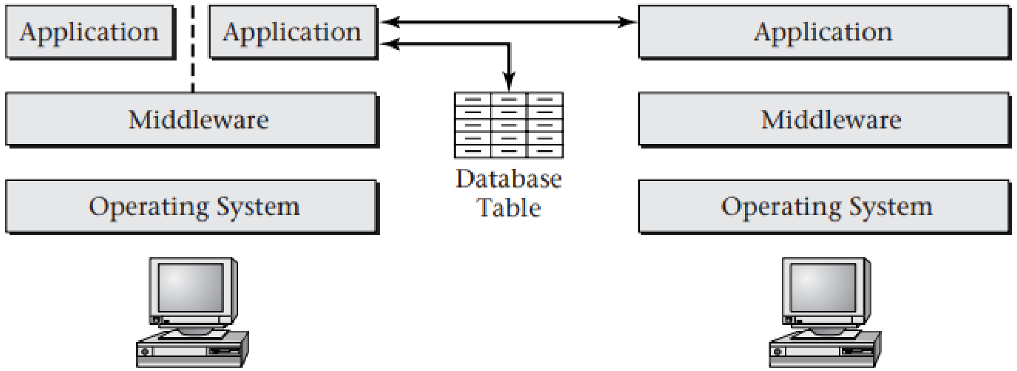
Middleware and ROS

CS646
Prateek Arora

What’s Middleware?

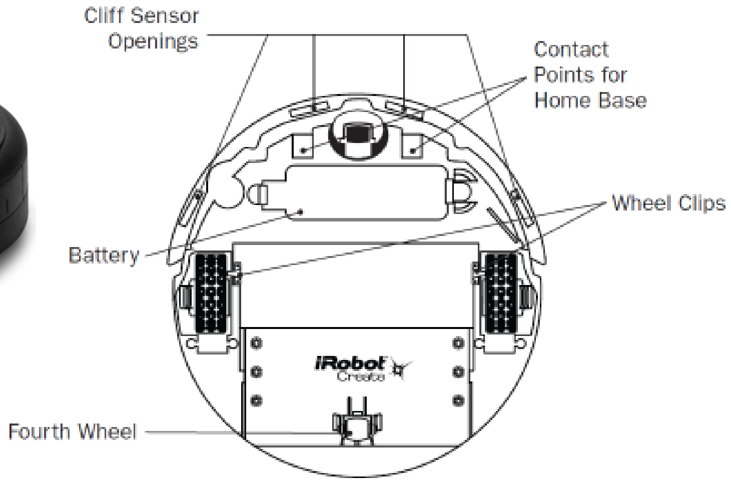
- Connectivity software acting as bridge between OS and applications, tools and databases.
- Helps developers build applications more efficiently
- Provides solutions to make development and deployment of software at large scale and in a cost-effective manner.



Few Middleware Types

- Message-oriented middleware
- Remote procedure call middleware
- Database middleware
- Robotics middleware
- Transactional middleware
- Asynchronous data streaming middleware
- Device middleware

Imagine building your own Roomba Robot



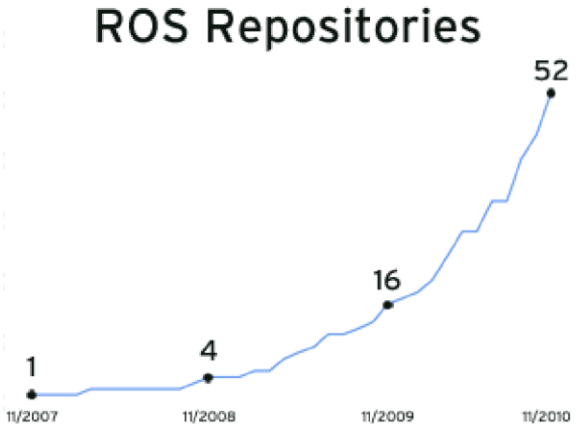
iRobot Create Bottom View

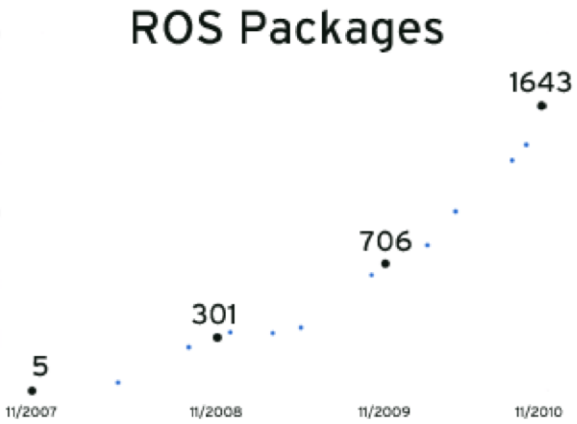
- Reinventing the wheel
  - Device drivers
  - Access to robot’s interface
  - Management of on-board processes
  - Inter-process communication protocols
- Lack of Standards
- Little code reusability
- Developing standard algorithms

