Navigating the World

with

ROS

Presented by

Sai Haneesh Allu

Outline

Introduction

Navigation concepts

- Map, Robot pose, and Path
- Taxonomy of Navigation
  - Localization
  - Path Planning
  - SLAM

ROS Navigation stack

- Planners
- Cost maps
- Mapping
- Localization

Recent work at IRVL

The Need for Navigation

Why Navigation is Crucial for Robots:

Autonomy: robots need to navigate their surroundings without human intervention.

Safety: Avoiding obstacles ensures that robots can operate without damaging themselves.

Efficiency: Optimized paths save time and energy.

Real-World Applications:

Warehouse Automation

Autonomous Vehicles

Service Robots

Challenges in Navigation:

Unknown Environments: Robots often have to explore and map new areas while keeping track of their position.

Dynamic Obstacles: People, animals, or moving objects require real-time path adjustments.

Localization: Robots need to constantly know their location relative to a map for precise movement.

The Sense-Plan-Act Paradigm

To accomplish a task, robot should understand the environment from its sensor measurements

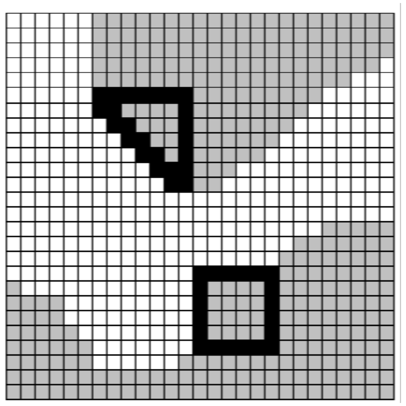
For autonomous navigation, robot should know

- What the environment looks like
  SENSE
- Where it is in the environment and determine how it should reach the target
  PLAN
- Follow the plan adjusting to the dynamics of the environment
  ACT

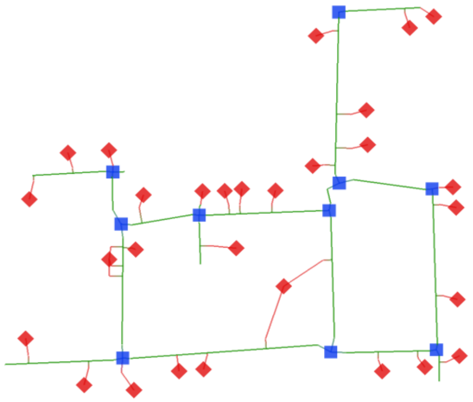
Courtesy: Giorgio Grisetti [2]

MAP

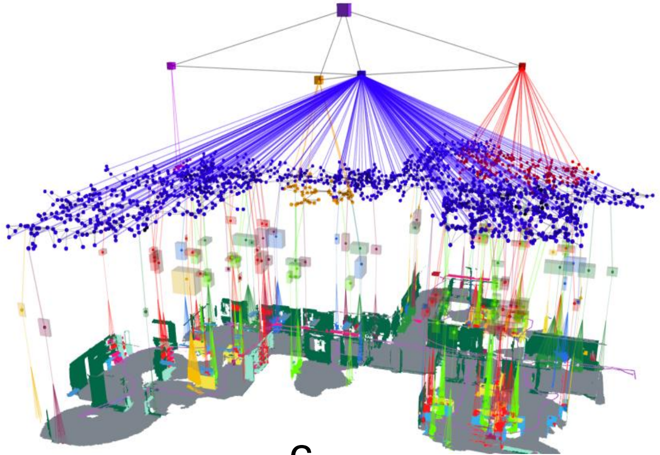
- Map is a representation of the environment
- It should contain enough information to accomplish the task
- Representations:
  a. Metric
  b. Topological
  c. Hybrid



a



b



c

Courtesy: Giorgio Grisetti [2]

Robot Pose and Path

Metric map defines the reference frame for the environment

Robot should know it's position relative to the reference frame

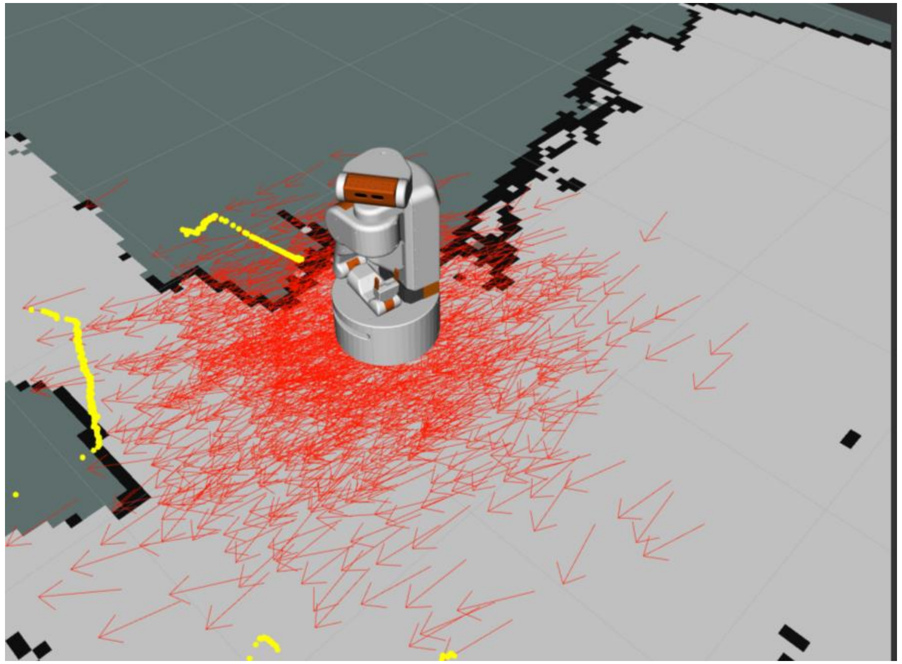
Robot pose Robot pose w.r.t reference frame

Path is defined as the sequence of waypoints from the start to the end location in the map

Courtesy: Giorgio Grisetti [2]

Localization

- Robot needs to **estimate its pose in the reference frame** using the sensor observations
- This task is accomplished by the **localization module**



Path Planning

- Determine (if it exists) a path to reach a given goal location, given a localized robot and a map of traversable regions

Courtesy: Giorgio Grisetti [2]

SLAM

Given a robot that has a perfect ego- estimate of the position, and a sequence of measurements, determine the map of the environment.

A perfect estimate of the robot pose is usually not available.

Instead, we solve a more complex problem: Simultaneous Localization and Mapping (SLAM)

SLAM

Process by which a robot creates a map of an unknown environment while simultaneously determining its position within that map.

It’s essential when a robot enters an unfamiliar space and needs to navigate without a pre-existing map.

