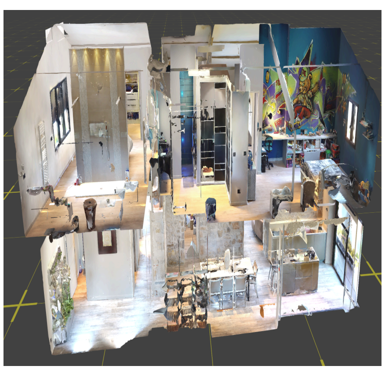
Modular Cross-Simulation Robotics Framework for Perception, Mapping, Planning, and Control

Lab: Scalable Spatial Intelligence Lab

Faculty advisor: Yulun Tian
Student mentor: Behrad Rabiei

Project description: Modern robotics systems rely on increasingly complex software stacks that integrate perception, mapping, planning, and control. However, most existing robotics frameworks are tightly coupled to a specific simulator (e.g., Habitat-Sim[1], Isaac Sim[2]), making it difficult to develop algorithms that generalize across simulation platforms. This project aims to design and implement a modular, simulation-environment-agnostic robotics framework that provides clean interfaces between sensing, world modeling, decision-making, and low-level control. The framework will allow plug-and-play replacement of perception modules, mapping sub-systems (e.g., occupancy, semantic BEV, scene graphs), planners (e.g., sampling-based, optimization-based), and control policies, with consistent APIs across simulation environments. This project is ideal for students interested in sim-to-real transfer, robotic software engineering, and modular system design for autonomous agents.



Key Milestones:

1. Install all required dependencies and create a clean, reproducible Docker image (by the end of Week 2).
2. Design a modular software architecture with clear APIs for perception, mapping, planning, and control. Implement a minimal working baseline that runs in both Habitat-Sim and Isaac Sim. An example baseline may include out-of-the-box semantic segmentation, occupancy mapping, an A* planner, and a proportional controller (Weeks 3–10).
3. Create adapter layers that allow modules to operate seamlessly across different simulators (e.g., sensor wrappers, physics wrappers). Implement a unified data representation that supports both simulated environments (Weeks 11–12).
4. For a motivated student who is ahead of schedule (choose one):

   a. Implement a research paper of your choice using the developed framework. For example [3,4] (remainder of the semester).
   b. Extend the framework for real world systems
5. Ensure the codebase is clean, well-documented, and reproducible. Provide usage tutorials, configuration examples, and a final project report (final weeks).

Application deadline: 12/15/2025
Application link:
https://docs.google.com/forms/d/e/1FAIpQLSdkrW3LawjwUMJL2WEcWK67VKcBoGK6BEgCVjgzvhODIwrYRA/viewform?usp=header

Desired qualifications:

- Relevant coursework for robotics (e.g., ROB 530, ROB 535, and ROB 550)
- Proficient programming in Python and C++
- Light course load during the corresponding semester
- Experience with at least one of the mentioned simulators
- Experience with Docker is a plus
- Experience with ROS 2 is a plus

References

[1] Puig, Xavier, et al. "Habitat 3.0: A co-habitat for humans, avatars and robots." *arXiv preprint arXiv:2310.13724* (2023).
[2] NVIDIA. Isaac Sim (Version 5.1.0) [Computer software]. https://github.com/isaac-sim/IsaacSim.
[3] Rabiei, Behrad, et al. "LTLCodeGen: Code Generation of Syntactically Correct Temporal Logic for Robot Task Planning." *arXiv preprint arXiv:2503.07902* (2025).
[4] Hu, Yue, et al. "Imaginative world modeling with scene graphs for embodied agent navigation." *arXiv preprint arXiv:2508.06990* (2025).