New robot -> code from scratch

Robotics Operating System (ROS)

- A software framework (middleware) for programming **robots** and **distributed systems**
- Prototype originated from *Stanford AI research*, officially created and developed by *Willow garage* starting in 2007
- Currently maintained by **Open Source Robotics Foundation**
- Consists of infrastructure, tools, capabilities, and ecosystem
- Exponential adoption





ROS Main Features

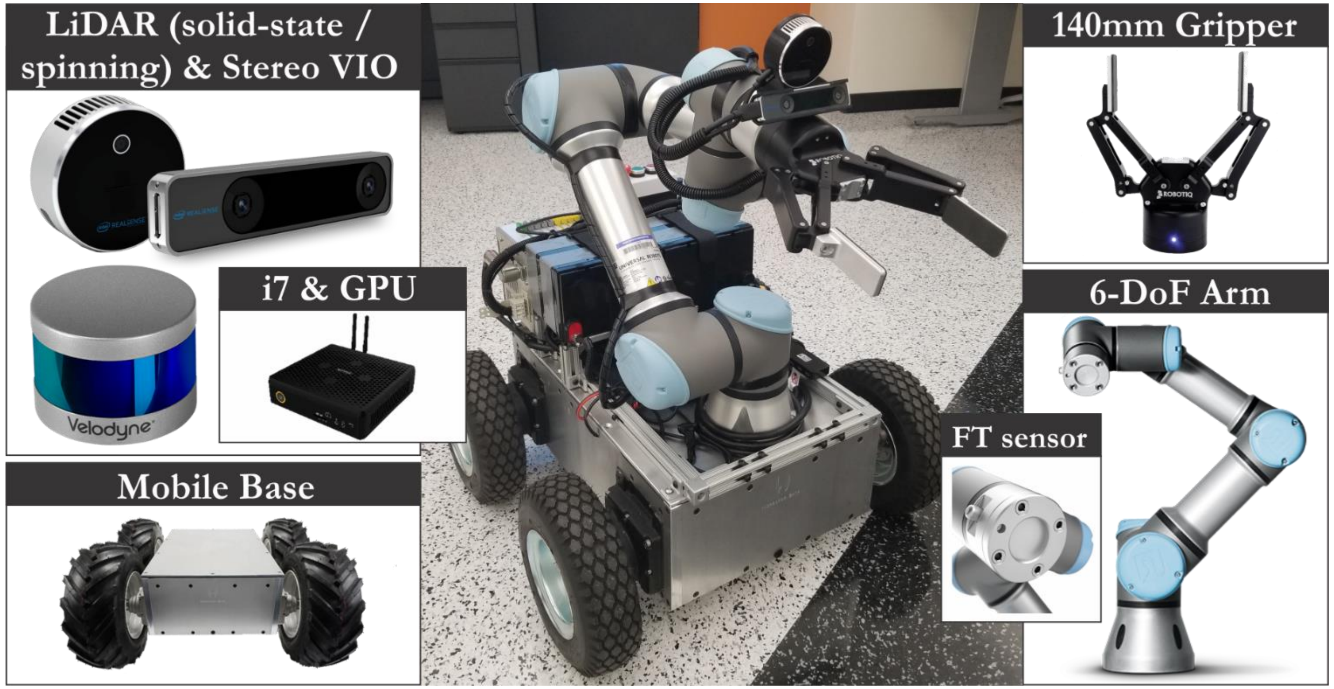
ROS has two “sides”

- The **operating system** side, which provides standard services such as:
  - Hardware abstraction
  - Low-level device control
  - Implementation of commonly used functionality
  - Message-passing between processes
  - Package management
- A **suite of user contributed packages** that implement common robot functionality such as SLAM, panning, perception, vision, manipulation, etc.

