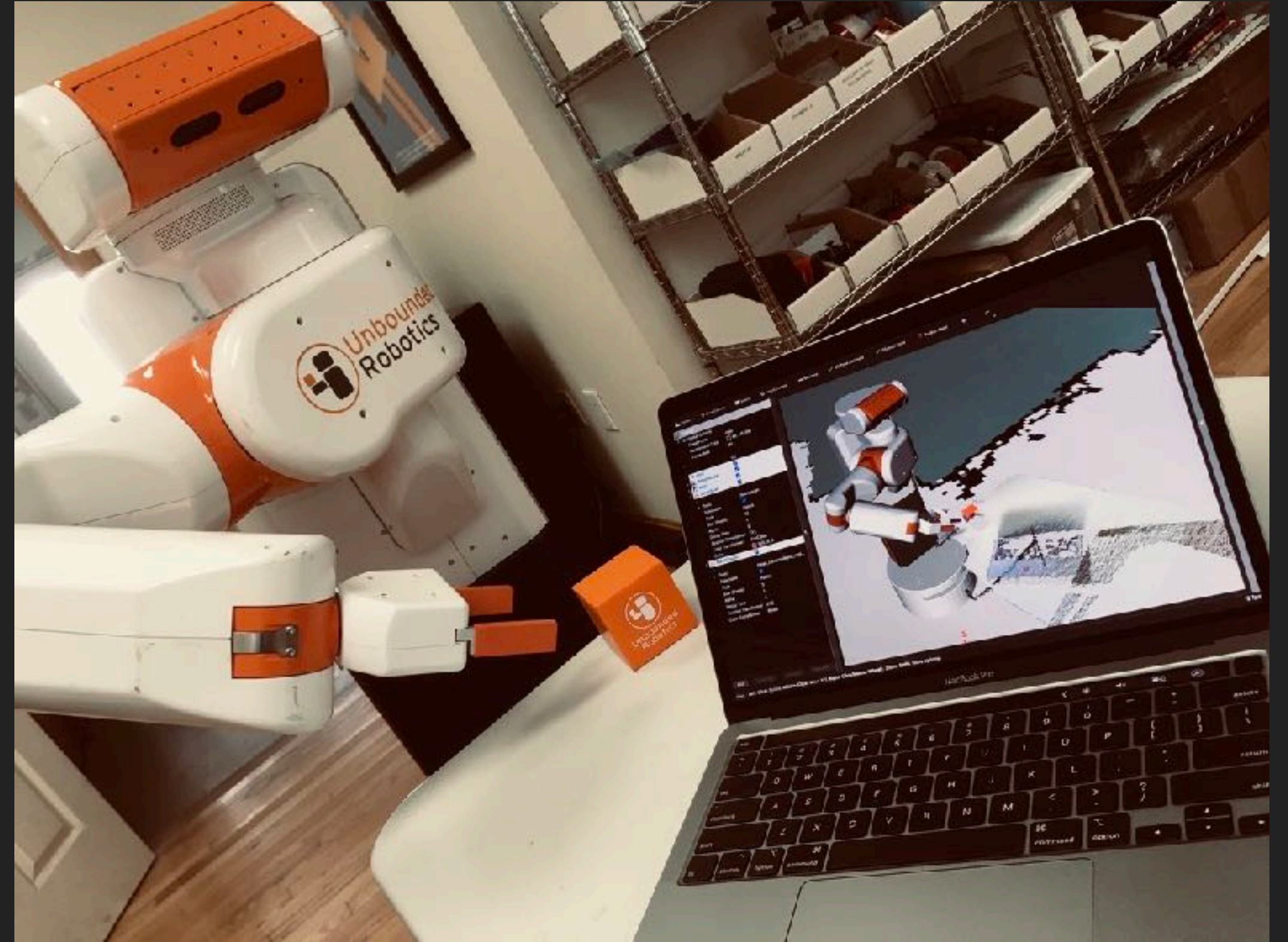


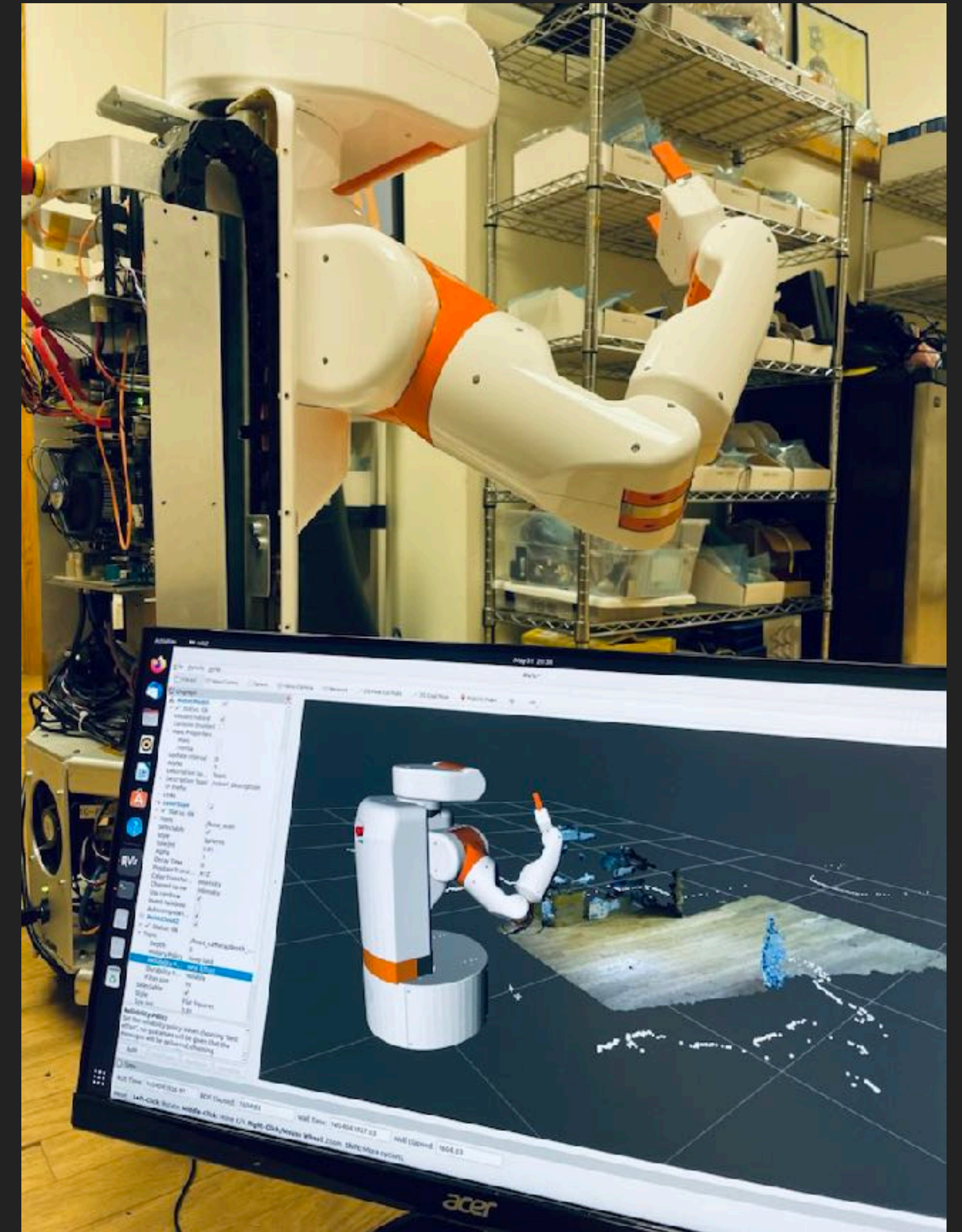
MICHAEL FERGUSON

MIGRATING A MOBILE MANIPULATOR TO ROS 2



TALK OVERVIEW

- ▶ Mobile Manipulation
- ▶ What is the UBR-1?
- ▶ ROS 1 - > ROS 2: Challenges and Features
- ▶ Essential Tools for Mobile Manipulation in ROS 2



UNBOUNDED ROBOTICS UBR-1

- ▶ 2013: Co-founder, CTO
- ▶ 2014: Company bankrupt
- ▶ ...
- ▶ 2020: Robot appears on Craigslist

