Autonomous Pick and Place Grasping Application Using Touch and Visual Sensing

Hyungjoo Kim ucabhki@ucl.ac.uk
KaKei Choi ucabkkc@ucl.ac.uk
Youssef Al Jrab ucabljr@ucl.ac.uk

Abstract— **Pick and Place robots are commonly being used in industry nowadays in several areas to achieve faster and more efficient outcomes. This paper goes through a problem setup in which a Pick and Place robot is used alongside other technologies to complete a task in hand. The task involves picking between two distinct objects and placing on three available tables. The task focuses on the decision making process the robot has to go through when placing the objects at the correct tables. The technologies involve using an RGB-D camera for object detection as well as bump sensors on the tables to keep track of certain states. The paper goes through the methodology and algorithm of the robotic system, as well as the evaluation of the experiments performed by the system.**

I. INTRODUCTION

Autonomous Pick and Place (P&P) tasks are heavily being performed in several industrial applications in this age. Those applications include manufacturing on assembly lines, packaging, anomaly inspection and handling, storage in warehouses, etc. P&P applications are usually performed by robotic manipulators with multiple degrees of freedom. This area of industry needs such robots to increase the output, speed, and efficiency of such tasks. Robots are known for performing repetitive tasks at a consistent rate which is why they are relied on this area of work.

This paper first discusses the problem setup for P&P grasping applications. The setup consists of three tables, a box object, a cylinder object, and a robotic arm manipulator. The main task is to pick the cylinder using the manipulator and place it on a specified table by providing the respective command. There are three commands where each one directly relates to picking the cylinder from its current position and placing it to the respective table. The two objects can't be on the same table at the same time. This implies that whenever a P&P command wants to take place to the desired table, the manipulator needs to make sure that the table is either free or that it needs to free it first from the box object, by moving it first to another free table, before proceeding with moving the cylinder object to the desired table. The manipulator always needs to place the objects at the center of the tables. The box object initially starts at the center of table 2, whereas the cylinder object starts at the ground next to the robot.

The project tackles two tasks to be performed for the problem setup. The first task consists of performing visual sensing for the cylinder that is placed initially on the ground. Visual sensing for using an RGB-D camera is required to accurately detect the cylinder before grasping for using the filtering methods. After successful detection and grasping of the cylinder, the manipulator should pick it up from the ground and wait for corresponding P&P commands. This is when the second task comes into play, which consists of waiting for P&P commands before performing what is required of the robot to properly place the objects in the correct locations. Bumper sensors are being utilized to keep track of object locations by knowing which table is free and which is occupied. This paper discusses the technologies used more thoroughly and showcases some of the previous work done using these technologies.

II. LITERATURE REVIEW

In this section, the paper explores methodologies and results of some of the previous work done in relation to pick and place robots.

P&P robots require object position and orientation estimation before picking which is described in this paper [1]. The paper focuses on how the pose of the object affects the planning process of the manipulator to grasp the object. Moreover, several methods have been developed in the area of image processing, where one of them utilizes feature extraction algorithms for detection. The work presented in paper [2] shows how feature extraction work by first converting raw images to grayscale images, then converting them to binary images using Otsu's method. After that the process includes smoothing, finding gradients, edge tracking, image dilation, and image filling to which then the objects detected are bounded by rectangles and fed into classifiers for object classification. This paper [3] focuses on object detection and 3D pose estimation from single images. The 3D model is extracted only using the object's 2D silhouette and projected in 3D to estimate its position. Furthermore, some work has been done on computing and planning the motion of a two-handed manipulator for pick and place tasks. The work done in paper [4] focuses on grasp planning and path planning. Grasp planning includes calculation of the grasp frame, set of contact points, and hand configuration. The idea is taken further to implement double grasp planning using two hand manipulators.