How does ROS work?

Core Concepts:

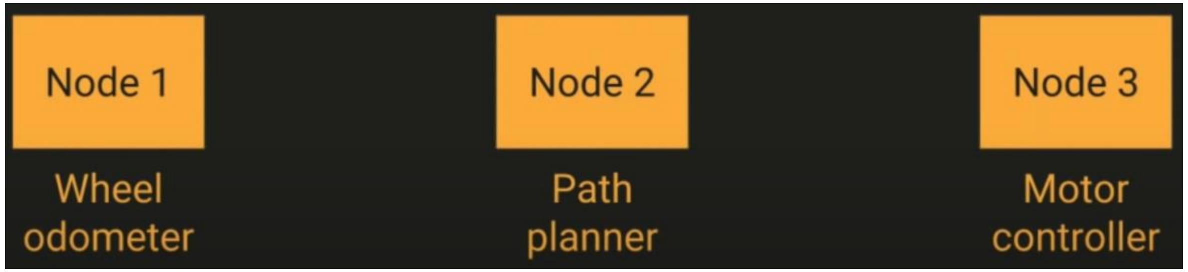
- Nodes
- Messages and Topics
- Services
- Actions
- ROS Master
- Parameters
- Packages and Stacks

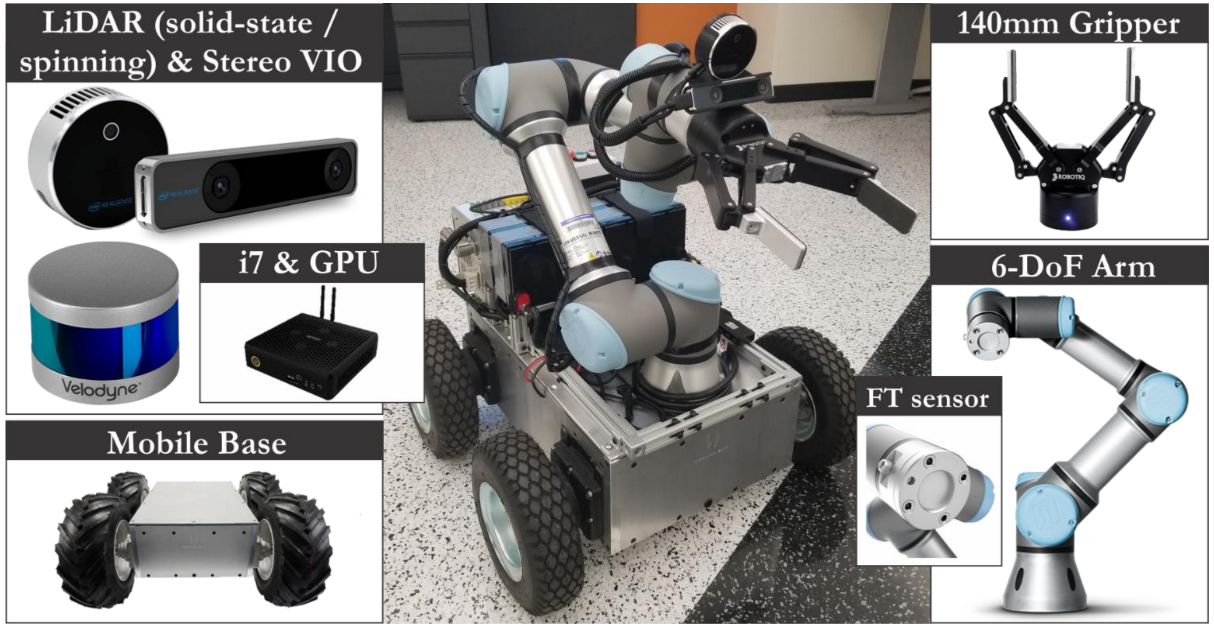


How does ROS work? (cont.)

