16662 Autonomy Project Report

Empty Container

Group 16: Atharv Sathe, Trung Nguyen, Zhekai(Scott) Jin

Date: May 4th, 2020

Motivation

In today's fast-moving production environments, picking and packing operations demand a lot from human operators including uninterrupted speed, reliability, inspection, sorting, accuracy, and dexterity. In these repetitive tasks, robots become superior and cheap with the fact that they can complete these tasks consistently at high speed without the need for breaks.

We have seen picking and placing robots taking place at tasks like wholesale distribution, assembly, medical application, and even tasks in hazardous environments. In this final project, we have chosen the empty container task to validate and strengthen our understanding of the intelligent robot for picking and placing solutions.

Key Challenges

The aim of our project was to tackle real-life scenarios and challenges observed for grasping objects by a robotic arm. The most important requirement is to perform a successful grasp irrespective of the orientation of the object. We devised our problem statement to achieve a successful grasp by using vision feedback and implementing Q-learning algorithm for cases where cameras are not available.

In the processes of implementing the said methods, we planned to take into consideration some special cases that would make the system more robust. One such case is when the object is too close to the edge of the bin, the gripper is not able to perform a successful grasp. The challenge was to push the object away from the edge and then perform the grasping maneuver. This idea can be extended to a scenario when two objects are in the close vicinity of each other. Another challenge was vision occlusion due to the position of the external camera. Some parts of the robot blocked the view of the environment. These were the key challenges that we tackled during the course of our project.

Overview

Vision-based approach

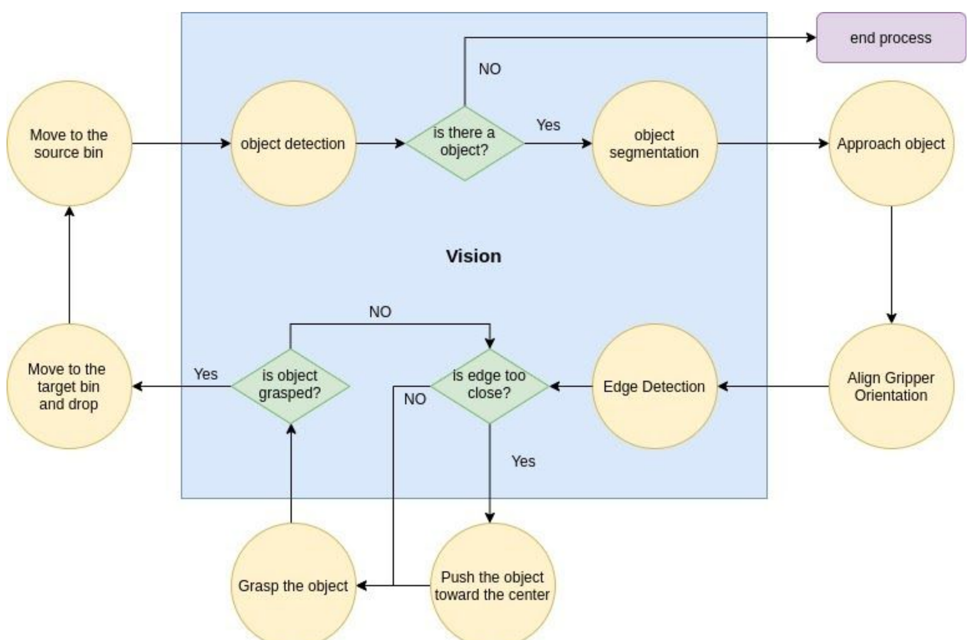
In this section, we present an overall overview of the solutions we implemented to solve the picking and placing task. Firstly, we obtain the noisy pose information about the source and target bins from the pose server. Then, we tried to locate the object using the cameras and if the object is detected to be too close to the edge, we pushed it toward the middle of the container to make the gripping task easier. And then, we obtain the orientation of the object and tried to grasp it. By confirming the gripping state by depth information, we then move the arm and object to the target bin and release the gripper. We repeat the above operations until there is no more object in the source bin and we then reset the environment by interchange the definition of source and target bins.

Reinforcement Learning approach

We attempted to use Q-Learning to train the robot to learn to move close to the objects and grasp it to the target bin. Our action space is the change of the joint angle at 1 degree in any of 7 joints as well as the change of state in the gripper. For each joint, the action can either be -1, 0, or 1 degree in change. For the gripper, it is either closed (0) or open (1). Therefore, the size of our action space is 3^7x2 = 4374 for each state. In addition, our state space is the joint angle in degrees of all joints as well as the state of the gripper.