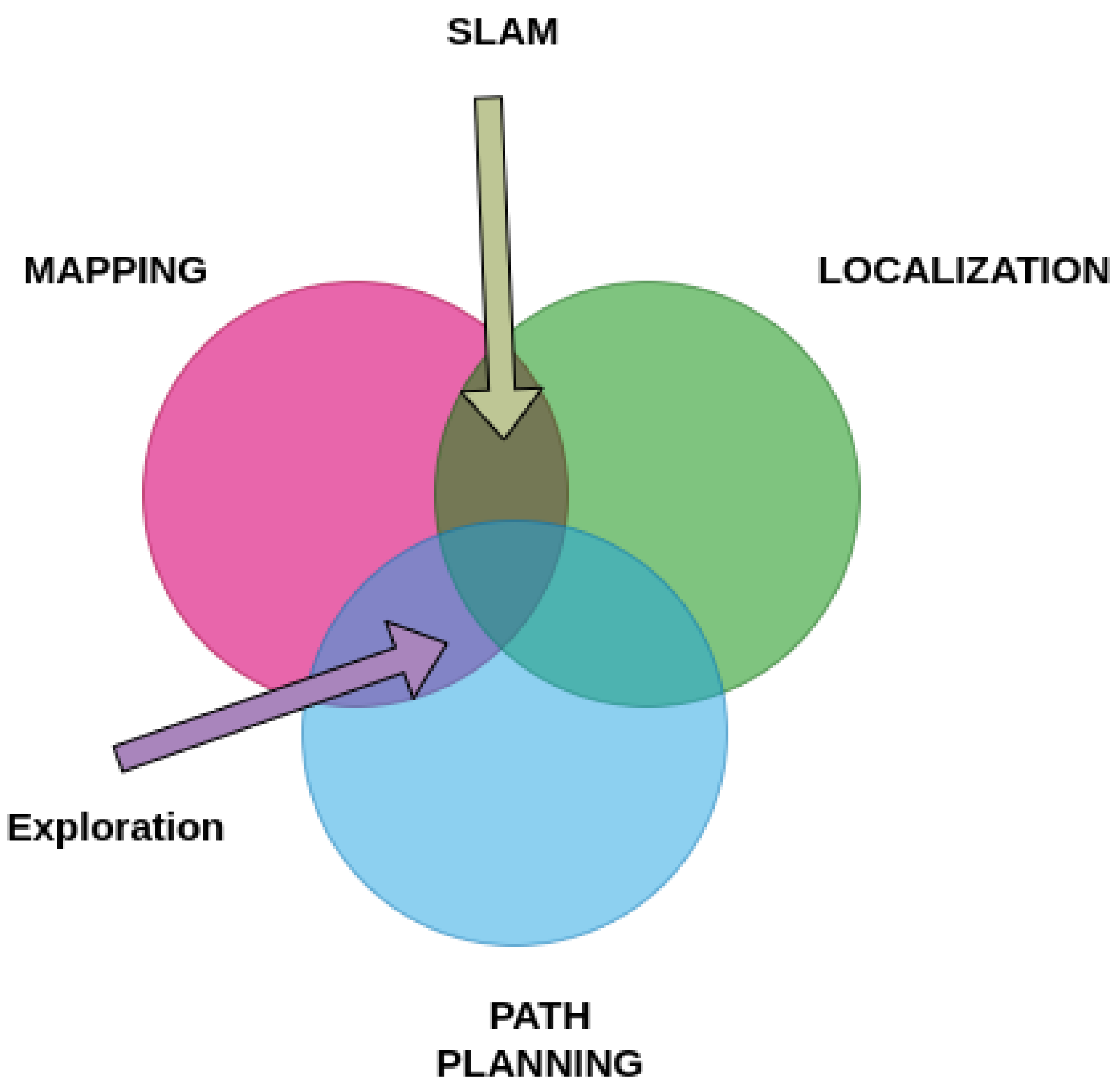
Estimate:

- The map of the environment
- The trajectory of the robot in the map

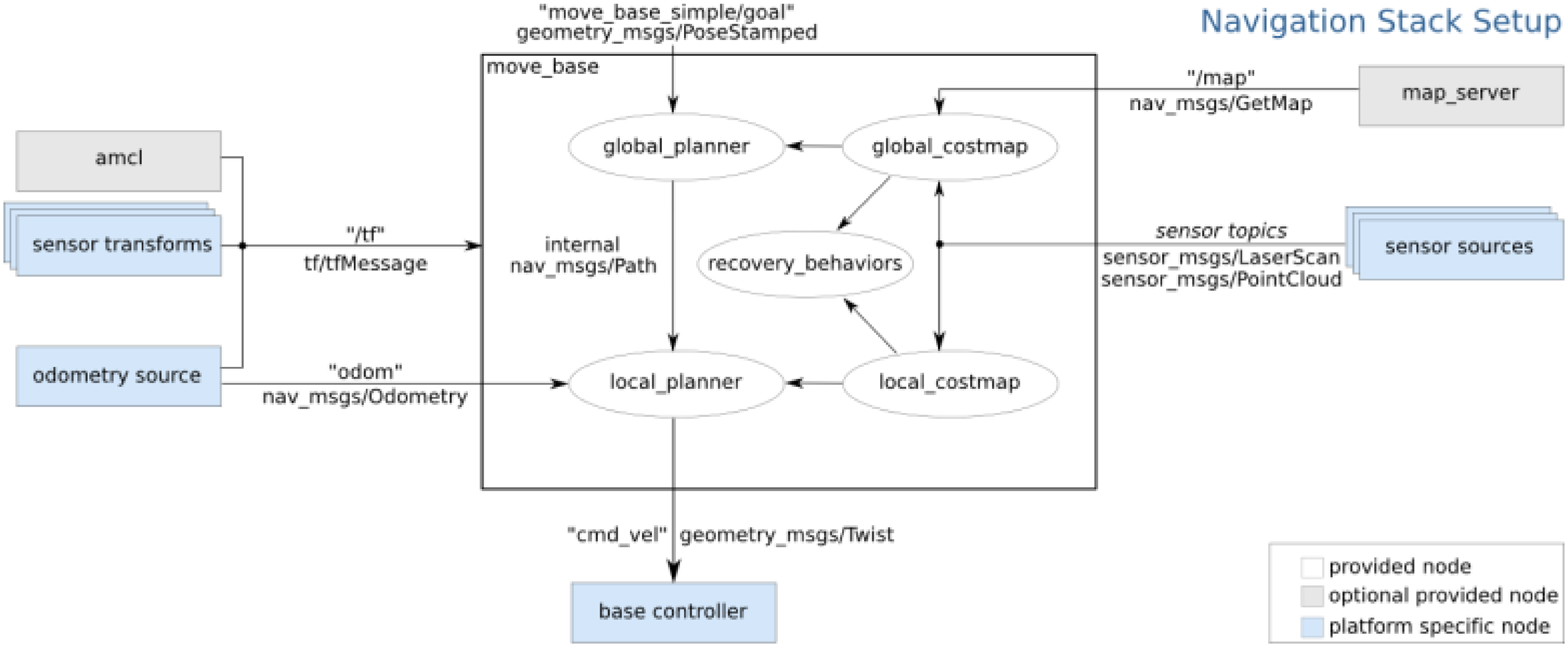
using sensor measurements

Courtesy: Giorgio Grisetti [2]

Summary of Autonomous Robot Navigation



ROS Navigation Stack



ROS Navigation Stack – Global Planner

To use move_base node, we need to have a global planner and a local planner.

Global planner:

A. Given a map, it plans a minimum cost path between start and end points

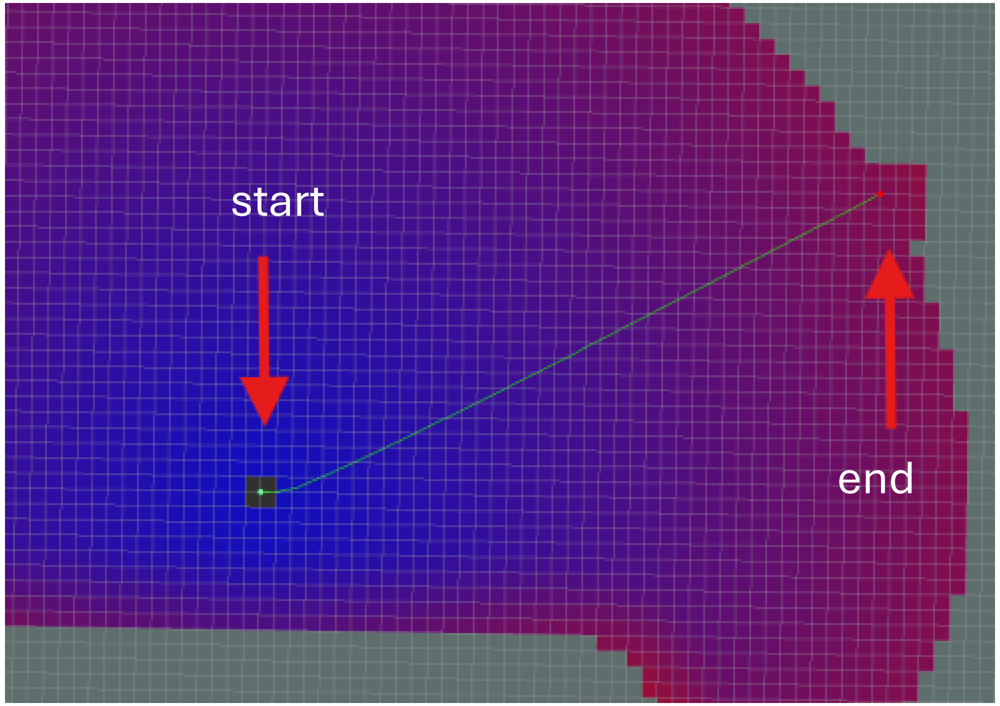
B. The planned path is free of obstacles

