. In this assignment, you are asked to use these knowledges to develop a program that allows the robot arm to pick up blocks. Given three randomly placed blocks, your program should locate these objects through TMRobot's eye in hand camera and pick them up one by one. To gain higher grade, try to stack up these blocks.

Following are some issues that need to be considered while doing this assignment:

(1) The relationship of the coordinate systems between the camera and robot arm base

(2) The pose of the end-effector is described as (X, Y, Z, A, B, C). (X, Y, Z) is the end-effector position in Cartesian coordinate. However, what do (A, B, C) stand for? Try to manipulate the arm with the software to figure it out. (Hint: A, B, and C represent a set of Euler angles. Please indicate their rotation axes and order.)

(3) You have to do object detection and calculate the centroid and principal angle as in assignment III (b).

A basic python code for connecting the robot arm is provided. To process images, you may also need OpenCV. You could use C/C++, but you'll have to handle the connection by yourself. Should you encounter any technical problems, please solve them by discussing with your team members or TAs, and write the solution(s) in your report. Gradings will be based on the TA's evaluation on your demo and report.

Warning:

(1) Be sure to check the safety rules and obey them.
(2) Please have a hand on the emergency stop button at all times.
(3) Please contact TAs immediately if encountered any collisions resulting robot arms to stop.
(4) Please contact TAs immediately if robot arm is damaged during usage.
(5) Please DO NOT attempt to fix the problem addressed above by yourself.
(6) Any team who violates the rules above will receive a penalty of grade deduction. (including arm collisions and damages.)
(7) Fail to notify TAs regarding damages to the robot arms will result in 0 grade of the assignment.

End-effector position definition

The pose of the end-effector is described as (X, Y, Z, A, B, C). (X, Y, Z) is the end-effector position in Cartesian coordinate relative to the arm base. (A, B, C) stands for ZYX Euler angles. The world frame first rotates along Z axis for A, then rotate B along the new Y axis for B, then rotates along the new X axis for C to get the same orientation as end-effector frame.

Design rationale

1. Scaling factor

By finding all counters of the cube, we could calculate all of the distances in the camera frame.

$$\Delta x_c^2 \ + \ \Delta y_c^2 = \ D_c, \text{ unit: pixel}$$

In the meantime, we measured all of the actual distances in mm by ruler in the world frame, D_w.
We then calculated the scaling value by:

$$S = D_w / D_c, \text{ unit: mm/pixel}$$

After we got scaling value from different distances, we calculated the average to be our scaling factor.

2. . Camera frame to world frame transformation

Excluding Z, by formula:

$$\begin{bmatrix} X_{w1} & X_{w2} & X_{w3} \\ Y_{w1} & Y_{w2} & Y_{w3} \\ 1 & 1 & 1 \end{bmatrix} = T \begin{bmatrix} X_{c1} & X_{c2} & X_{c3} \\ Y_{c1} & Y_{c2} & Y_{c3} \\ 1 & 1 & 1 \end{bmatrix}$$

We can find the transformation matrix by three sets of corresponding (Xc, Yc), (Xw, Yw).
So first we need three pose including both by camera frame representation and world frame ones.
Therefore, we did the steps below:

(1). Move the arm to the word frame pose (Xw, Yw, 150) close to the table.
(2). Put the cube to the pose where the end effector is, the cube is now at (Xw, Yw).
(3). Move the arm to world frame pose (230, 230, 530) and take a picture, the image_sub would calculate the centroid of cube and get (Xc, Yc).
(4). Repeat step 1-3 until three sets of coordinates are obtained.

Using three coordinate sets and the scaling factor we previously got, we substitute them into the previous formula and by matrix multiplication we can get T.

3. Gripping Strategy

With the transformation matrix, we could map the position from image (pixel) to the real world coordinate (mm). Using principle angle calculation from hw3 with the transformation, we are able to locate the position and orientation required by the end-effector.
Our gripping strategy is to first hover on top of the target with targeted orientation (Z: 200), move down to catch (Z: 100), moving up without x, y drift. Then we hover to the target location (Xw: 350, Yw: 350) and release the cube. Repeat the same steps as catching, down, up, hover, only with release height add an object height (25). By doing so, we are able to accomplish the task of assignment.

Problems

Homogeneous coordinate and scaling factor
At first, we scaled entire homogeneous coordinate by the scale, the matrix we got is erroneous.

Solutions

Homogeneous coordinate and scaling factor
We found out that if we scale the entire homogeneous coordinate, the scale in the homogeneous coordinate itself would change. Thus, later we only scale Xc, Yc and got the correct transform matrix.

Demo

https://youtu.be/l3mNQjmSfdA