Core Concepts:

- Nodes
- Messages and Topics
- Services
- Actions
- ROS Master
- Parameters
- Packages and Stacks

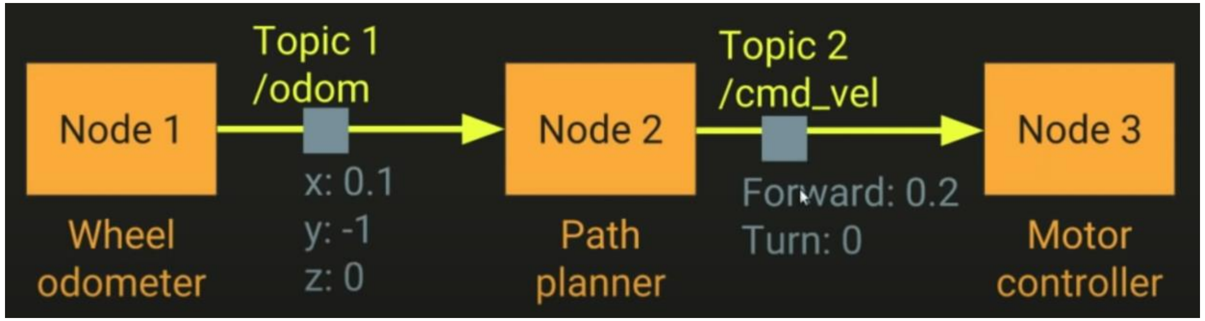




How does ROS work? (cont.)

Core Concepts:

- Nodes
- Messages and Topics
- Services
- Actions
- ROS Master
- Parameters
- Packages and Stacks



