NeuroGrip: Autonomous Reshelving using Vision Language Models

Yu-Hsin (Thomas) Chan[1], Boxiang (William) Fu[2], Joshua Pen[3], and Jet Situ[4]

***Abstract*— NeuroGrip is a robotic reshelving system that integrates Vision-Language Models (VLMs) to enable adaptive pick-and-place operations. Traditional warehouse robotics rely on fiducial markers and fixed structures, limiting flexibility. NeuroGrip utilizes a Franka Panda manipulator and a RealSense depth camera to interpret visual and language-based inputs for object handling. This paper reviews related work in industrial pick-and-place automation, including chain-of-thought (CoT) reasoning and vision-language-action (VLA) models, and provides a detailed overview of NeuroGrip's hardware setup. Our MVP implementation can be found here.**

I. INTRODUCTION

NeuroGrip is designed to improve robotic reshelving by reducing reliance on structured environments and predefined object locations. Traditional pick-and-place systems depend on fiducial markers, prepositioned shelving, and rigid localization techniques, limiting adaptability. By integrating Vision-Language Models (VLMs), NeuroGrip processes visual and language-based inputs to handle objects more flexibly. This paper examines related work in pick-and-place automation and vision-language integration, focusing on NeuroGrip's hardware setup, which includes a Franka Panda manipulator, a RealSense depth camera, a structured prop setup consisting of a pickup zone and shelf, and a prompt-to-location system for translating user commands into actionable placements.

II. RELATED WORK

A. Industrial Pick and Place

A primary motivation for pick and place behavior in our VLM context is the use within industrial warehouse robotics, referred to as material handling. Current solutions have a strong focus on prior organization, specialized labeling, which can then be fed into a simplified ArTag or AprilTag model that provides localization for mobile robots. In this variation, a Universal Robotics 5 cobot, a robust 6DOF mounted robot has rapidly become an industry and scientific standard, and in many cases, have been attached to a Ridgeback robot, enabling it to move autonomously in pick and place environments.

However, the existing pick and place space is strongly reliant on prepositioning components for the robots to work, including guidelines, markers, signal extenders, shelving design, walkway and railway design, etc. While effective, it results in a highly static and maladaptive configuration, losing all versatility and cross-applicability.

[1]yuhsinch@andrew.cmu.edu
[2]boxiangf@andrew.cmu.edu
[3]jpen@andrew.cmu.edu
[4]jets@andrew.cmu.edu

B. Chain-Of-Thought (CoT) Vision Language Models

A similar project to ours was published late last year by Wang and Zhang et. al [5], in which they proposed the SeeDo framework (Figure 1), whereby they provide their own methodology for the translation of VLM output into robot-parsable language.

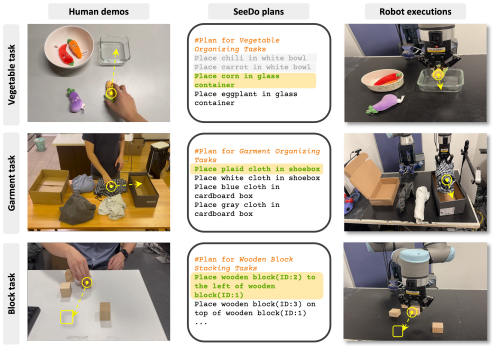

Fig. 1. SeeDo framework, as implemented by Wang and Zhang

Their variation is moderately more complex, utilizing chain-of-thought reasoning to decide what and where to perform object segmentation on a wide class of objects, thus performing generalizable tasks. From there, the CoT pipeline outputs predetermined positions, from which they can execute it with the robot arm, similar to our method of prepositioning the shelving unit and our existing calibration methods.

C. Vision-Language-Action (VLA) Models

More recent advances also propose the use of vision-language-action models. Such models forgo the translation into robot-parsable language entirely, and instead utilize an end-to-end learning model that takes in user prompts and directly outputs manipulator commands. Similar projects include the OpenVLA framework (Figure 2), TinyVLA framework, Google RT framework, and the OmniManip framework [6, 3, 1, 2].

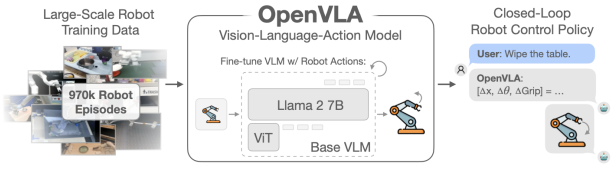

Fig. 2. OpenVLA framework

This type of model greatly generalizes the task space and can be used for a variety of end-to-end manipulation

tasks. However, the compute requirements are far higher compared to traditional methods, often requiring more than 16 gigabytes of RAM and recent NVIDIA GPUs and CUDA drivers.

III. METHODOLOGY

Our methodology involves using a prompt-to-text Large Language Model to convert the user input prompt to a pick-up item and drop-off location. Simultaneously, the RealSense on the end-effector takes a screenshot of the pickup zone. Based on the input prompt, the center pixel location of the item is calculated in the pickup zone. This is converted to an end-effector location through camera calibration. The end-effector picks up the item and moves to the drop-off location specified. Finally, the robot arm moves to its reset position ready to take another user input.

