Autonomous Pick and Place Grasping Task, using RGB-D Camera and Bumper Sensing

Mohamed Shire, Hou Nam Chiang, Renjie Zhou

I. INTRODUCTION

Robotic research has consistently been seeking for improved efficiency and productivity at a lower cost. Pick and place has been an age-old problem in robotics, where a robot grasps, places and then ungrasps an object. Automating the pick and place problem with accuracy and efficiency frees up manual labour for other more complex tasks, increasing the productivity [1].

However, the environment in real life is always complex. Apart from the object of interest, many other objects are usually on scene, resulting in potential collision with the robot when performing the pick-and-place task. Hence visual and touch sensors are required for object detection. However, in places like factory with big open space, where high-precision and high-accuracy exteroceptive sensors may not be easily available or even feasible.

Hence, we propose to use only two types of sensors, namely RGB-D camera and bumper sensors, to learn about the environment and perform the task such as pick and place in a constrained environment with obstructions in scene. We propose to store the information of the objects on the scene using point cloud and handle object extraction using Point Cloud Library (PCL). Then we proceed to use `ROS` to communicate cylinder coordinates and bumper sensor data information for pick-and-place tasks.

II. PROBLEM SETUP

The problem we are facing here is to use the information from RGB-D camera and bumper sensors to perform the object detection and solve pick-and-place via grasping under constraints of potential obstructions that need to be cleared from the designated surface.

There are two parts of this problem. First, there needs to be a way to detect and locate the initial position and orientation of the target in the 3D scene. Second, the robot needs to pick up the target and place it onto the designated surface under the constraints, such as the designated surface being occupied by other objects. Hence, the robot needs to know the latest pose of the targeted object so that it knows where to grasps the target. The robot also needs to know whether the designated surface is free, and if not, where to move the obstructions to. We will be further discussing the existing work to handle the above-mentioned problems and their limitations.

III. RELATED WORK

As the problem can be subdivided into two parts, we have looked into the two sections: object detection and segmentation and detection of hindering objects in the environments.

A. Object detection and segmentation

There exists various ways of detecting objects in robotic research. Many work with pick-and-place and sorting as the final goal implements image processing, segmentation and classification to detect and categorise the object. Kumar et al. has used feature extraction, "Canny Method" based edge detection followed by classification using the Artificial Neural Network (ANN)[2]. Vijayalaxmi et al. has used grey-level image segmentation and pre-defined positions on the conveyor belt for object detection and identification[3]. However, these methods are mainly for object detection and classification in 2D, and not really provide depth information of the object. Moreover, the classification is not part of the task in the problem we want to study. We then looked into the work by Tsarouchi et al., who have combined 2D vision system with data from computer-aided design (CAD) files for generation of 3D coordinates[4]. However, this requires storage, mapping and maintenance of both the CAD data and vision data. Also, not all objects can be generated in CAD. Börcs et al. has discussed the effective object detection method using point cloud formed by using LiDAR, followed by a classification using Convolutional Neural Network (CNN) to effectively segment, extract and identify the objects[5].

B. Detection of hindering objects in the environment

The ability to detect and avoid obstacles is one of the most basic functions an autonomous robot should have. The use of LIDAR, sonar and cameras as sensors is common in the field of object detection. Romdhane et al have explored obstacle detection for collision avoidance through the use of monocular and stereo camera approaches [6]. However, monocular based methods are heavily dependant on a prior of an approximated location of the object for obstacle detection [6], which may not be always available.

In recent years, RBGD cameras have been widely explored as an alternative as they overcome the planar limitations of the previously mentioned sensors, providing real-time fulcrum based depth readings [7]. Peasley et al have suggested mounting an RBGD camera on the front of the mobile robot. The 3D point cloud is then transformed to a birds-eye view to create a map of obstacles in the current field of view of the objects[7]. This approach, however, is constrained to an indoor environment as a ground-plane constraint was applied. This is where it is assumed the robot will be traversing a flat ground plane, and anything above the identified ground plane can be a potential obstacle [7]. Since the robot is traversing the ground, any obstacles detected with a height above 0.5m were filtered out as the object would not be at risk of colliding with surfaces of that height. By obtaining a planar obstacle map, obstacle avoidance algorithms can then be employed.