C. Runs at a lower frequency

`navfn, global_planner`

ROS Navigation Stack – Global Planner

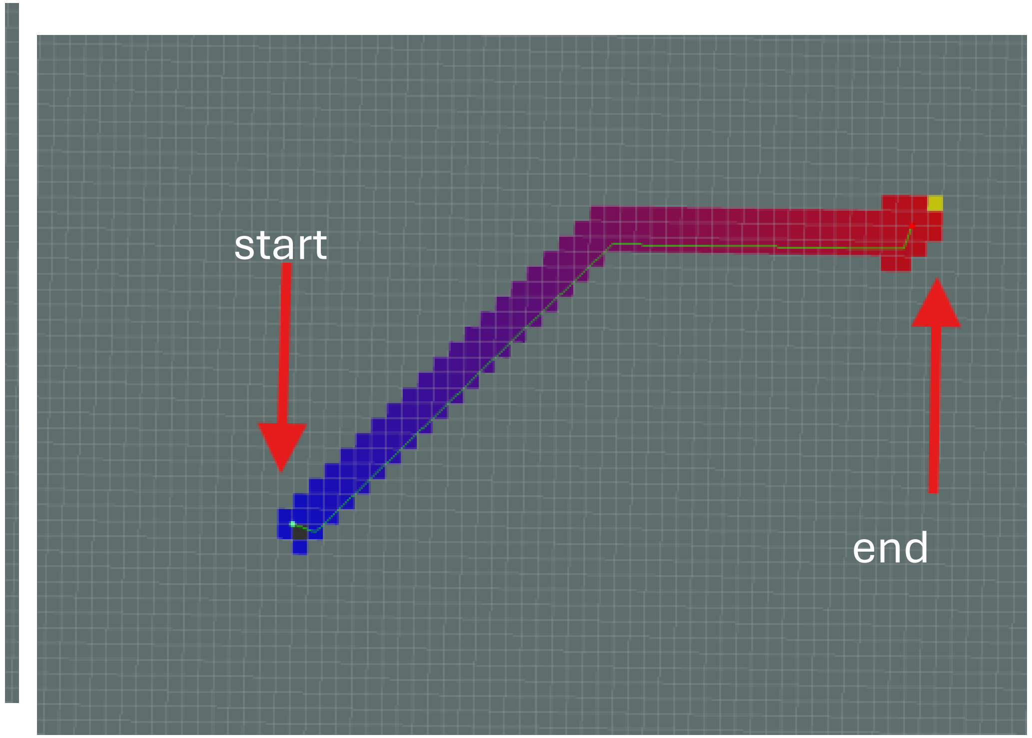
- Navfn
  - Based on Dijsktra algotihm



Courtesy: Kaiyu Zheng[1]

ROS Navigation Stack – Global Planner

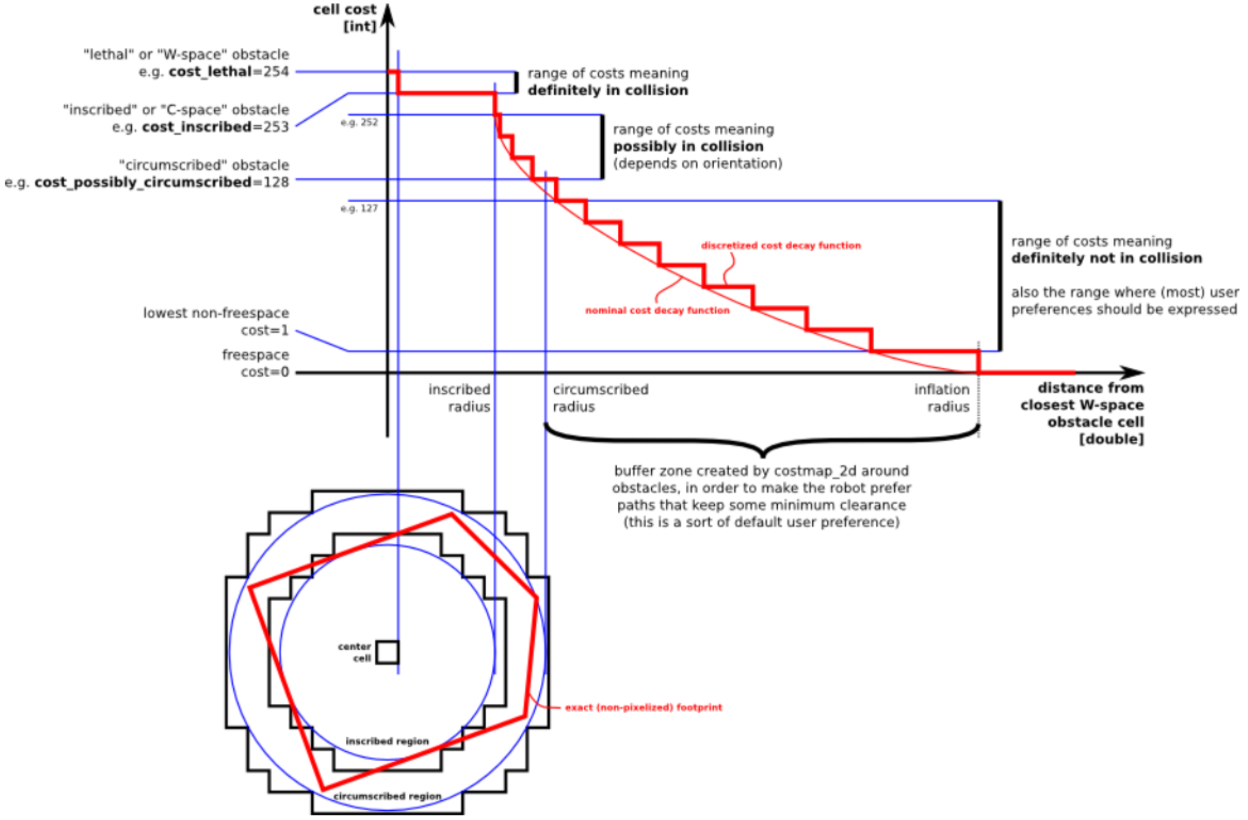
- Global_planner
  - Enhanced version of navfn
  - Added support for A*



Courtesy: Kaiyu Zheng[1]

ROS Navigation Stack – Cost Maps

- Keep the robot away from the obstacles
- Inflate the obstacles
- Inflate in an exponential manner



