# Robotic Manipulation - Final Project Report

Hande Huang

***Abstract*—This project explores advancements in robotic manipulation, focusing on model-free antipodal grasping techniques. In previous work (assignment 6), the task was limited to grasping two cuboid objects. In this extension, we address the challenge of grasping multiple objects with varying shapes, including cuboids, spheres, and cylinders, while maintaining optimal performance with a gripper. The primary objective is to enhance the robotic system's ability to identify and grasp multiple objects with different geometries and orientations in an unstructured environment. We implement a methodology based on point cloud data, using clustering algorithms like DBSCAN to identify object centers and segregate objects for independent grasping. The project assumes that the objects lie flat on the ground and that their sizes are within the gripper's reach. The core of the work revolves around developing a function to calculate the optimal gripper pose for each object, ensuring stability and efficiency during manipulation. The results demonstrate the feasibility of this approach for diverse object types and multi-object scenarios, contributing to more versatile robotic grasping capabilities in real-world applications. This work advances the field by extending model-free grasping from single to multiple, more complex objects, and lays the groundwork for future exploration into more adaptive robotic systems.**

## I. Introduction

Robotic manipulation, especially grasping, remains a fundamental challenge in the field of robotics, particularly when dealing with a variety of object shapes and configurations. In industrial and service robotics, the ability to manipulate objects efficiently and reliably is crucial for applications such as assembly, sorting, and even healthcare. However, current grasping systems are often limited to handling only a small subset of objects, typically with simple shapes and well-defined geometries. As the diversity of objects encountered in real-world environments increases, robots must be capable of grasping multiple objects of varying shapes and sizes with the same precision and reliability. This motivates the need for more adaptive grasping algorithms that can generalize across different object types and handle dynamic environments.

**Problem Statement:** In this project, we aim to extend existing model-free antipodal grasping techniques to handle multiple objects with different shapes. Specifically, we focus on grasping cuboids, spheres, and cylinders using a robotic gripper. The problem setup involves the following assumptions: (1) objects are limited to rotationally symmetric shapes, (2) objects rest flat on the ground with the longest side in contact with the surface, (3) the gripper is capable of grasping objects of varying sizes, and (4) point cloud data is available for the scene. The goal of this work is to develop an approach that can compute optimal gripper poses for grasping each object in such multi-object configurations. The challenge lies in handling the increased complexity of grasping multiple objects with different shapes while ensuring stability and efficiency. Our approach builds on previous work in model-free grasping, extending it to a more general and practical setup involving multiple objects of different shapes.

## II. Literature Review

For this literature review, I have selected three papers that are relevant to the model-free direction of this project.

1) First, a striking and somewhat intimidating paper: *Data Scaling Laws in Imitation Learning for Robotic Manipulation* [1], accepted at ICLR 2025. This work investigates how scaling up training data impacts the performance of imitation learning in robotic manipulation. Through extensive experiments, the authors demonstrate that increasing the amount of data—across environments, object types, and demonstrations—significantly improves policy generalization. Their results show that a policy trained on large-scale data can perform zero-shot manipulation tasks across similar categories without further fine-tuning. This finding parallels trends in NLP and vision, suggesting that data scaling is a key to building versatile and generalizable robotic systems.
   Conclusion: This is groundbreaking, but also daunting. I'm not sure I'm ready for that level of scale—yet.
2) The second paper, *3D-VLA: A 3D Vision-Language-Action Generative World Model* [2], offers a more grounded and accessible approach. It uses ChatGPT to interpret high-level task instructions, and then a diffusion model to iteratively transform an "initial state" image into a "goal state" image. The resulting visual plan is then translated into robot commands using inverse kinematics. This pipeline effectively links language, vision, and action, enabling the robot to respond to dynamic tasks more flexibly. The use of both generative vision and language models provides a robust, adaptive system that can reason about goals and modify its plan on the fly based on real-world changes.
3) The third paper slightly diverges from the model-free theme, but is too interesting to omit: *VLMPC: Vision-Language Model Predictive Control for Robotic Manipulation* [3], presented at RSS 2024. It proposes a hybrid method that augments traditional Model Predictive Control (MPC) with the perception capabilities of Vision-Language Models (VLMs). Standard MPC is strong in planning but weak in perception; VLMPC addresses this by using goal images or language instructions to guide a conditional action sampling module, which proposes candidate action sequences. These sequences are then evaluated and executed via MPC. The integration of perception and control in this way enables better generalization across tasks and environments, and

equips the system to adapt to unforeseen changes during manipulation. It's a compelling fusion of model-based planning with high-level perceptual reasoning.