Similarly, Singh et al described their obstacle detection process as first segmenting point clouds into ground and non-ground planes. Then through the use of feature extraction methods, obstacles in the cameras view can be identified [8]. These methods are useful and effective when the initial locations of the hindering objects are unknown or changing.

IV. METHODOLOGY

This paper proposed using visual and touch sensors only to perform the automatic pick and place grasping task in a constrained environment with given surfaces and initial location of the hindering object. The visual sensor used in the study is a RGB-D camera and the touch sensors used are bumpers. Both are for object detection purpose. We proposed to extract a three-dimensional point cloud of the cylinder using RGB-D camera. We decided to use the PCL mentioned earlier for point cloud processing and object segmentation and extraction. We applied various filters such as voxel-grid, which uses clustering to down sample the size of the point clouds, and pass through filters, which filter out points irrelevant to the objects we are interested, to reduce the number of points in the cloud. As the extraction of the location and height of the cylinder needs to go through every point in the cloud, decreasing the number of points lowers the computational complexity of cylinder extraction.

We then use model based extraction methods to extract and remove the plan surface to facilitate extraction of cylinder. A mathematical approach is taken to calculate the normal of the remaining points to find the points with similar normal as the cylinder to extract the cylinder. Following that, the extracted normal is used to extract the cylinder using model based extraction with mathematical parameters of an infinite cylinder. The height of the cylinder is determined using the coordinates of the highest and the lowest point in the point cloud.

After the cylinder is detected, we keep track of the latest cylinder pose. We also detect and track whether the surface that cylinder is supposed to be moved to is free from any other object by using bumper sensors. Each surface has a bumper sensor located at the centre that notifies whether the surface is occupied. If occupied, a pick-and-place is performed to move the object to some other unoccupied table before placing the cylinder. The robot only receives which surface to put the cylinder as input information from the users, and perform the rest of the tasks automatically. Hence, Our proposed algorithm needs to handle the detection of any object on the designated table, clearing of the object and pick-and-place of cylinder to transport it to the designated surface. The details of the experimental setup and the simulations are introduced in section V.

V. EXPERIMENTS

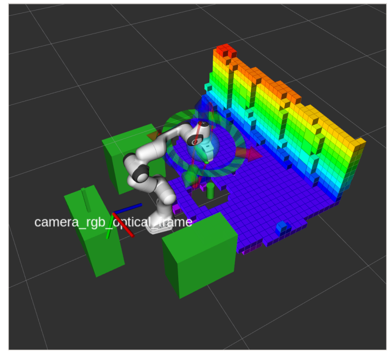

Fig. 1: Experimental setup

Due to the COVID-19 situation, we couldn't have the access to actual equipment and therefore proceed with simulations to experiment our algorithms for automatic pick-and-place grasping.

Here we are using PANDA Robotic Arm with 7 degrees of freedom. We set the environment to contain one surface of wall and plane and three cubes of dimension $0.2\times0.4\times0.4$ on the scene as shown in figure 1, namely "table1", "table2" and "table3".At the centre of each table, there is a bumper sensor to detect whether the table is occupied, namely "bumper1", "bumper2" and "bumper3". A RGB-D camera is installed behind the robot, facing the wall, as shown in the 1, for detecting objects on scene and forming point cloud. At the beginning, there is an object of dimension $0.02\times0.02\times0.2$ on table2, which results in the bumper2 informing the robot that table2 is occupied, whereas the the other two bumper sensors inform the robot of the unoccupied surface.