A. Hardware Setup

The entire hardware setup of our project is depicted in Figure 3. The setup consists of a Franka manipulator, shelf, pickup zone, RealSense stereo camera, and an off-location PC that runs the prompt-to-location software.

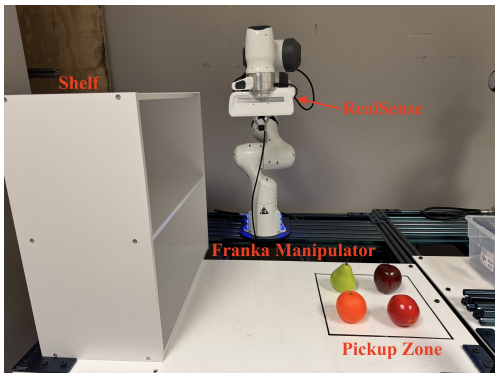

Fig. 3. Hardware setup of NeuroGrip

1) Franka Manipulator Setup: The manipulator used is a Franka Panda manipulator made available to us in the REL lab. In particular, we were allocated to the `iam-sneezy` manipulator. The robot arm has 7 degrees-of-freedom and can grab a payload of 3 kilograms. The hardware-software interface between the Franka manipulator and the control PC was already set up beforehand. The documentation for the interface setup can be found in Reference [7].

2) Shelf Setup: The shelf is placed in the left section of the world-frame table. We split the shelf into four quadrants (top-left, top-right, bottom-left, and bottom-right) for the end-effector to place items in. If the user inputs a prompt outside these four quadrants, the input will be invalidated and require the user to input a new prompt.

3) Pickup Zone Setup: The pickup zone is placed in the right section of the world-frame table. This is the location where the end-effector will attempt to pick up items and place them on the shelf. We have used dummy fruits for our setup, although the model is much more general and can recognize any object in the SAM2 database [4].

4) RealSense Setup: We installed a RealSense depth camera behind the Franka robot arm's end-effector and used it to capture images of the picking area. Figure 4 shows a clearer view of the RealSense mount location.

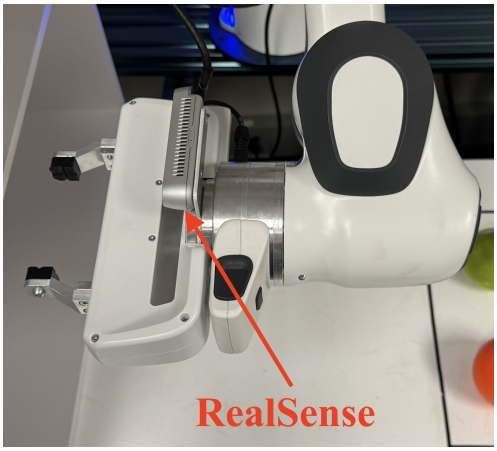

Fig. 4. RealSense setup of NeuroGrip

5) Prompt to Location Setup: Since we intend to run a visual language model in our project, we require a PC with a GPU. We discovered that the PC in the lab has an RTX3060Ti GPU installed, but the CUDA version is outdated for the model we want to run. While a software update could potentially resolve this issue, we were concerned that it might disrupt the entire environment. Consequently, we decided to deploy the model on Thomas' PC. Once a RealSense camera captures an image, we will send a request to Thomas' PC, which will process the image and provide the location of the target.

IV. REFERENCE

[1] Anthony Brohan et al. *RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control*. 2023. arXiv: 2307.15818 [cs.RO]. URL: https://arxiv.org/abs/2307.15818.

[2] Moo Jin Kim et al. *OpenVLA: An Open-Source Vision-Language-Action Model*. 2024. arXiv: 2406.09246 [cs.RO]. URL: https://arxiv.org/abs/2406.09246.

[3] Mingjie Pan et al. *OmniManip: Towards General Robotic Manipulation via Object-Centric Interaction Primitives as Spatial Constraints*. 2025. arXiv: 2501.03841 [cs.RO]. URL: https://arxiv.org/abs/2501.03841.

[4] Nikhila Ravi et al. "SAM 2: Segment Anything in Images and Videos". In: *arXiv preprint arXiv:2408.00714* (2024). URL: https://arxiv.org/abs/2408.00714.

[5] Beichen Wang et al. *VLM See, Robot Do: Human Demo Video to Robot Action Plan via Vision Language Model*. 2024. arXiv: 2410.08792 [cs.RO]. URL: https://arxiv.org/abs/2410.08792.

[6] Junjie Wen et al. *TinyVLA: Towards Fast, Data-Efficient Vision-Language-Action Models for Robotic Manipulation*. 2024. arXiv: 2409.12514 [cs.RO]. URL: https://arxiv.org/abs/2409.12514.

[7] Kevin Zhang et al. "A modular robotic arm control stack for research: Franka-interface and frankapy". In: *arXiv preprint arXiv:2011.02398* (2020).