## III. Methodology

A brief recap: in this project, we extend from the 6th homework - model-free antipodal grasping, by upgrading the objects involved from two to multiple, and from only cuboids to also including spheres and cylinders.

### A. Assumptions

We assume the following:

1) the objects are limited to cuboids, spheres, and cylinders, which are rotationally symmetric
2) the objects lie "flat" on the ground, meaning they have the longest side touching the ground
3) the size of the objects are indeed graspable by the gripper
4) the number of objects is limited to the extent that a reasonably good point cloud can be produced

### B. Models

- DBSCAN - $cluster\_dbscan()$

### C. Approach

We mainly work on the function that returns optimal gripper poses for grasping objects. The steps of the approach are as follows:

1) An outsider function, find_object_centers(), is run first to find the centers of separate objects in the point cloud using DBSCAN clustering and return both the centroids and the segmented point clouds. Loop through each centroid/segmented point cloud:
2) $Startloop$

   For each object, find the point within a range around z = centroid_z closest to the centroid. In other words, find the points in the object point cloud that lies on the horizontal plane (z value / height of the plane is fixed) where its centroid lies on, as seen in Figure 1.

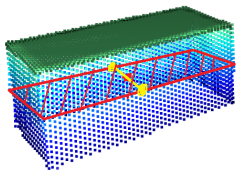

Fig. 1. Centroid (red dot), centroid horizontal plane (red plane), and grasp points (yellow dots). In actual implementation, it won't be just this one red plane but a range of horizontal planes around it.

   a) note that we select a range of z values, so it's technically not a plane, but rather a few horizontal planes that wrap around the perfect horizontal plane of the centroid. This is to account for imperfections in point cloud structure / alignment.

3) Then from the object points that lies on the selected plane/region, find the point that has the shortest distance to the centroid. This is the point where the gripper finger should land on. Mirror this point with respect to the centroid to get the other gripper finger point on the other side of the object.
4) From the normals of the two points above, we can figure out the angle parameter of the gripper state to return.
   a) We need to return: [x_centroid, y_centroid, z_centroid, theta_from_normals]
5) Finally, add a "elevation" constant value to the z value of the gripper state. It can be positive (elevation) or negative (descension), which negative/descension actually performs better, as point clouds doesn't capture the points near the bottom of the object (see Figure 2), thus producing a "centroid" that is higher than the actual object centroid. To fix this, we use this constant to reduce the z value / height.

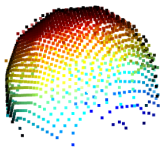

Fig. 2. Point cloud of a ball, which shows that the bottom is not captured.

6) We append: [x_centroid, y_centroid, z_centroid + elevation, theta_from_normals] to the list of gripper states for each object
   a) Technically we can just use this one instead of calculating gripper pose for other objects as well, but I'm doing it in case I need them later.

   $Endloop$

7) From the list of gripper states (for each object), we select the first one as the function return.

## IV. Experiments

As mentioned earlier, I have attempted picking up ball and cylinder objects that I created. The grasping didn't work at first, but after I tuned the parameter for fine-tuning grasp height in my code to a bit lower than before, things started to work pretty well.

As seen in Figures 3 and 4, the arm can successfully grab the added objects (cylinder and ball) and later place them aside without slipping out. I might upload a video to demonstrate this.