ROS Navigation Stack – Cost Maps

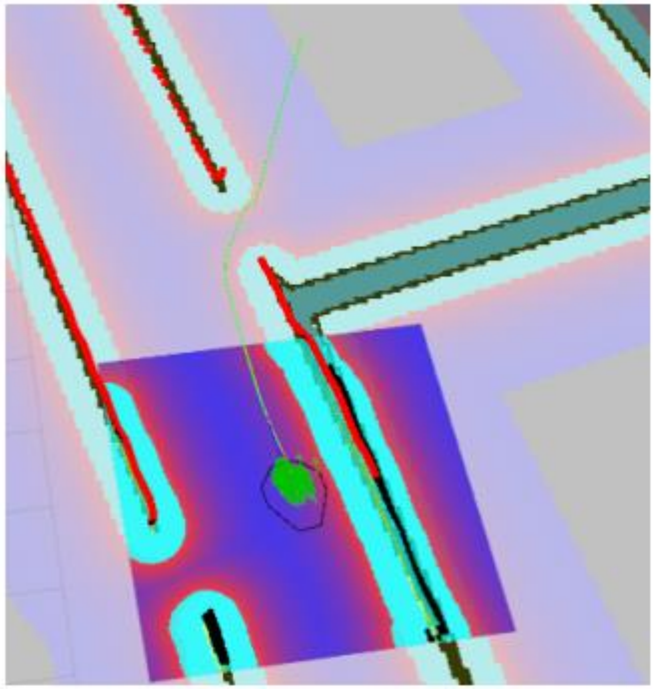


Figure 5: cost_factor = 0.01

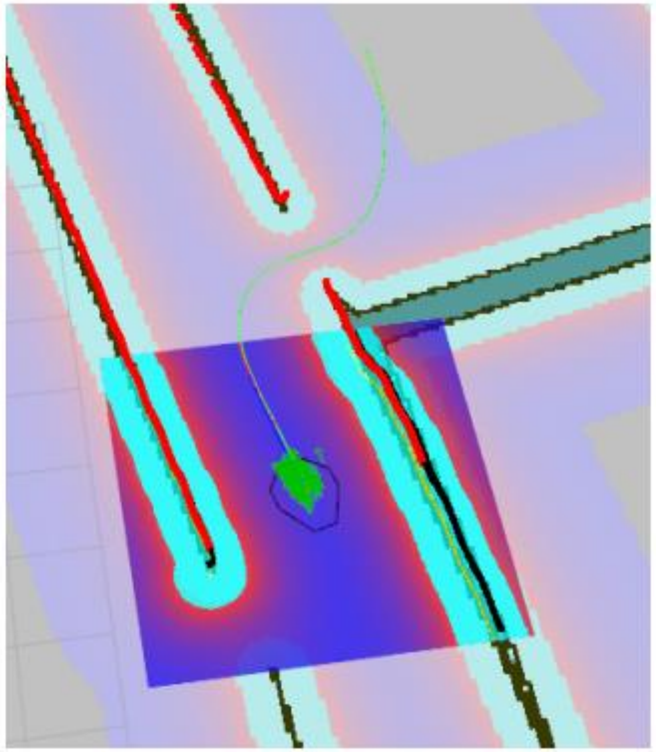


Figure 6: cost_factor = 0.55

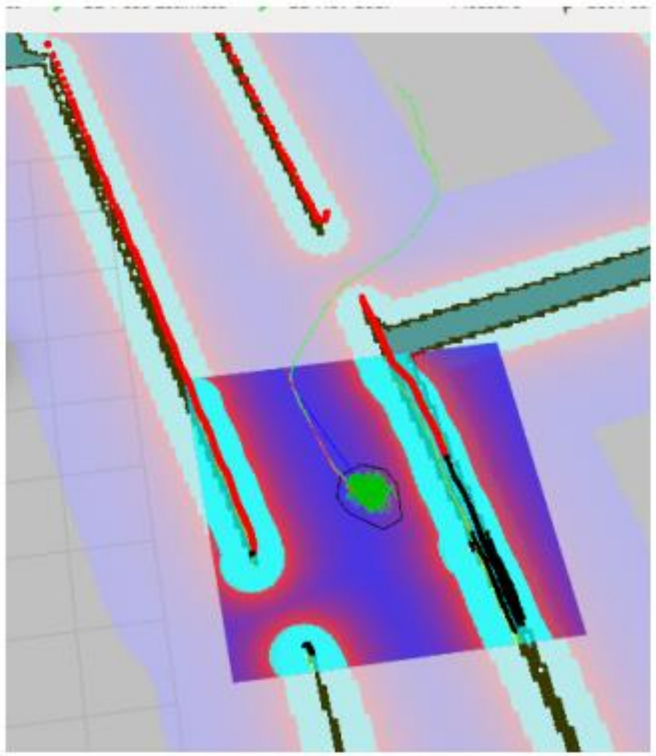


Figure 7: cost_factor = 3.55

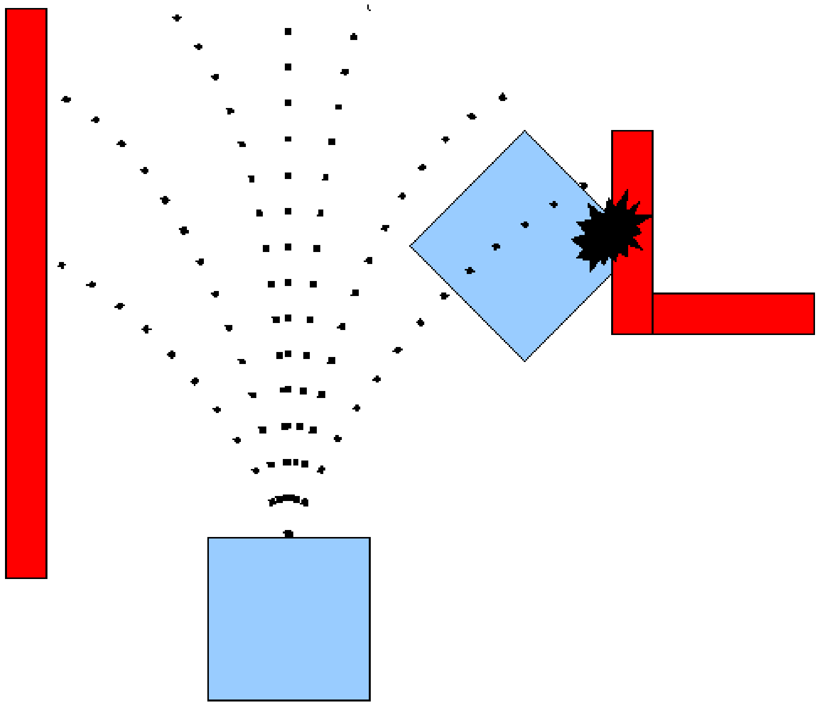
Courtesy: Kaiyu Zheng[1]

ROS Navigation Stack – Local Planner

- Expensive to update the global planner at a high frequency in a dynamic environment
- A local planner to avoid obstacles not present during the planning
- `dwa_planner`, `eband_local_planner`, `teb_local_planner`

ROS Navigation Stack – Local Planner

- DWA PLANNER: Dynamic window Approach
- Discretely sample robot control space $[v_x, v_y = 0, \omega\ ]$
- Perform forward simulation
- Evaluate each trajectory
- Pick high scoring legal trajectory



Courtesy: ROSCON 2021[3]

ROS Navigation Stack – Mapping

GMapping, slam_karto, hector_slam, slam_toolbox

Most popular in community – GMapping

Based on Rao-Blackwellized particle filter

- Initializes particles , each representing robot pose and predicted map
- As the robot moves, its pose is estimated along with the map
- Uses scan matching to match the laserscan with the map estimated by the particles

