Implement of Model-based Grasping

Team 16

Ximeng Liu

Ziyi Zhao

Junjie Hao

INTRODUCTION

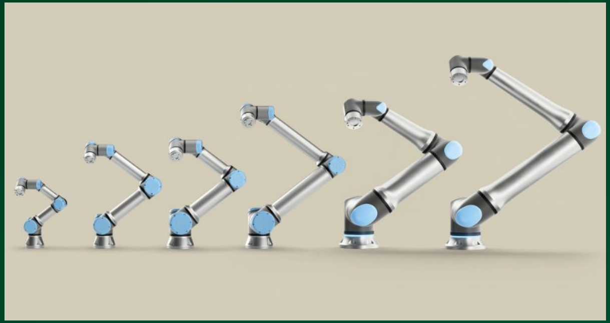
Objective of the Project:

- Implementing a model-based grasping system within a simulated environment

Real-World Applications:

- Robotic manipulation in industrial environments is mostly fixed
- Focuses on grasp and motion planning

Environment



Simulation Platform:

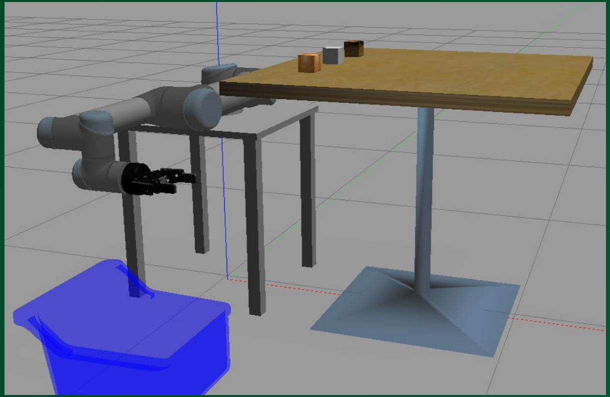
- Gazebo
- ROS (Robot Operating System)

Robot Used:

- UR5 Robotic Arm
- Robotiq 85 (Two-finger Gripper)

Object Models:

- Gazebo
- Virtual models in simulator



Implementation

Pipeline：

Use UR5 + Robotiq 85 Gripper

D435 for point cloud, PCL for grasp position calculation and MoveIt

MoveIt:

Handles motion execution

Camera:

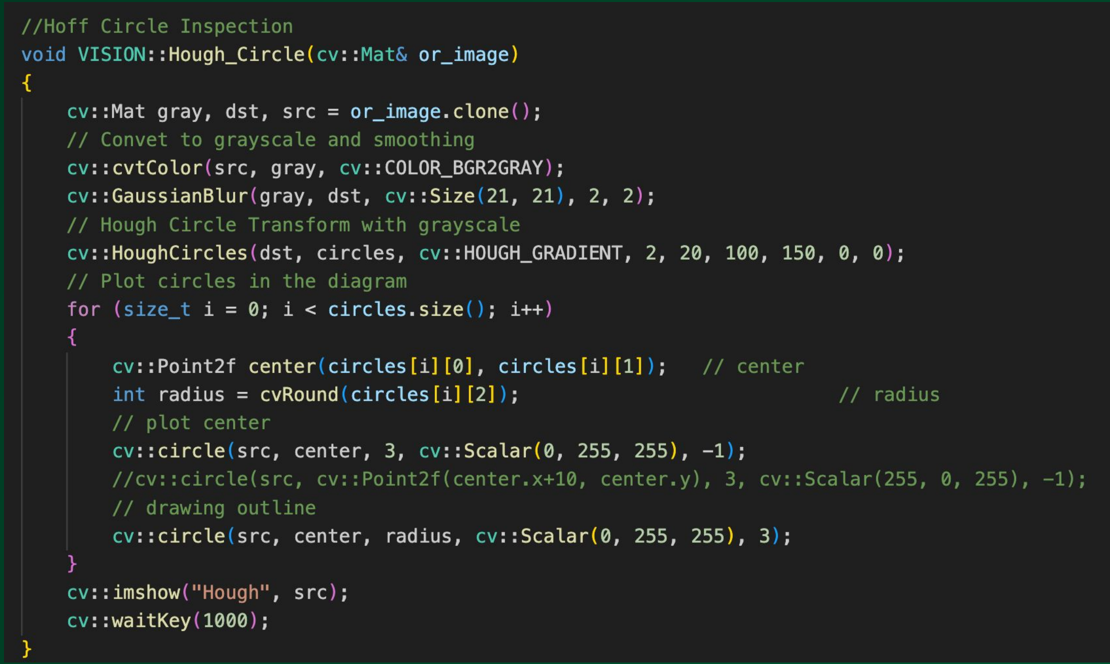
Intel RealSense D435 (rgb-d) point collection

Outputting both color images and depth data

OpenCV:

Open-source library for real-time computer vision

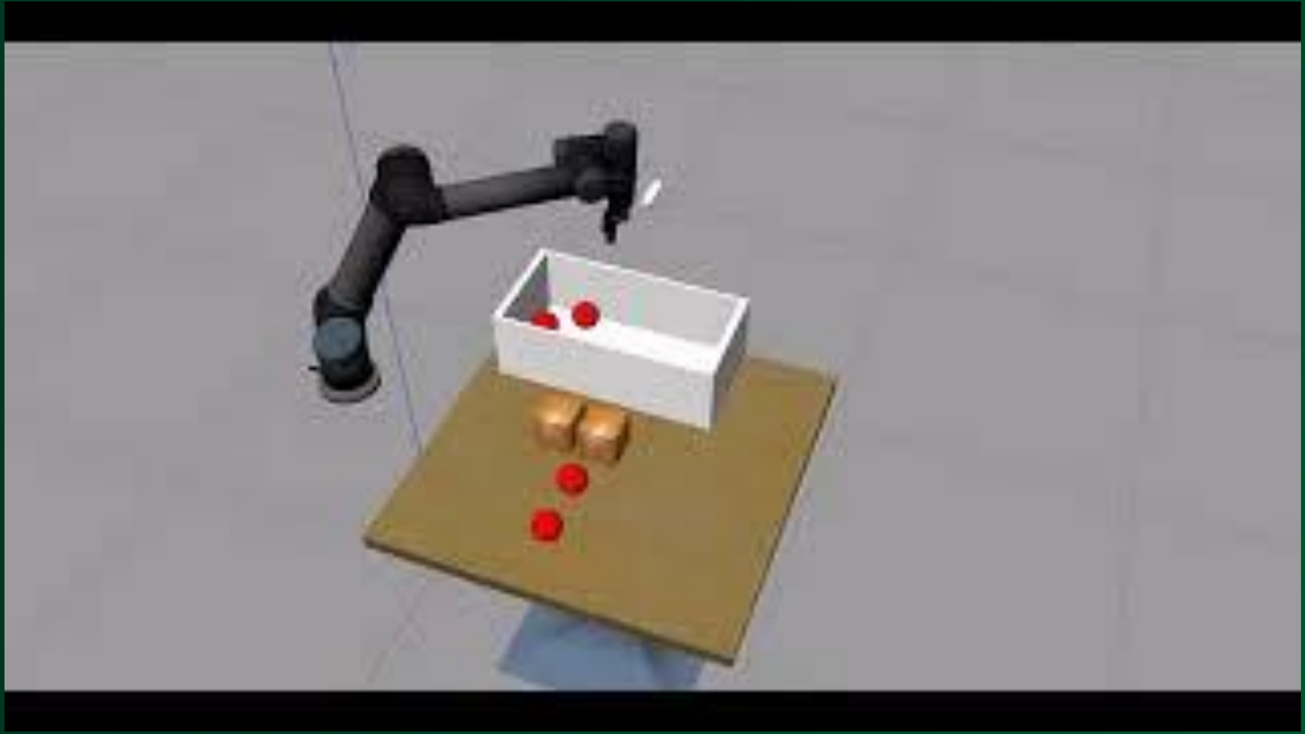
Calculating grasping position



```
//Hoff Circle Inspection
void VISION::Hough_Circle(cv::Mat& or_image)
{
    cv::Mat gray, dst, src = or_image.clone();
    // Convet to grayscale and smoothing
    cv::cvtColor(src, gray, cv::COLOR_BGR2GRAY);
    cv::GaussianBlur(gray, dst, cv::Size(21, 21), 2, 2);
    // Hough Circle Transform with grayscale
    cv::HoughCircles(dst, circles, cv::HOUGH_GRADIENT, 2, 20, 100, 150, 0, 0);
    // Plot circles in the diagram
    for (size_t i = 0; i < circles.size(); i++)
    {
        cv::Point2f center(circles[i][0], circles[i][1]);   // center
        int radius = cvRound(circles[i][2]);                            // radius
        // plot center
        cv::circle(src, center, 3, cv::Scalar(0, 255, 255), -1);
        //cv::circle(src, cv::Point2f(center.x+10, center.y), 3, cv::Scalar(255, 0, 255), -1);
        // drawing outline
        cv::circle(src, center, radius, cv::Scalar(0, 255, 255), 3);
    }
    cv::imshow("Hough", src);
    cv::waitKey(1000);
}
```

Demo

https://www.youtube.com/watch?v=wnhJhORoVHA



Future Work

Future Work

- More testing on grasping to collect more data
- Enhancements on grasp planning algorithms
- Start working on the evaluation metrics
- Documenting the project into the final report

Reference

1] “Moveit.” [Online]. Available: https://graspit-simulator.github.io/
[2] “Gazebo.” [Online]. Available: https://gazebosim.org/home
[3] “Moveit.” [Online]. Available: https://moveit.ai/
[4] “Ycb.” [Online]. Available: https://www.ycbbenchmarks.com/
[5] J. Kuffner and S. LaValle, “Rrt-connect: An efficient approach to singlequery path planning,” in Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), vol. 2, 2000, pp. 995–1001 vol.2.
[6] M. Jordan and A. Perez, “Optimal bidirectional rapidly-exploring random trees,” 2013.

Thank you