Methods

Vision-based approach



-
- Figure 0: the overall pipeline of the vision based approach

We implemented the vision based approach following the above pipeline and each individual step are explained in the following sections. Note that the reset is implemented with the same logic but interchanging source and target bins, we simply explain the forward logic.

1. Moving to bin

We accessed the position of the bins from the noisy pose sensor and moved the arm above the bin to get a good vantage point of all the objects placed in the bin.

2. Object detection and segmentation

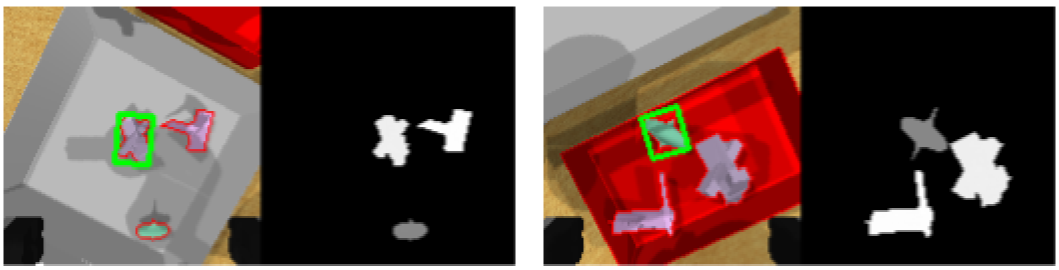
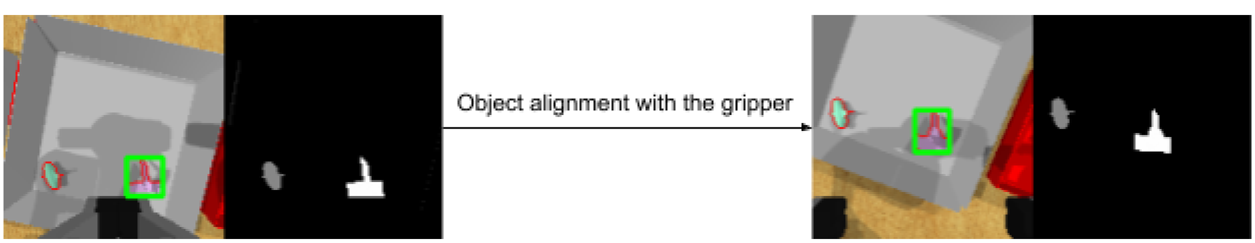


Figure: Object segmentation at the two bins. Red lines are the contours and green rectangle is the bounding box of the closest object to the center of the image view.

After the robot moves to the vantage point of the bin, it uses object segmentation based on colors and only captures small objects close to the center view in the image view and ignores segments that are too big or close to edges of the view. In the above figure, the robot is able to filter just only the objects (grayscale images). If there are found objects, the robot will move the next stage. If not, the whole process is completed

After object detection, the robot finds the contour of the objects (red lines on the left images), and locate the one closest to the center of the image view to pick up and draw a bounding box for it.

3. Aligning gripper to the object and grasp



With the bounding box of the object from the previous step, the robot will first move close to the bounding box so that the center of the bounding box is at the center of the image view. Then the gripper rotates so that the bounding box is vertically aligned with the image view as in the figure above. The gripper will do final adjustment if the box is not at the center after the rotation, then it executes the grasping from above to the object. After grasping, the gripper pulls back to the vintage point and checks whether the object is in the gripper using depth images. If the object is in the gripper, it moves to the target bin location and drops the object. If the grasping fails, it will retry again with the horizontal alignment of the bounding box to the gripper and repeats the cycle of alignments until it can grasp it.