We will be having access to two information sources: the latest cylinder pose and the three bumper sensors' data. We used `PCL` and `ROS` in our simulation experiments. The cylinder pose is updated and published to the ROS topic `/cylinder_pose`, and the data from the bumper sensors is published to ROS topic `/bumper1`, `/bumper2` and `/bumper3`, with `/bumper2` publishing boolean messages of "true" and others publish "false" to indicate only table2 is occupied at the beginning.

Our experiments include testing the increase in extraction speed by applying various filters to the point cloud, the grasp of object, the process of pick-and-place to transfer the cylinder from the ground onto a free table, and the similar process of transferring the cylinder to an occupied table, which involves the removal of the object from the table. The results and discussions are detailed below.

A. *Increase in speed of cylinder extraction*

We have experimented various filters to speed up the cylinder extraction and test the time reduced. Below is the table to show the impact of the filtering on the time taken to extract the cylinder.

As shown in table I, the voxel grid filter alone is more effective than fast filter alone in extraction speed boost.

Action	Number of points	Time of cylinder extraction (microseconds)
RGB-D camera scan	307200	NA
Pass through filter to identify region of interest with range applied on z axis	83484	4278
Basic pass through filter + voxel grid filter to down sample the cloud	5687	490
Basic pass through filter + pass through filter (fast filter) identifying the region of cylinder	43122	2995
Basic pass through + voxel grid filter + fast filter	2769	384

TABLE I: Impact of applying different filters on speed of cylinder extraction

However, the combined effect of voxel grid and fast filter is the best.

B. Grasp of cylinder from ground to surface

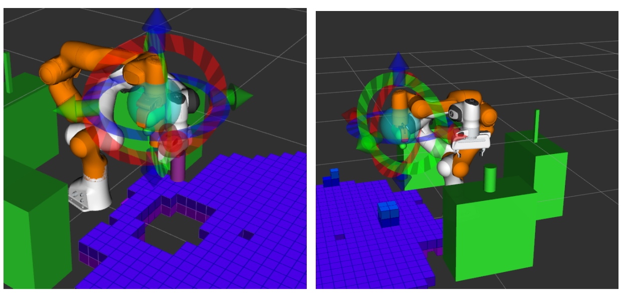

Fig. 2: Picking up the cylinder and placing onto table1

As shown in figure 2, upon receiving "1" as the input, the robot successfully automatically locates the position of the cylinder, picks up the cylinder and place it onto table1.

C. Automatic pick and place of cylinder in constrained environment

As shown in the figure 3, upon receiving instructions, the robot will automatically detect whether the designated table is occupied, and, if so, moves the occupant to another free table before placing the cylinder there. In this case, when the user input is "2", the robot detects that there is an object on the table 2, and moves the object from table 2 to table 3 upon detecting that table 3 is empty, before moving the cylinder from table 1 to table 2.

From the simulated experiments, we can see that the algorithm we developed using RGB-D camera and bumper sensors can successfully detect the presence of objects on the surfaces and automate the process of pick-and-place of the target under the constraints of having obstructions on the surfaces.

However, the experiments conducted in the simulation are largely simplified. The context of the constraints in the environment can be much more complicated. For example, most objects may not be placed right at the center on the surface in real life, and therefore may not be detected if there is only one bumper sensor at the center of the table. It could also be that there are many obstructions on the scene and the RGB-D camera cannot detect the target of interest, therefore cannot generate the 3D point cloud of the object. These limitations need further considerations when conducting experiments in real life.