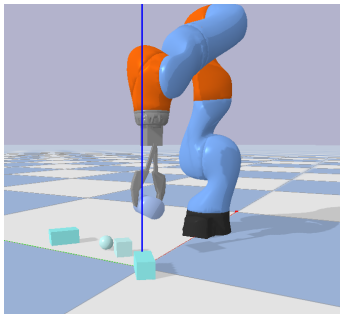

Fig. 3. Arm picking up a cylinder

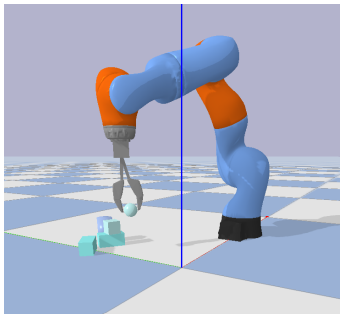

Fig. 4. Arm picking up a ball

## V. Discussion

### A. What worked well

The approach used in this project, particularly for grasping spheres and cylinders, showed positive results. The task of identifying object centers and calculating optimal gripper poses worked effectively within the constraints of the assumptions made. Specifically, the algorithm was able to successfully identify and grasp rotationally symmetric objects in flat configurations, which proved the feasibility of extending the previous model-free antipodal grasping technique to handle these new object types. This task was achieved with a reasonable level of accuracy and consistency, providing a strong foundation for more complex scenarios.

### B. What I liked about the approach

One of the key strengths of the approach was its simplicity and generality. By leveraging clustering algorithms like DBSCAN for point cloud segmentation, the method was flexible enough to handle different object shapes (cuboids, spheres, cylinders) without the need for precise object models or detailed prior knowledge of the environment. This made the approach robust to varying object configurations and easily scalable for multiple object grasping scenarios. I also really like the picking grasp plane by the horizontal plane of the centroid method, as it's pretty intuitive and it works.

### C. Limitations and Potemtial Improvements

1) **Assumption of Rotation Symmetry and Non-Tall Objects**
One major limitation was the assumption that all objects were rotationally symmetric and not too tall. While this assumption worked for simple cases, it failed when encountering objects with irregular shapes or tall, non-symmetric geometries. To overcome this limitation, future work could incorporate more sophisticated algorithms that consider the object's orientation and adjust the gripper pose accordingly. A quaternion-based system could be used to evaluate the orientation of non-symmetric objects and ensure that the gripper always lands at a valid graspable pose, regardless of the object's rotation.
2) **Limitations Due to Camera Placement**
The current setup also faced challenges related to camera placement. Objects were assumed to be within a fixed area, and the limited number of camera views sometimes resulted in incomplete or noisy point cloud data, making object detection and segmentation less accurate. A potential solution could be the use of multiple cameras placed at different angles and heights to capture a more comprehensive view of the scene. This would help in generating a more complete and accurate point cloud, improving object detection and segmentation performance.
3) **Segmentation of Stacked Objects**
The segmentation of stacked objects was another area where the approach struggled. The current segmentation algorithm often treated stacked objects as a single entity, as seen in Figure 5 leading to incorrect grasping behavior. To address this, a more advanced segmentation algorithm could be developed that takes into account the relative position and interaction of stacked objects. Alternatively, a more hands-on approach could be implemented, where the robot physically interacts with the stacked objects—pushing or picking them gently—to determine if they behave as a single object. If the stack remains intact, the system could treat it as one object for grasping purposes.

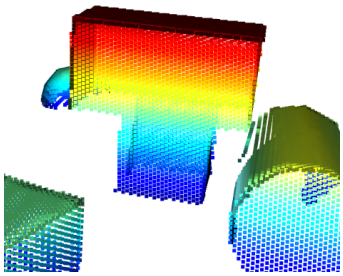

Fig. 5. Two stacked objects being treated as one object for the point cloud

## References

[1] F. Lin, Y. Hu, P. Sheng, C. Wen, J. You, and Y. Gao, "*Data Scaling Laws in Imitation Learning for Robotic Manipulation,*" in Proceedings of the International Conference on Learning Representations (ICLR), 2025.
[2] H. Zhen et al., "*3D-VLA: A 3D Vision-Language-Action Generative World Model,*" in Proceedings of Machine Learning Research (PMLR), 2024
[3] W. Zhao, J. Chen, Z. Meng, D. Mao, R. Song, and W. Zhang, "*VLMPC: Vision-Language Model Predictive Control for Robotic Manipulation,*" in Robotics: Science and Systems (RSS), 2024