4. Container edge detection

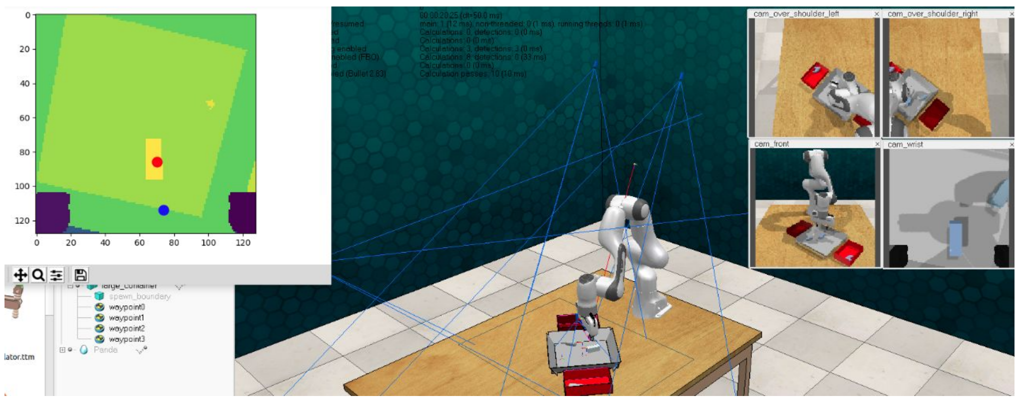


Figure 3. Picking process with segmentation mask shown

We tried to detect the edge of the container firstly using canny edge detection and shape detection, but the performance is poor with the fact that the container edge is often occluded when the arm is moving around. Then, we tried to use the segmentation mask provided by the RL bench and use breadth-first search from the object center location to identify the closest edge around it, with its orientation relative to the object. And if the squared pixel-wise euclidean distance is too large, we will then enter the gripper state to push the object into the middle, which is explained in the next section.

5.Pushing objects at edges

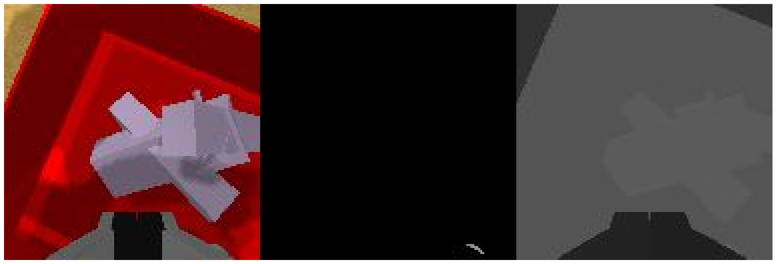


Figure 4: Object is close to the edge and starts the push maneuver.

An edge detection algorithm detects the edges of the object and the bin. It calculates the minimum distance between the edges in pixel values and also the direction of the edge with respect to the object. If this distance is less than the programmed threshold, the arm performs the push maneuver. During this motion, the arm moves towards the object as mentioned above but instead of grasping, it just places its gripper on the object. Once this is achieved, the joint torques and increases to put normal force on the object. Then the arm is slowly moved in the direction opposite to that of the edge thus moving the object away from the edge.

Reinforcement learning approach

Because the reward function of RLBench is very sparse that it only gives a reward of 1 when the whole task is complete. The robot cannot learn anything useful with it. As a result, we had to implement a custom reward function that gives the robot a reward for every action so that it would learn better. Our learning criteria is

- Distance between the objects and the target bin.
- Distance between the gripper and the objects.
- Desired orientation of the gripper.
- Gripper state (reward if the gripper is closed).
- Reward for object in gripper.

With these criteria and Q-Learning, the robot is able to learn to move close to the object that is closest to the target bin. However, it stucks at that location without being able to properly grasp the object probably due to hitting the local maxima point.

Demo

The demo consists of a total set reset cycle performed by the arm using the visual feedback. One cycle takes approximately 10 mins to complete. The robot is able to successfully grasp objects from the larger bin in the first attempt for almost 90% with a successful grasp within 3 attempts. However, it faces difficulties while grasping from the smaller bin as the objects are very near to each other. Here the success rate in the first attempt drops down to 50% with a successful grasp usually within 8 attempts.
The second demonstration is of the arm trying to pick up objects using Q learning. However, the robot gets stuck in local minima and is not able to pick the objects.
The final component of the project is arm pushing the object away from the edge. The success rate of the arm pushing down on the object is approximately 70%. When the arm pushes down on the object, half of the time it just nudges the object while other times it is able to successfully drag the object away from the bin edge. This is integrated into the vision-based demo.

Future Work

- As currently, our robot can only move close to an object with Q-Learning, our next goal is to remodel our reward function so that the robot can grasp the object and move to the target bin to drop it
- When the object is near the edge, the arm tries to push it away. However, sometimes it is not able to reach the object as the bin edge comes in collision with the gripper. A mechanism to orient the gripper ideally parallel to edge should be implemented to avoid the collision. Another solution is to approach the object from the inner side of the bin to avoid the edges all together.
- The arm tries to drag the object away from the edge once it applies the normal force. But sometimes the object slips away and is not dragged successfully. Implementing a feedback loop to check if the object is getting pushed successfully will result in a more robust push.

  We could also use vision sensor to detect the bin location which is currently obtained from the pose server to make the solution purely vision-based. This could be implemented by leveraging multiple object and shape detection methods to find the largest rectangular shape in the segmentation hierarchy.