Topic 1
/odom

Topic 2
/cmd_vel

Node 1

Wheel odometer

x: 0.1
y: -1
z: 0

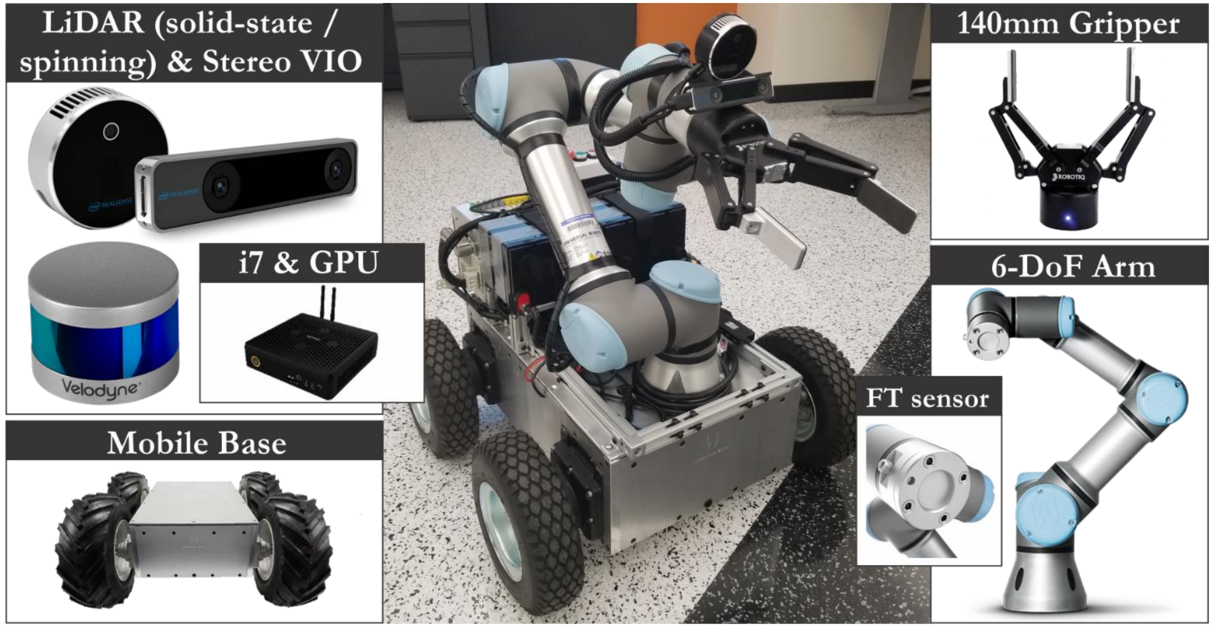
Node 2

Path planner

Forward: 0.2
Turn: 0

Node 3

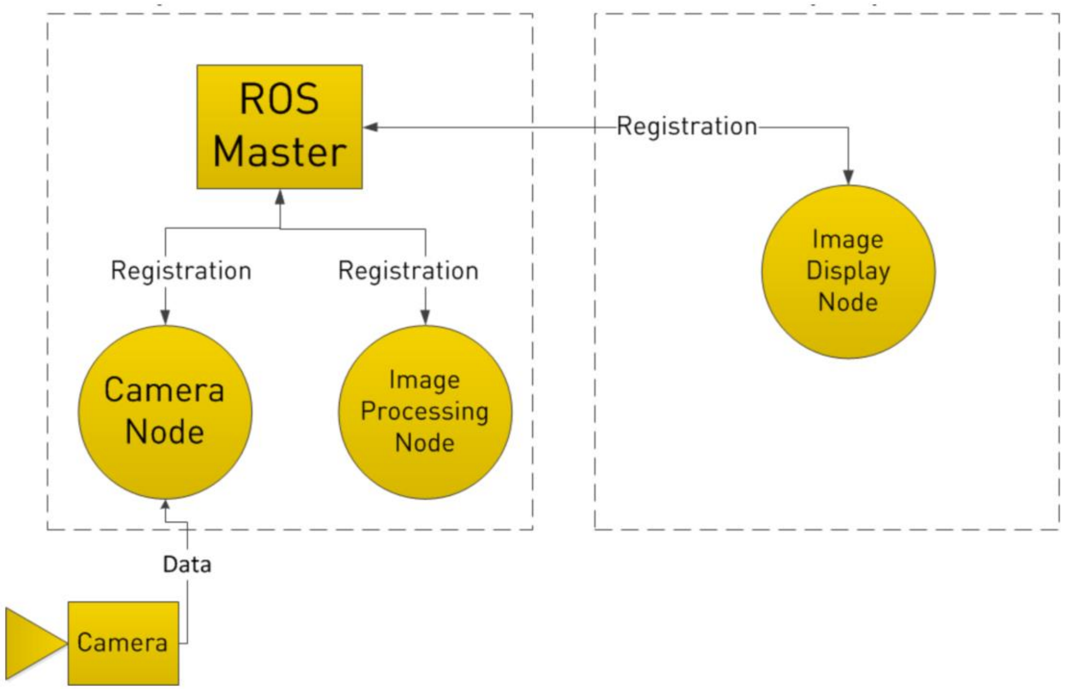
Motor controller



How does ROS work? (cont.)

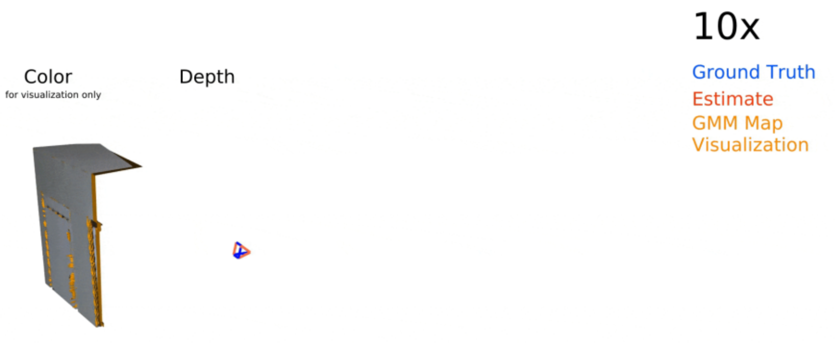
Publishing and Subscribing:

- Any node can publish a message to any topic
- Any node can subscribe to any topic
- Multiple nodes can publish to the same topic
- Multiple nodes can subscribe to the same topic
- A node can publish to multiple topics
- A node can subscribe to multiple topics



