Lab Project Proposal: Open-Vocabulary Mobile Manipulation

Supervisor: Rohit Menon

Figure 1: Overview of an Open Vocabulary Mobile Manipulation Process

Goal

Enable a mobile manipulator (**Toyota HSR**) to execute **open-vocabulary pick-and-place tasks** in home-like real-world environments. The system should understand high-level, **robot-agnostic natural-language instructions** (e.g., “Bring me the red cup”) and map them into executable action sequences. A central objective is to design the grounding layer in a form that is **transferable to any other robot with the same functional capabilities**, such as mobile navigation, RGB-D perception, and pick-and-place manipulation.

The pipeline will explicitly follow:

- **RGB-D SLAM** to build a consistent 3D map of the environment.
- **3D instance segmentation (Mask3D)** on the reconstructed scene.
- **Command grounding** into abstract task primitives.
- **Primitive execution** through robot-specific controllers.

Students will first build and validate the system in simulation and then deploy it on the real robot.

Motivation and Background

Many robotic systems still rely on rigid pipelines: they recognize only a fixed set of objects and execute pre-defined task scripts. This limits flexibility, scalability, and transfer to new settings.

In contrast, **open-vocabulary perception** combined with **abstract primitive-based task execution** allows a robot to interpret a broad range of user instructions while keeping the high-level task logic independent of a specific hardware platform. This makes the system more modular and more easily transferable to other service robots with comparable sensing and manipulation capabilities.

This lab project focuses on the complete chain from semantic 3D scene understanding to language-guided action execution in household environments.

Approach and Work Packages

The project is divided into work packages (WPs). Each student (2–3 students total) will take responsibility for 1-2 WP, with integration and testing performed jointly.

WP1 – Home Simulation in Isaac Sim: Create a realistic household environment in **Isaac Sim** and integrate a simulated **Toyota HSR**. The simulator should support RGB-D sensing, navigation, object interaction, and reproducible experiments for open-vocabulary mobile manipulation. This simulation environment will serve as the first validation platform before real-robot deployment.

WP2 – RGB-D SLAM and Mapping: Build a dense or semi-dense 3D map using RGB-D SLAM. Fuse onboard sensor observations into a globally consistent world model that can be used for downstream perception and planning.

WP3 – 3D Instance Segmentation: Apply **Mask3D** on the reconstructed 3D scene to obtain object-level instance segmentation. Combine this with open-vocabulary semantic labeling so that queried objects can be localized in the mapped environment.

WP4 – Robot-Agnostic Command Grounding: Define a set of reusable **abstract task primitives**, such as:

- navigate-to(location)
- scan-area
- pick(object)
- place(location)

Implement a command-grounding layer that converts natural-language instructions into a **sequence of abstract primitives**. This layer must remain **independent of Toyota HSR-specific APIs** and should instead assume only a generic capability interface. The goal is that the same grounding module can later be transferred to any other robot that provides the same features, e.g., navigation, object perception, grasping, and placing.

Example:

- "Bring me the red cup" → scan-area → navigate-to(cup) → pick(red cup) → navigate-to(user) → place(user zone)

WP5 – Primitive Execution and System Integration: Connect the abstract primitives to the Toyota HSR execution stack. Implement the robot-specific realization of navigation, scanning, picking, and placing. Integrate mapping, segmentation, grounding, and execution into a unified ROS-based system.

WP6 – Deployment and Evaluation: Validate the full pipeline first in Isaac Sim and then on the real Toyota HSR. Evaluate performance using metrics such as:

- task success rate
- execution time
- robustness to unseen objects
- transferability of the command-grounding module

Available Resources

- **Software:** ROS2, MoveIt2 or equivalent robot interfaces, Isaac Sim, Python and C++.
- **Models:** Mask3D, open-vocabulary detectors or vision-language models (e.g., OWL-ViT, GroundingDINO).
- **Hardware:** Toyota HSR with onboard RGB-D sensing and manipulation capabilities.

Requirements

- Programming: Python (mandatory), Linux, shell scripting, basic C++.
- Libraries: Open3D, OpenCV, ROS2, PyTorch.
- Interest in robotics, perception, embodied AI, and simulation.

Expected Outcome

- A modular pipeline: **SLAM → 3D instance segmentation → abstract command grounding → robot execution**.
- A household simulation environment in Isaac Sim containing a Toyota HSR for repeatable experiments.
- A command-grounding module that is **robot-agnostic** and transferable to other robots with comparable capabilities.
- Demonstration of open-vocabulary pick-and-place for unseen objects in simulation and on the real robot.
- Reproducible code and documented experiments for future lab projects.