ROS Navigation Stack – Localization

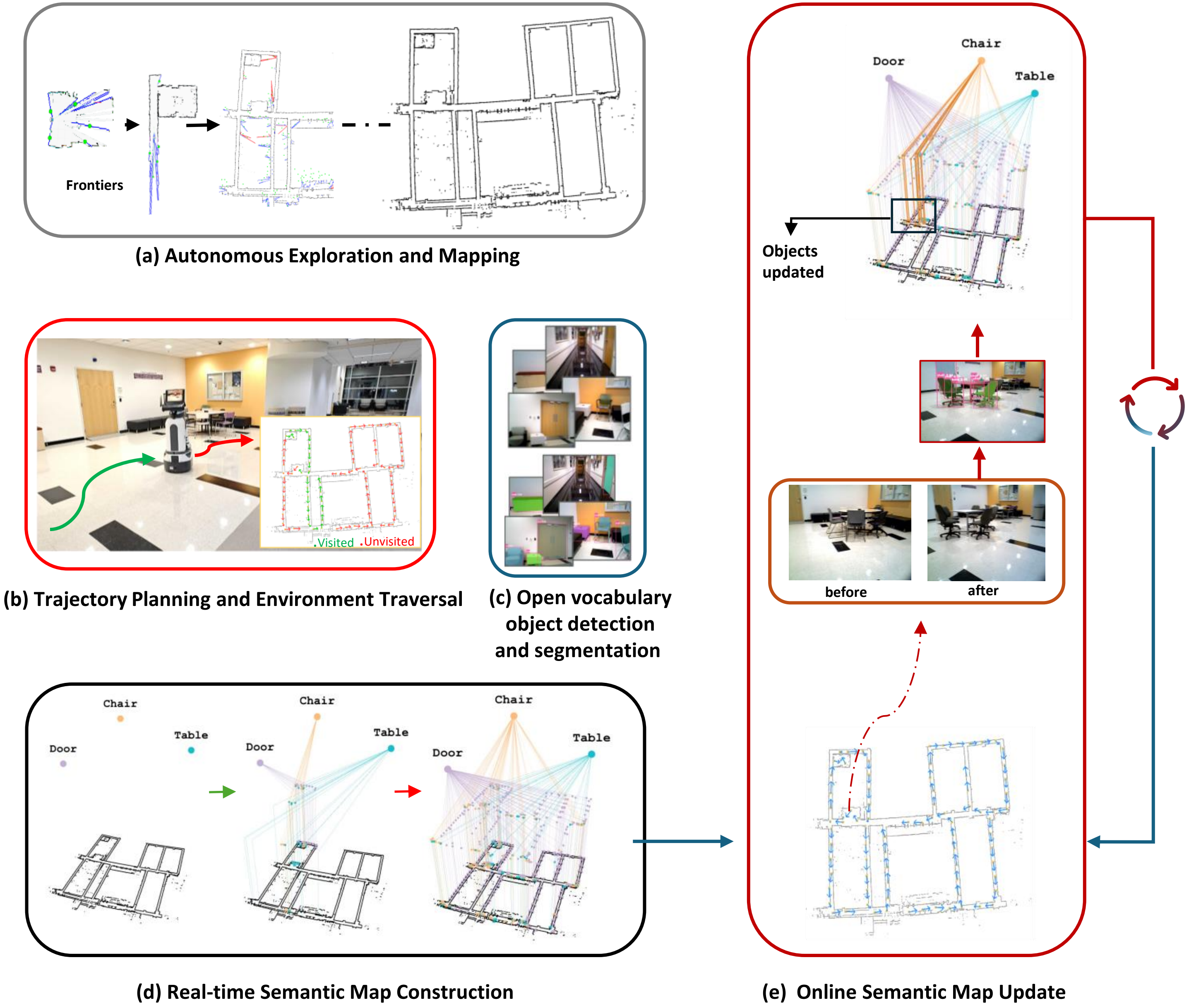
- Popular localization module - AMCL ROS (Adaptive Monte Carlo localization)
- Uses particle filter in determining robot pose in the given map
- Initializes particles all over the map, representing robot's estimated pose
- As the robot moves, particles are resampled based on their current state and robot's action

Autonomous Exploration and Semantic Updating of Large-Scale Indoor Environments with Mobile Robots

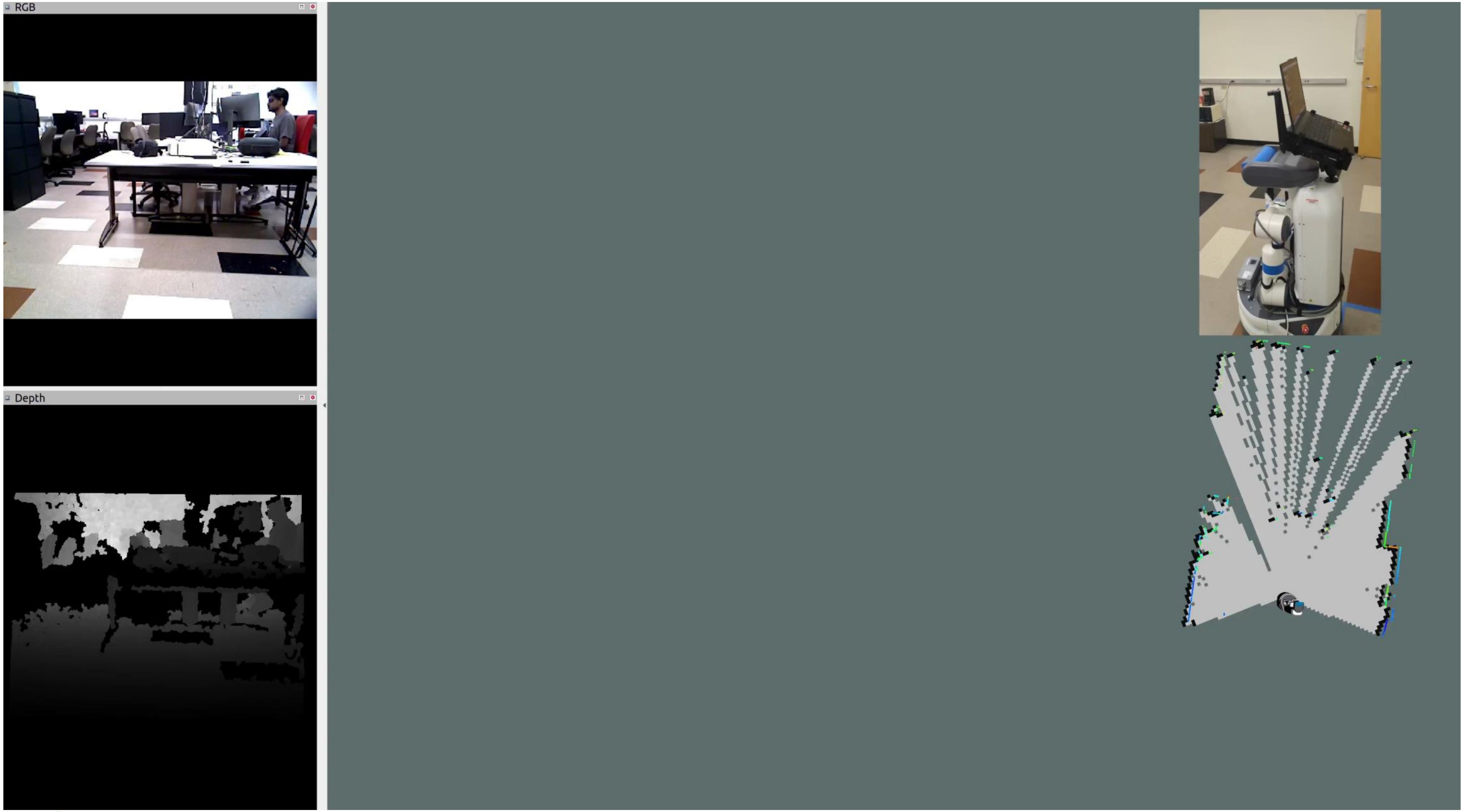
Key Idea:

- Build a map of the environment autonomously
- Include objects semantic along with the geometry in the environment – real-time
- Update the semantic according to changes in environment in real time

This helps in performing downstream tasks like object navigation, Q&A, object manipulation etc.,.


Frontiers
(a) Autonomous Exploration and Mapping
Visited
Unvisited
(b) Trajectory Planning and Environment Traversal
(c) Open vocabulary object detection and segmentation
Chair
Door
Table
(d) Real-time Semantic Map Construction
Objects updated
before
after
(e) Online Semantic Map Update