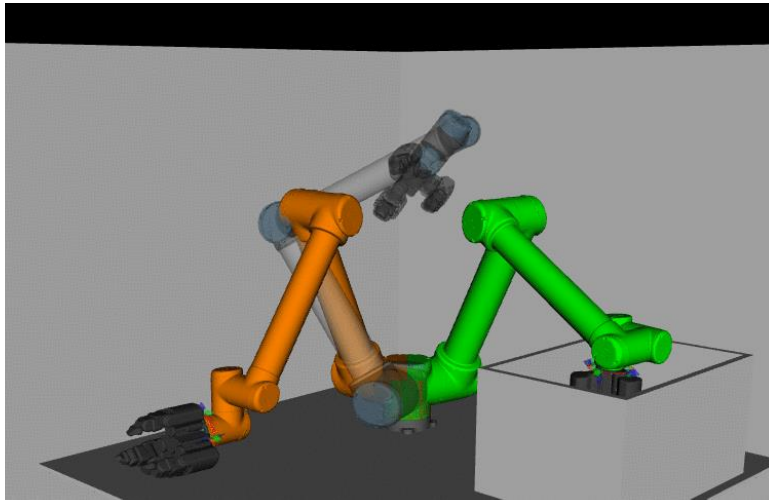
Robotics Research Primer

SLAM

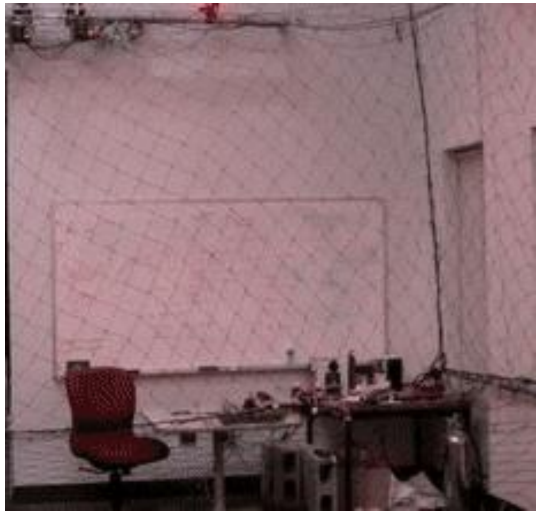


Iterative Closest Distribution:
A Gaussian Mixture Model (GMM) to input point cloud registration pipeline.

