ARC 380 / CEE 380 – Introduction to Robotics for Digital Fabrication
Mondays & Wednesdays 11:00am-12:30pm | Arch Bldg N107 / ECL
Princeton University | School of Architecture | Spring 2024

Professor: Arash Adel | Assistant-in-Instruction: Daniel Ruan

Assignment 5: Disk Sorting

Due: Tuesday, April 2, 11:59pm

Instructions:

In this assignment, you will work together with your group and program the robot to perform various 2D perception and sorting tasks. Document the process and note the deliverables for each part.

Git pull the latest files from https://github.com/ADRLaboratory/arc380_s24.git. You will need the latest 'docker-compose.yml' and helper Python code file under the Assignment5 folder.

Part 0 – Prerequisites (0 pts)

We have Intel RealSense D435i cameras set up for the workcell, one for each robot. The helper Python code includes a function that connects to the camera and captures an image. You do not need to change any of the code in the helper file, but you can look through it to understand how it works. To use this function in your own code, make sure to import it:

```
from ARC380_Assignment5_helper import capture_img
```

You will also need the RealSense SDK and Python bindings installed. For Windows, download the installer at the following link (you only need the 'Python 3.7 Developer Package'), as well as install the bindings using `pip install pyrealsense2`.

https://github.com/IntelRealSense/librealsense/releases/download/v2.54.2/Intel.RealSense.SDK-WIN10-2.54.2.5684.exe

Unfortunately, we were unable to get the SDK working for macOS – we have set up a Windows desktop computer for one of the robots with the SDK, VS Code, and Docker so that you can transfer your code and run everything on that machine.

There is no template starter code for this assignment. Feel free to utilize code from previous assignments and workshops to accomplish each part.

Part 1 – Calibration and Teaching (2 pts total)

a. Calibrate a teaching tool (e.g., a pencil mounted in the pen-holder end-effector).

b. Use your calibrated tool to teach the positions of the ArUco markers that define the boundary of the task space.

Part 2 – Feature Extraction (8 pts)

a. Pick a selection of 10 colored acrylic pieces to work with throughout the assignment. Your collection should include 3-4 different colors and at least 2 different sizes/shapes within each color. At least one of your pieces should be square.
b. Arrange your collection of acrylic pieces within the task space and capture an image.
c. Create functions to process this image and extract the following information for each object. This information can be stored in a data structure of your choosing (e.g., tuple, dict, class, etc.).
    i. Color
    ii. Shape (i.e., circle or square)
    iii. Size
    iv. Position (in the world frame)
    v. Orientation (if it is a square)
d. Annotate the image with your extracted information (e.g., using OpenCV functions such as `cv2.circle`, `cv2.line`, `cv2.putText`).

Deliverable: Annotated image with the extracted features for each of the acrylic pieces in your collection.

Part 3 – Pick and Place Sorting (10 pts)

a. Create function(s) that pick or place the acrylic pieces using COMPAS_RRC given a target pose. Take note of the following:
    - The vacuum gripper end-effector has already been defined and calibrated for each robot. It is named 'vac_gripper'.
    - Use `compass_rrc.SetDigital` to turn the gripper on or off. The name of the signal is 'DO00'.
    - Be sure to take the thickness of the acrylic (1/8 in.) into account. (Don't crash the robot.)
    - The best practice for pick and place is to approach and retract the end effector perpendicularly to the workpiece (so that you don't potentially collide with anything when coming in at an angle). This means breaking your trajectory into several steps:
        i. Go to offset point above target
        ii. Linear down to target point
        iii. Grip/release object
        iv. Linear up to offset point above target
    - You might need to add a short delay (`compass_rrc.WaitTime`) to allow for the vacuum to properly engage/disengage.
b. Create function(s) that sort your collection into piles. Each pile should be the same color, and smaller pieces should go on top of larger pieces. Squares should be rotated so that their sides align with the world axes.
    - Remember to account for the thickness of the acrylic as you stack!
c. Run your sorting with the robot three times, each with a different randomized initial configuration of your collection. Document the start and end images for each trial. Record the sorting process for each trial.

Deliverables: Images of the starting configuration of each sort trial and the final sorted stacks, and a video recordings of the sorting process.

Submission guidelines: Create a folder named "ARC380_Assignment5_GroupLastNames" (e.g., ARC380_Assignment5_AdelRuan). Put your code and specific deliverables for each part in this folder. Zip the main folder and upload it to Canvas for submission. Each group only needs one person to submit their assignment.