CL stockton > for sale > electronics - by owner

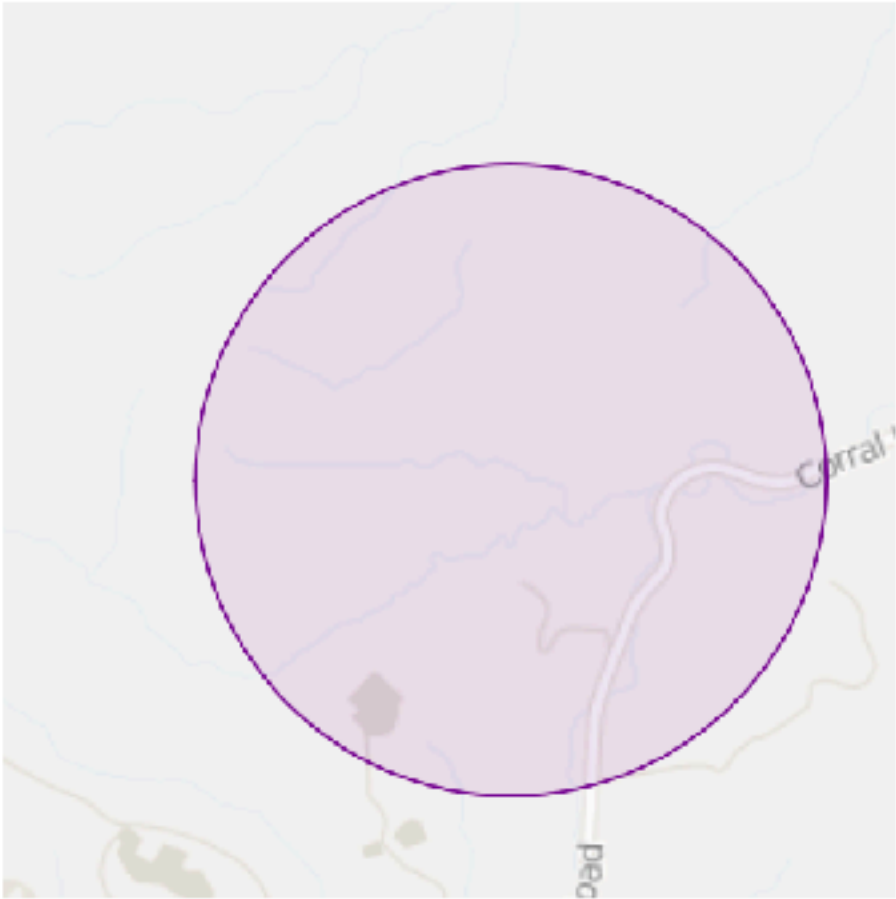

favorite hide flag

Posted about 13 hours ago on: 2020-04-25 16:29

Contact Information:

Robot prototype - \$5000 (Tracy, CA)


image 1 of 10



condition: new

This prototype robot is now for sale.
It worked when it was put in the box and is from a failed robotic company.
They were going to sell them for \$50,000 and this prototype is yours for the low price of \$5,000.
There may be some spare parts too.
This robot comes with his (her) own storage box.
I believe the lithium battery needs to be recharged at this point.

QR Code Link to This Post



SAVE THE ROBOTS: MORE THAN JUST A BEER NAME

- ▶ Updated from Indigo->Melodic
- ▶ Melodic->Noetic: Python3
- ▶ How hard can it be to go to ROS 2?
- ▶ 4 years later - giving this talk!



ROS 1 - > ROS 2

- ▶ More than just an API change
 - ▶ Different threading models
 - ▶ More fully featured (but also more complex/verbose)
 - ▶ Some late arriving features (lazy subscribers, etc)
- ▶ Many packages took the opportunity to largely re-write/re-architect (ex: Nav2)

UBR-1 ON ROS2 TIMELINE

- ▶ Started with ROS 2 Foxy / 20.04
- ▶ Ported to Humble / 22.04
- ▶ Ported to Iron
- ▶ Ported to Jazzy / 24.04