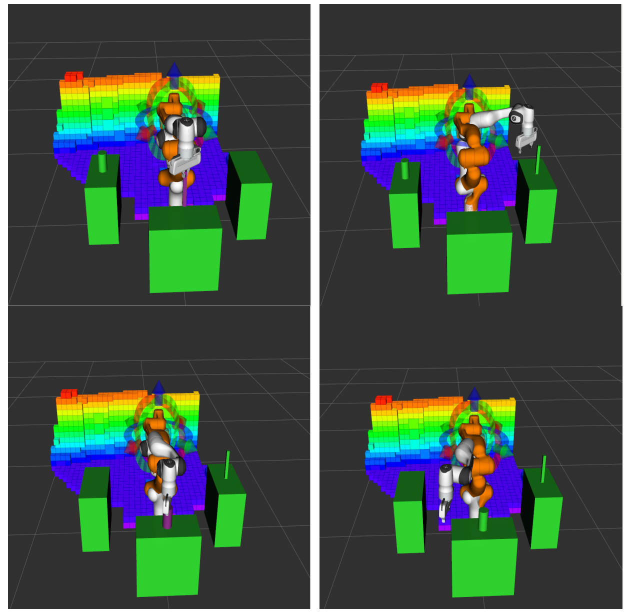

Fig. 3: Picking up the cylinder and placing onto table1

VI. CONCLUSION AND FUTURE WORK

A. Conclusion

In conclusion, we have developed the algorithms to successfully solve the automation of the pick-and-place problem of a target of interest in a simplified constrained environment. This work only requires the 3D point cloud generated from the RGB-D camera and the bumper sensors located on the surfaces. However, as the experimental setup is much simplified, possibly more sensors (e.g. one more RGB-D camera to capture the target of interest from different angles, or more bumper sensors to cover the entire surface) may be needed to successfully complete the autonomous pick and place in real life scenarios.

B. Future Work

As described, more work can be done to improve the algorithm to work better in a more complex real-life scenario, such as incorporating more than one RGB-D camera or bumper sensor data. Also, we can look into object identification by implementing classification algorithms such as CNN models for image recognition for further functions, such as picking up the required objects input by users and put them to the designated tables.

REFERENCES

[1] "The impact of robots on productivity, employment and jobs," 2017. [Online]. Available: https://ifr.org/img/office/IFR_The_Impact_of_Robots_on_Employment.pdf

[2] R. Kumar, Sanjesh Kumar, S. Lal, and Praneel Chand, "Object detection and recognition for a pick and place robot," 2014. [Online]. Available: http://rgdoi.net/10.13140/2.1.4379.2165

[3] G. S. P. L. P.B.VIJAYALAXMI, ROHAN PUTTA, "Object detection using image processing for an industrial robot," 2013. [Online]. Available: http://www.iraj.in/journal/journal_file/journal_pdf/3-24-139087678821-26.pdf

[4] P. Tsarouchi, S.-A. Matthaiakis, G. Michalos, S. Makris, and G. Chryssolouris, "A method for detection of randomly placed objects for robotic handling," *CIRP Journal of Manufacturing Science and Technology*, vol. 14, pp. 20–27, Aug. 2016. [Online]. Available: https://doi.org/10.1016/j.cirpj.2016.04.005

[5] A. Borcs, B. Nagy, and C. Benedek, "Instant object detection in lidar point clouds," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 7, pp. 992–996, July 2017. [Online]. Available: https://doi.org/10.1109/lgrs.2017.2674799

[6] N. B. Romdhane, M. Hammami, and H. Ben-Abdallah, "A generic obstacle detection method for collision avoidance," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, June 2011. [Online]. Available: https://doi.org/10.1109/ivs.2011.5940503

[7] B. Peasley and S. Birchfield, "Real-time obstacle detection and avoidance in the presence of specular surfaces using an active 3d sensor," in *2013 IEEE Workshop on Robot Vision (WORV)*. IEEE, Jan. 2013. [Online]. Available: https://doi.org/10.1109/worv.2013.6521938

[8] Y. Singh, , and L. Kaur, "Obstacle detection techniques in outdoor environment: Process, study and analysis," *International Journal of Image, Graphics and Signal Processing*, vol. 9, no. 5, pp. 35–53, May 2017. [Online]. Available: https://doi.org/10.5815/ijigsp.2017.05.05