III. Methodology

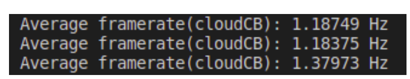
The problem setup has been described in the introduction of this paper where it showcased having three tables surrounding the robotic arm manipulator. The whole robotic system operates on ROS, where the objects and tables are rendered as collision objects using the MoveIt library. The system utilizes several filters to increase its framerate and perform faster. Using a pass-through filter leads to low frequency values, thus the system tries implementing a voxel-grid filter or a combination of pass-through and voxel-grid filters to increase performance. The voxel-grid filters are classified as down-sampling filters because they reduce the number of points in the cloud. This means down-sampling the space of the points in the cloud to an average value. The robot consists of a gripper to grasp the required objects and also consists of an RGB-D camera that is used only once on the cylinder object after the first command to properly detect the object and grasp it before continuing to perform corresponding actions. This only applies when giving commands to place the cylinder object on free tables. If a command at the beginning chooses to place the cylinder object at table-2 which is occupied, the robot must first free table-2 by picking the box object and placing it on another table, described by the algorithm, then the robot can proceed to detecting the cylinder object and grasping it to then place it on table-2. All objects should be placed on the middle of the tables where bump sensors are allocated to keep track of relevant information.

At the beginning, the cylinder object is placed on the ground. The RGB-D camera mounted on the robot searches for the object and tries to completely detect it by collecting depth data. Once completely detected, the manipulator approaches the object and grasps it to then place it on one of the tables. From now on, the RGB-D camera is no longer used, all decision are being made based on the locations of the two objects which are being tracked by the bump sensors and some logic. Two variables are created, one to keep track of the box object location and the other for the cylinder object location. The box object is located initially in the center of table-2, so its location variable is initially set to table-2. Robot commands are given initially before the object detection step so that the robot knows which actions to perform first before picking and placing the objects. The initial location of the cylinder object depends on the first command given to the robot. The algorithm of picking and placing the objects using the bumper sensors is described as follows:

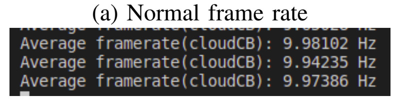
Algorithm 1 Pick and place under constraints

Initialization←the pose of a cylinder/box object
while *Check Available tables* **do**
    If a distance between the objects and sensor is less than 0.2m, the sensor is activated
    if *bumper1 is activated* **then**
        if *bumper2 is activated* **then**
            Pick the box object and place it on table 3
        else
            Pick the box object and place it on table 2
        end
    end
    if *bumper2 is activated* **then**
        if *bumper1 is activated* **then**
            Pick the box object and place it on table 3
        else
            Pick the box object and place it on table 1
        end
    end
    if *bumper3 is activated* **then**
        if *bumper2 is activated* **then**
            Pick the box object and place it on table 1
        else
            Pick the box object and place it on table 2
        end
    end
    Pick the cylinder object and
    Place it on table 1 ← '1' is pressed *or*
    Place it on table 2 ← '2' is pressed *or*
    Place it on table 3 ← '3' is pressed
end

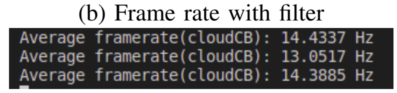
IV. Experiments



(a) Normal frame rate

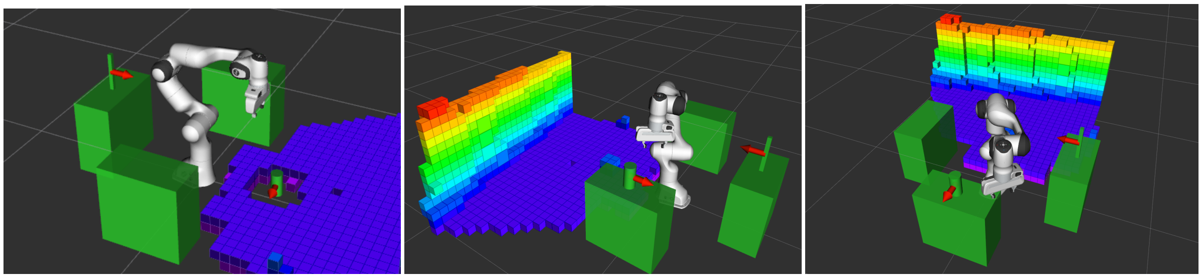


(b) Frame rate with filter



(c) Frame rate with fast-filter

Fig. 1: Frame rate for (a) normal (b) filter (c) fast-filter



(a) Publish the pose of objects (b) Pick and place without constraint (c) Pick and place under constraints