Planning

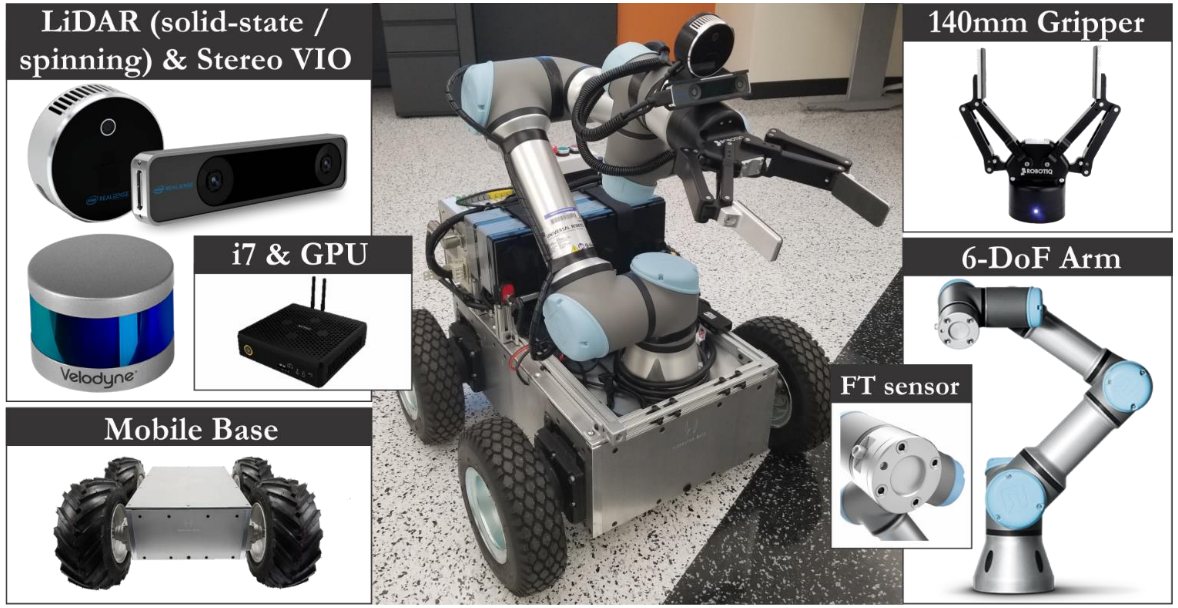


Control



ROS in Robotics Research

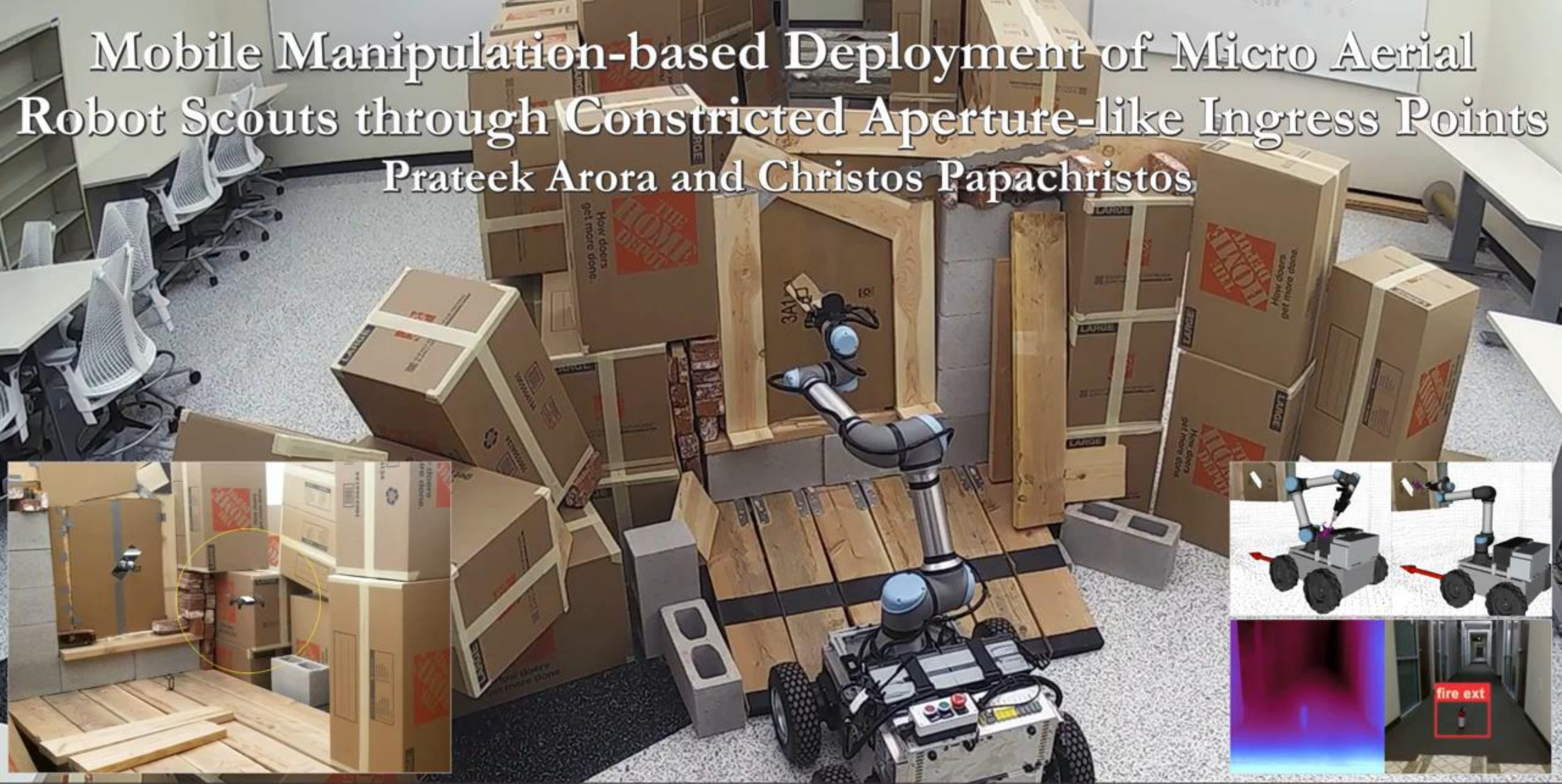
- Differential 4-wheel Drive.
- UR5e 6-DOF Manipulator.
- 2-Finger Parallel Gripper.
- Intel i7-9th & NVIDIA RTX-2070 GPU computer.
- Sensors:
- Intel Realsense L515 solid-state LiDAR.
- Intel Realsense T265 stereo visual-inertial module.
- Force/Torque sensing.



- Capabilities:
- Autonomous real-time Simultaneous Localization And Mapping.
- Autonomous Exploration Path-Planning and Traversability-aware Navigation.
- Autonomous Manipulation Motion Planning.

Mobile Manipulation-based Deployment of Micro Aerial Robot Scouts through Constricted Aperture-like Ingress Points

Prateek Arora and Christos Papachristos



This material is based upon work supported by the
NSF Award: AWD-01-00002751
RI: Small: Learning Resilient Autonomous Flight.

University of Nevada, Reno — Robotic Workers (RoboWork) Lab