Autonomous Exploration and Mapping

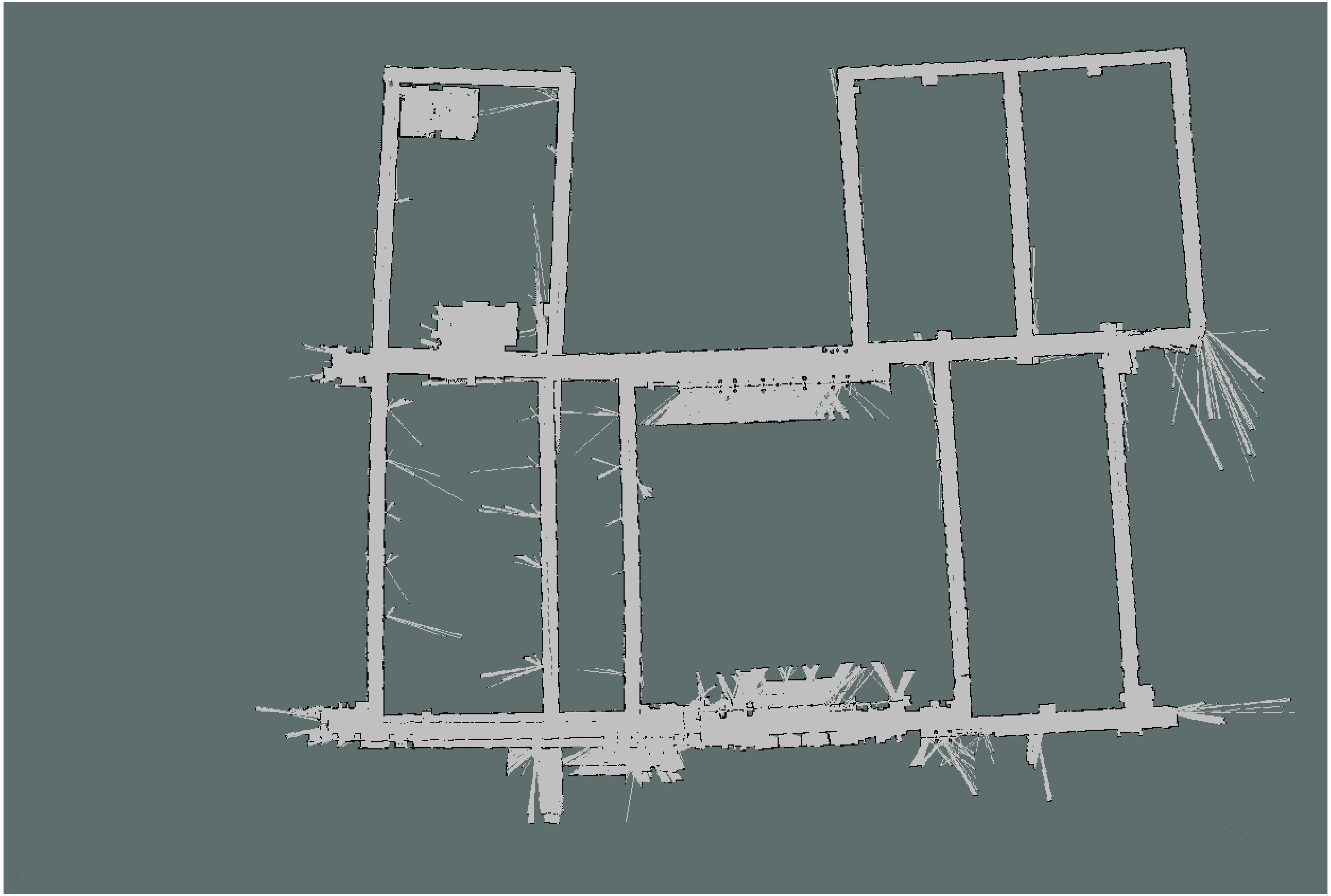


Frontiers

Note: During exploration, Laptop is used only for video capturing

Autonomous, 25x

Autonomous Exploration and Mapping

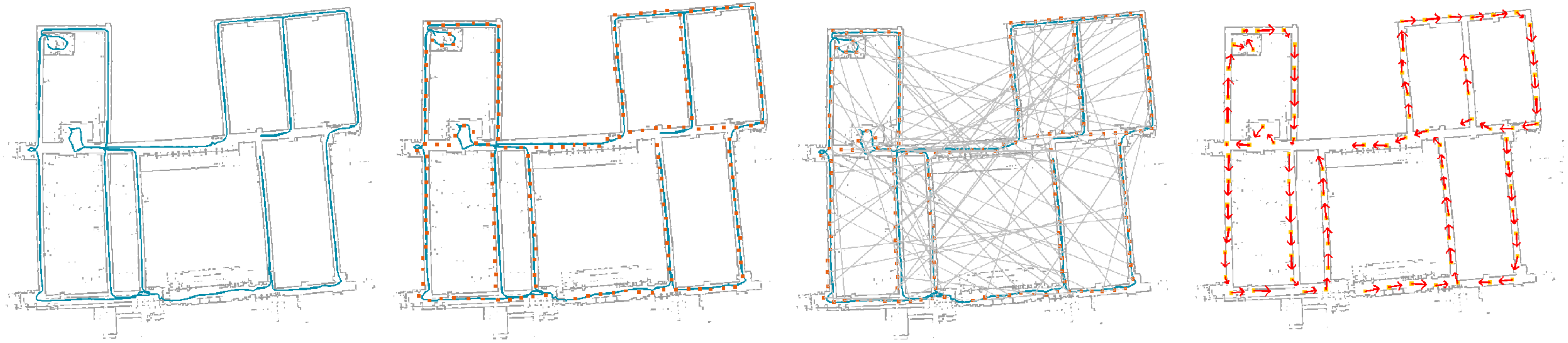


Occupancy map
at
T = 150 minutes

Exploration is completed

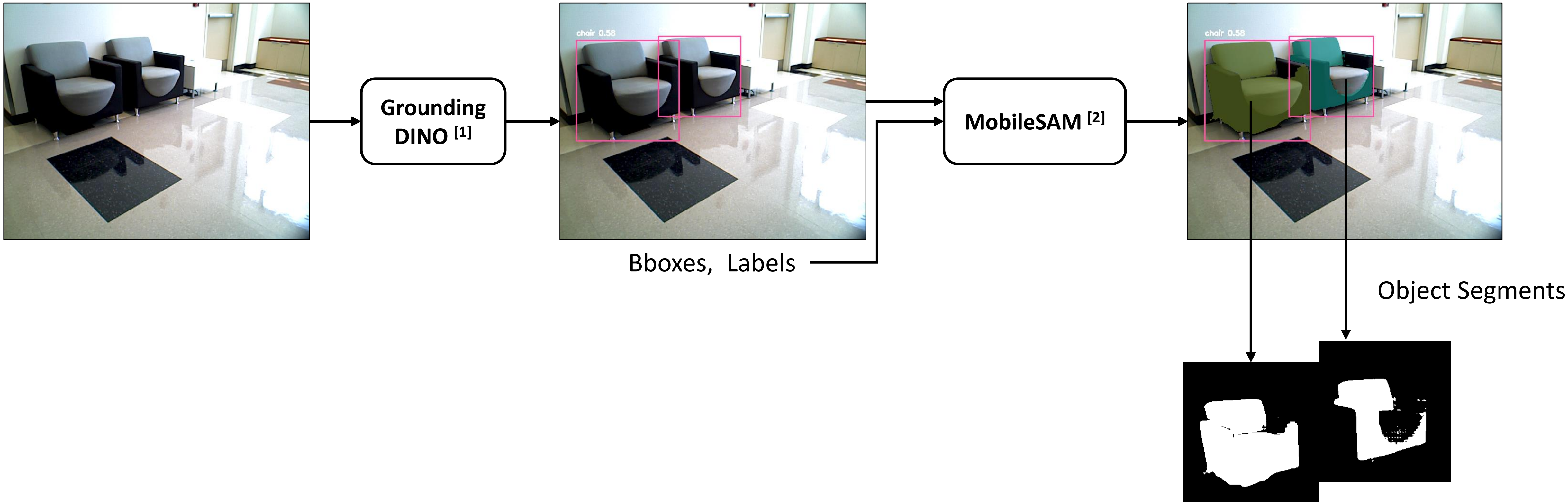
96m x 93m

Environment Traversal Trajectory Planning



(a) Robot exploration trajectory (b) Sampled waypoints (c) TSP formulation (d) Low cost traversal trajectory

Object Detection and Segmentation



[1] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu et al., “Grounding dino: Marrying dino with grounded pre-training for open-set object detection,” arXiv preprint arXiv:2303.05499, 2023.
[2] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, “Faster segment anything: Towards lightweight sam for mobile applications,” arXiv preprint arXiv:2306.14289, 2023.