Fig. 2: Strategies for picking and placing the objects with/without constraints

Several experiments have been conducted with the robotic setup to evaluate the overall performance. Fig.1 showcases the average frame rate of the system under different conditions. Under no filters, the average frame rate reaches values around 1.2 Hz. After adding a filter, the average frame rate increases to reach values around 9 Hz which is a significant improvement from before. However, after adding a fast-filter, the average frame rate increases more to reach a values around 14 Hz. Using the fast-filter will increase the performance of the system and increase its speed.

```
bump1:0 bump2:1 bump3:0
```

(a) Bumper status of Fig. 2-(a)

```
bump1:1 bump2:1 bump3:0
```

(b) Bumper status of Fig. 2-(b)

```
bump1:0 bump2:1 bump3:1
```

(c) Bumper status of Fig. 2-(c)

Fig. 3: Check the updated bumper sensors

Fig.2 and Fig.3 describe the experiments of picking and placing the objects with and without constraints. Fig.2-a shows the initial position of the objects, where the box object is on table-2 whereas the cylinder object is on the ground. This is confirmed by the bump sensors, where Fig.3-a shows that only bumper-2 is activated. Fig.2-b shows the robot performing a P&P action on the cylinder object to table 1 where there are no constraints of having another object on that table. Fig.3-b confirms that, where bumper-2 stays activated as the initial case, but now bumper-1 is also activated since the cylinder object is occupying table-1. Fig.2-c shows the robot performing a P&P action on the cylinder object to table-2 where there is a constraint of having the box object occupying that table. The robot performs well and realizes the constraint. It first picks the box object from table-2 and places it on table-3, the free table. Then, it proceeds on picking the cylinder object from table-1 and places it on table-2. Fig.3-c confirms that, where bumper-3 is now activated due to the box object, bumper-2 is activated since the cylinder object is occupying table-2 now, and bumper-1 isn't activated since it's free of the objects. Therefore, all results can prove to work on the same principle as Algorithm 1, which generates reasonable and successful outcomes.

V. Conclusion

In this paper, we proposed a robotic system capable of performing Pick and Place tasks within an environment consisting of two distinct objects to pick and three tables of where the objects are being placed. The system consists of a robotic arm manipulator for Pick and Place grasping, an RGB-D camera for object detection, and bumper sensors on the tables to keep track of table states and object locations. The system performs based on the methodology proposed in the paper. Experiments have been performed to better assess and evaluate the robotic system based on the environment given.

Future work on this project focuses on increasing the complexity of the problem at hand. This could be done by increasing the number of tables and increasing the number of objects which requires much better algorithms and better decision-making strategies to perform the required task. Algorithms are needed to develop that can detect objects faster and accurately by studying better filtering techniques of point cloud library, since some would focus on having the fastest response by the searching environment. Furthermore, the complexity could be further increased by adding more robots that are performing Pick and Place tasks simultaneously to the same set of tables and objects which requires the robots to work collaboratively.

References

[1] P. A. S. Rawat, "Pick and place industrial robot controller with computer vision," 2016.

[2] R. K. S. L. S. K. P. Chand, "Object detection and recognition for a pick and place robot," 2014.

[3] M. Z. K. G. D. Y. Y. S. B. M. Z. C. P. M. L. K. Daniilidis, "Single image 3d object detection and pose estimation for grasping," 2014.

[4] J.-P. S. M. G. J. C. D. S. T. Siméon, "Planning pick-and-place tasks with two-hand regrasping," 2010.