Realtime Semantic Map Construction

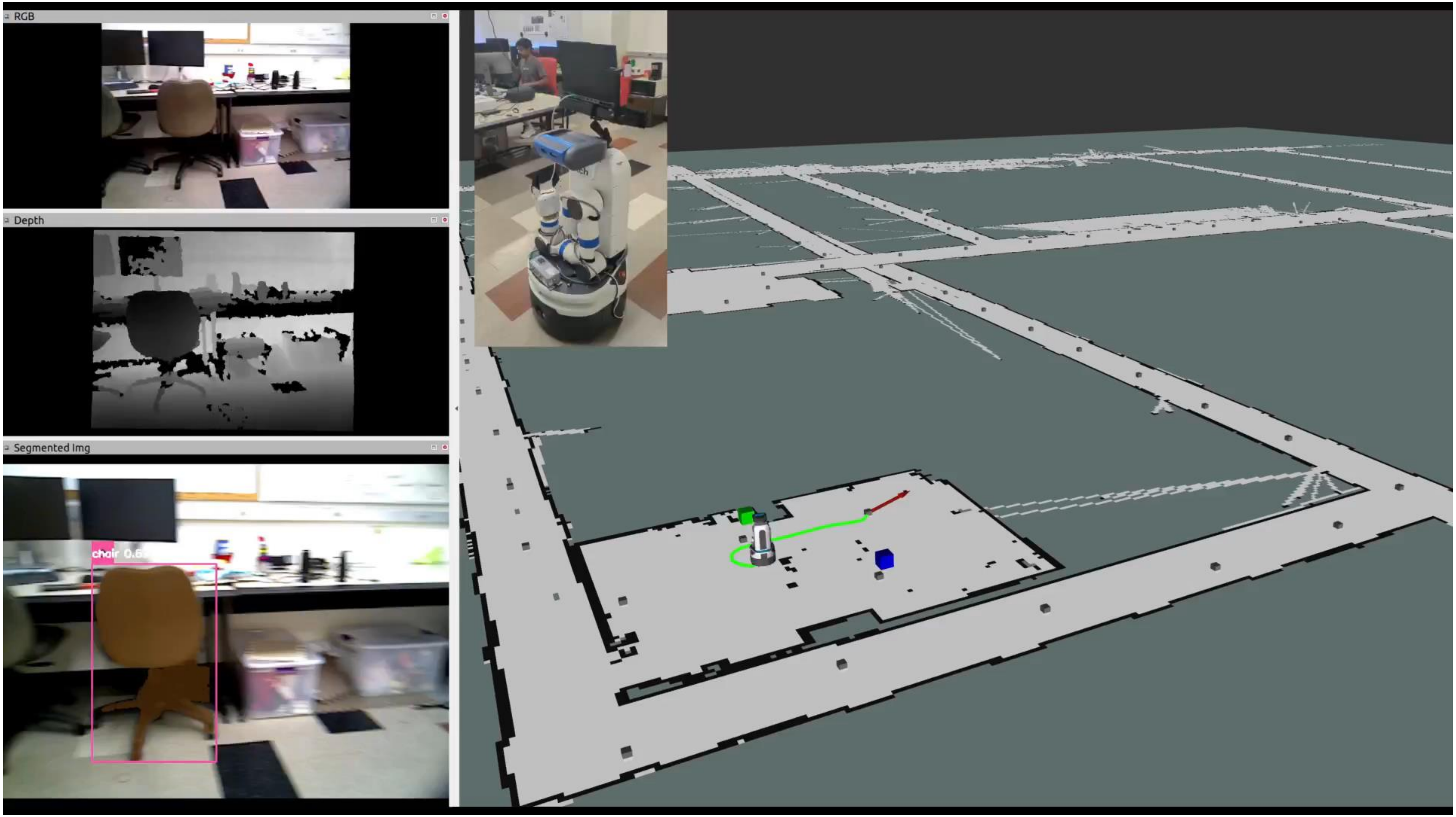
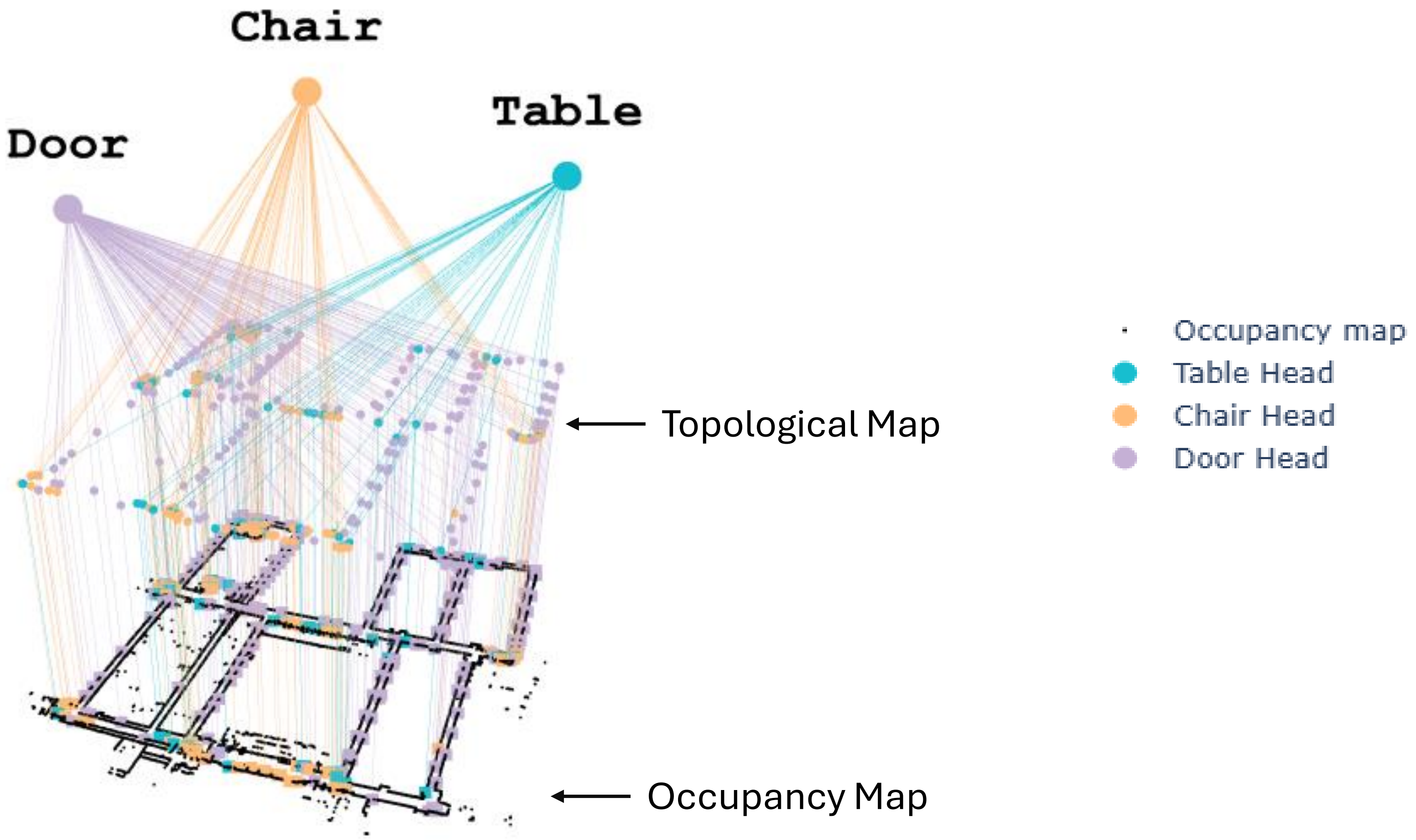


Table Chair Door

Autonomous, 15x

Realtime Semantic Map Construction



Online Update of the Semantic Map

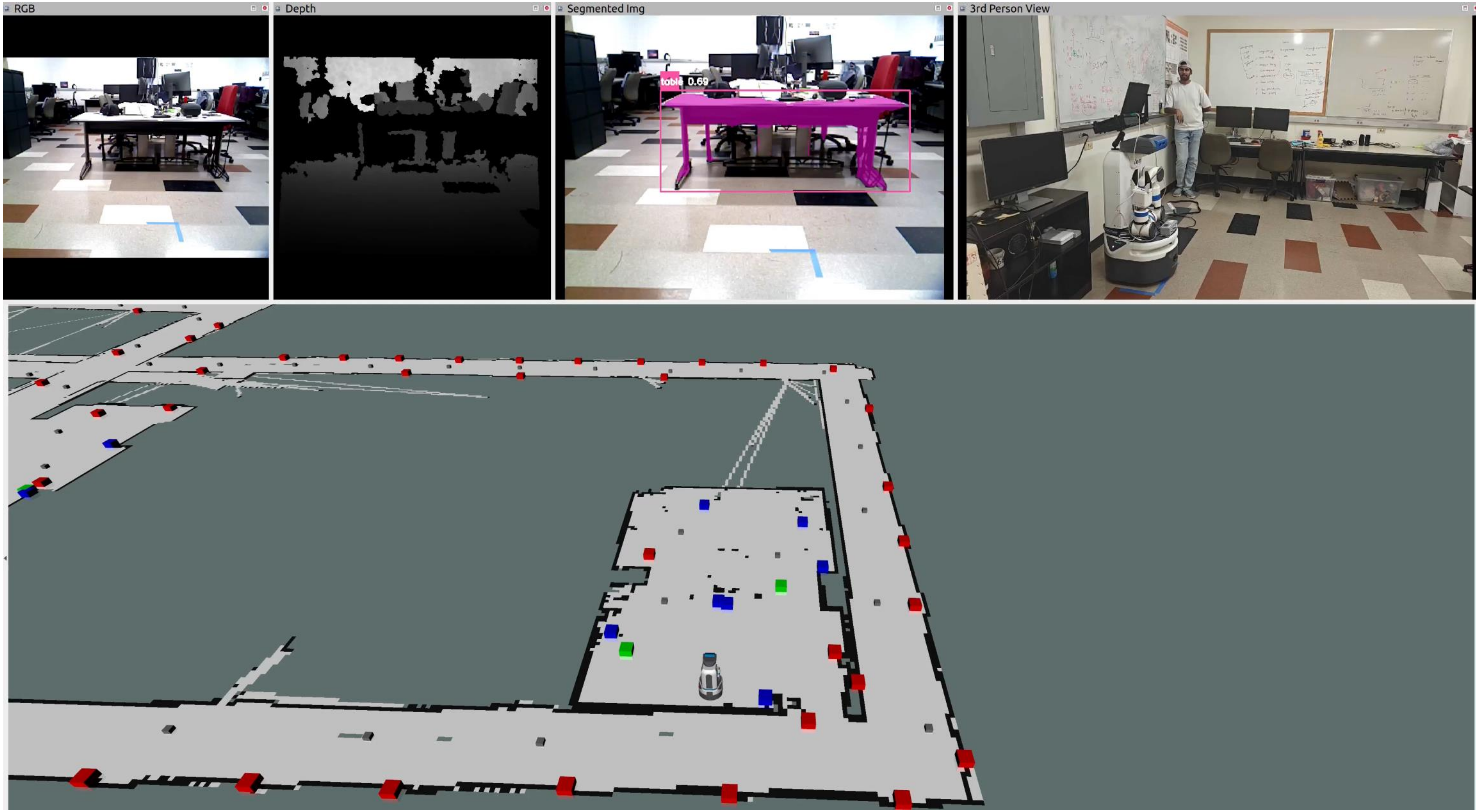
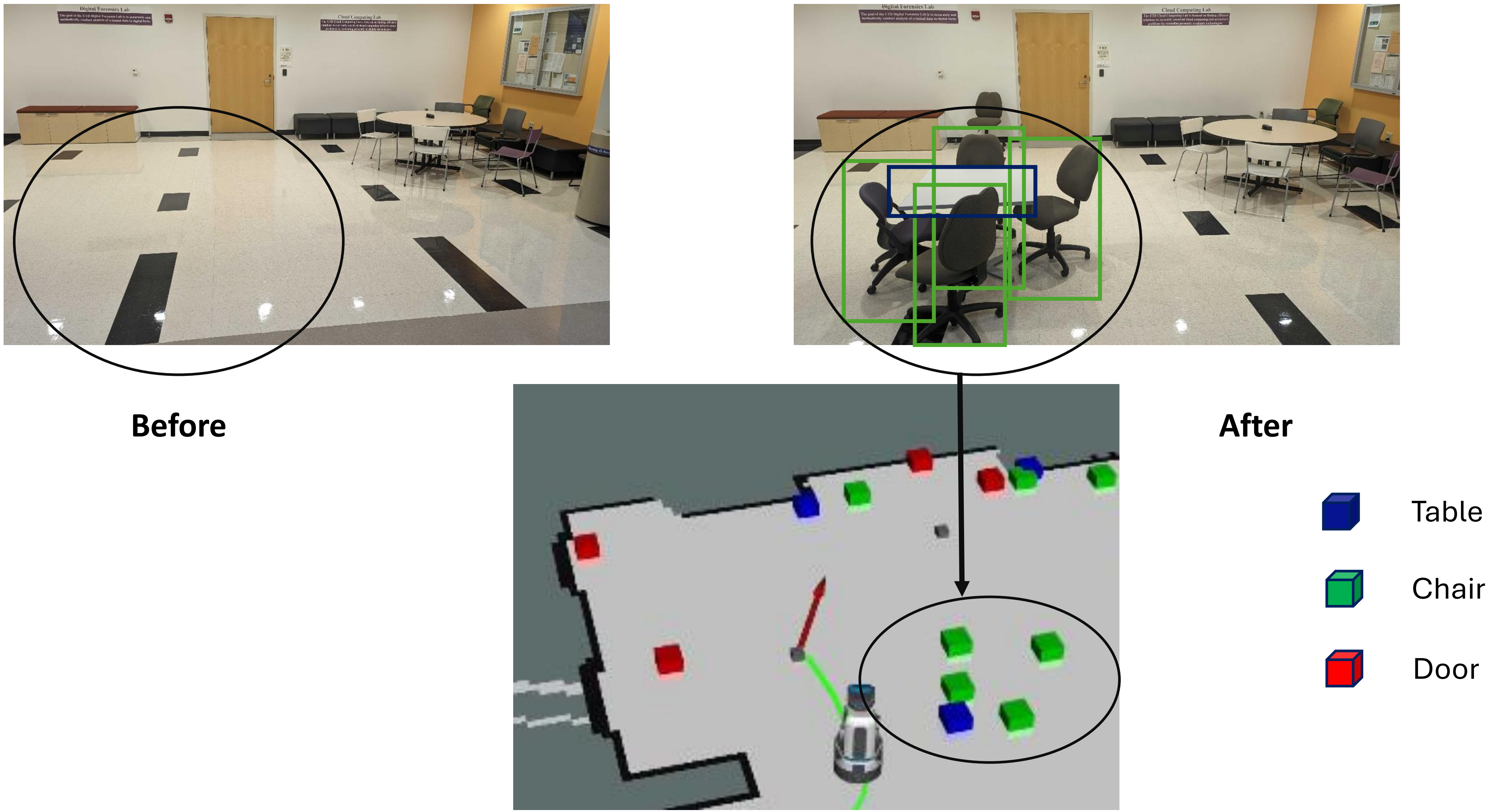


Table Chair Door

Autonomous, 15x



Before
After
Table
Chair
Door

Conclusion

- Real-world Robotic system capable of autonomously exploring unknown environments
- Recognize objects and build semantic map hierarchically real-time
- Update semantic map online to reflect environment changes

References

[1] https://www.diag.uniroma1.it/~nardi/Didattica/CAI/matdid/robot-programming-ROS-introduction-to-navigation.pdf

[2] https://kaiyuzheng.me/documents/navguide.pdf

[3] https://roscon.ros.org/jp/2021/presentations/8.pdf

Thank You !

Website: https://irvlutd.github.io/SemanticMapping

Acknowledgement: DARPA NSF SONY AIR FORCE OFFICE OF SCIENTIFIC RESEARCH

Scan Me !