**USE DOCKER/ROCKER TO
OVERCOME OPERATING
SYSTEM MISMATCH**



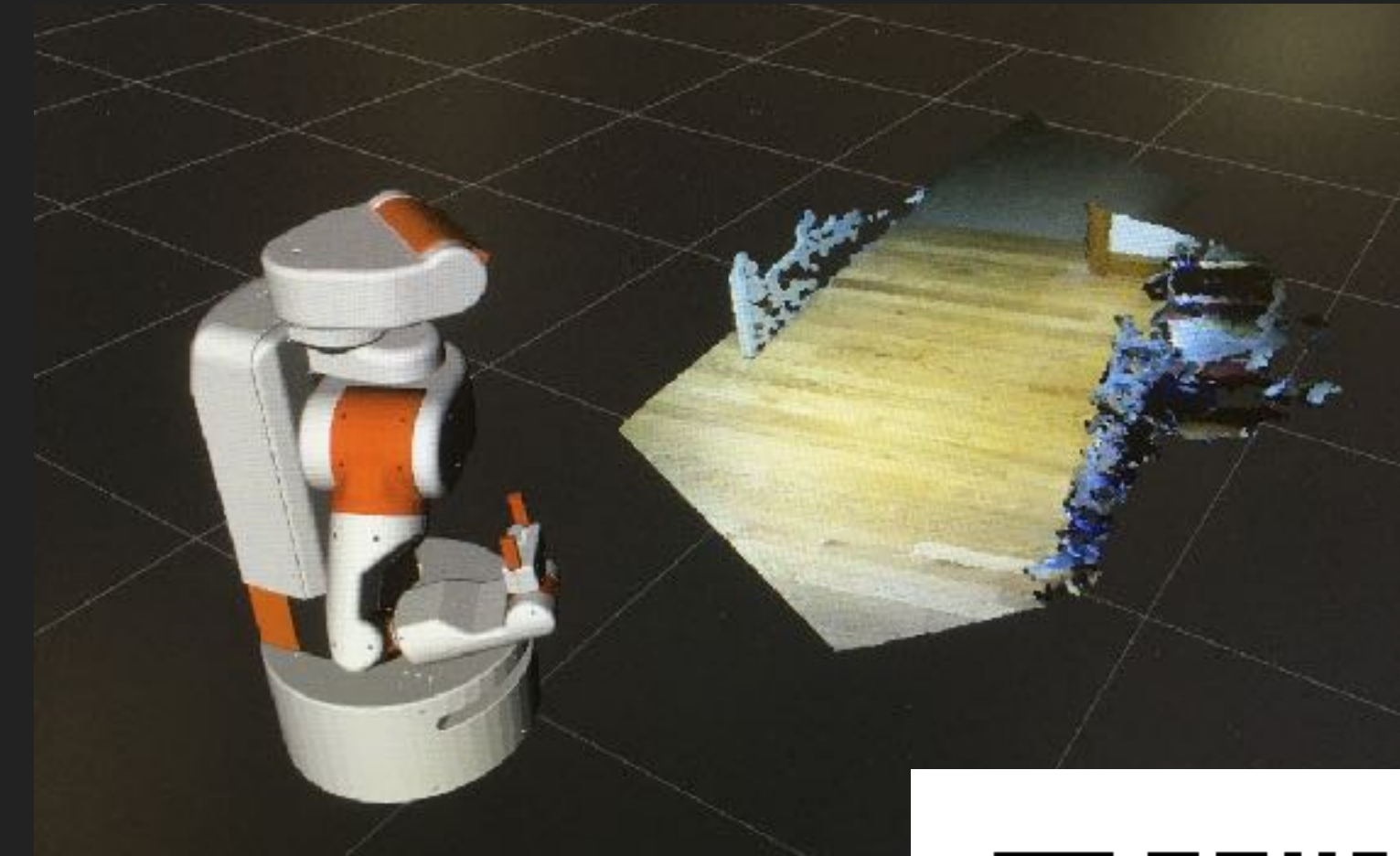
HARDWARE DRIVERS IN ROS2

- ▶ Still a bit like the wild west!
 - ▶ Some are not merged into mainline - have to find the random fork! Stack Exchange and forums help to find these.
 - ▶ Some are not released into Debian packages!
 - ▶ Some features still arriving (lazy subscribers)!
- ▶ I ended up becoming maintainer of urg_node, openni2_camera packages.



PORTING OPENNI2_CAMERA TO ROS 2

- ▶ Nodelets -> Components (well documented)
 - ▶ Derived from `rclcpp::Node::SharedPtr`
 - ▶ `shared_from_this()` limitations
- ▶ Lazy subscribers - added in Iron/Jazzy. Workaround with timers.
- ▶ `openni2_launch` still not ported

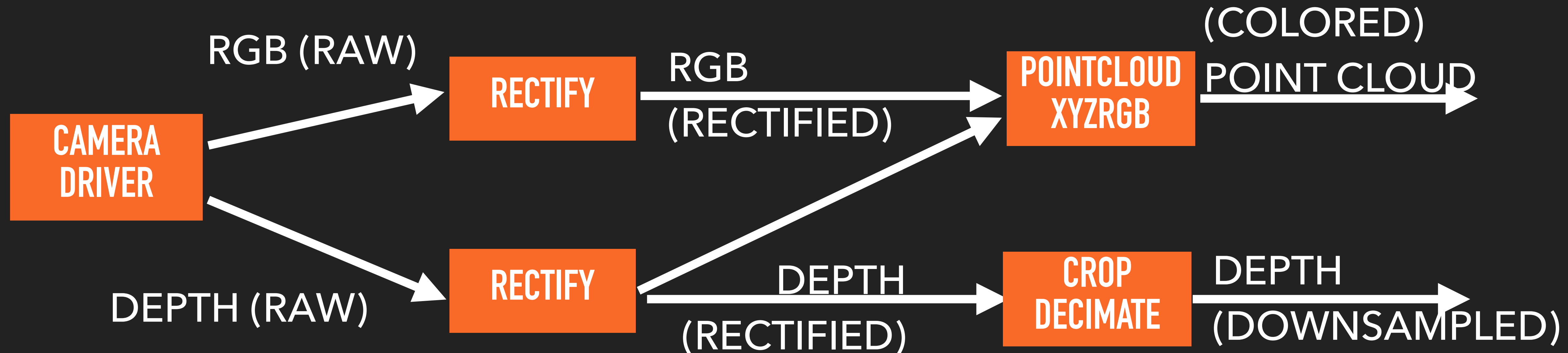


FULL WRITE UP:



IMAGE PIPELINE

- ▶ Transform and preprocess image data
- ▶ Proper ROS 2 components (were nodelets in ROS 1)
 - ▶ Easier to introspect and debug - same performance boost!

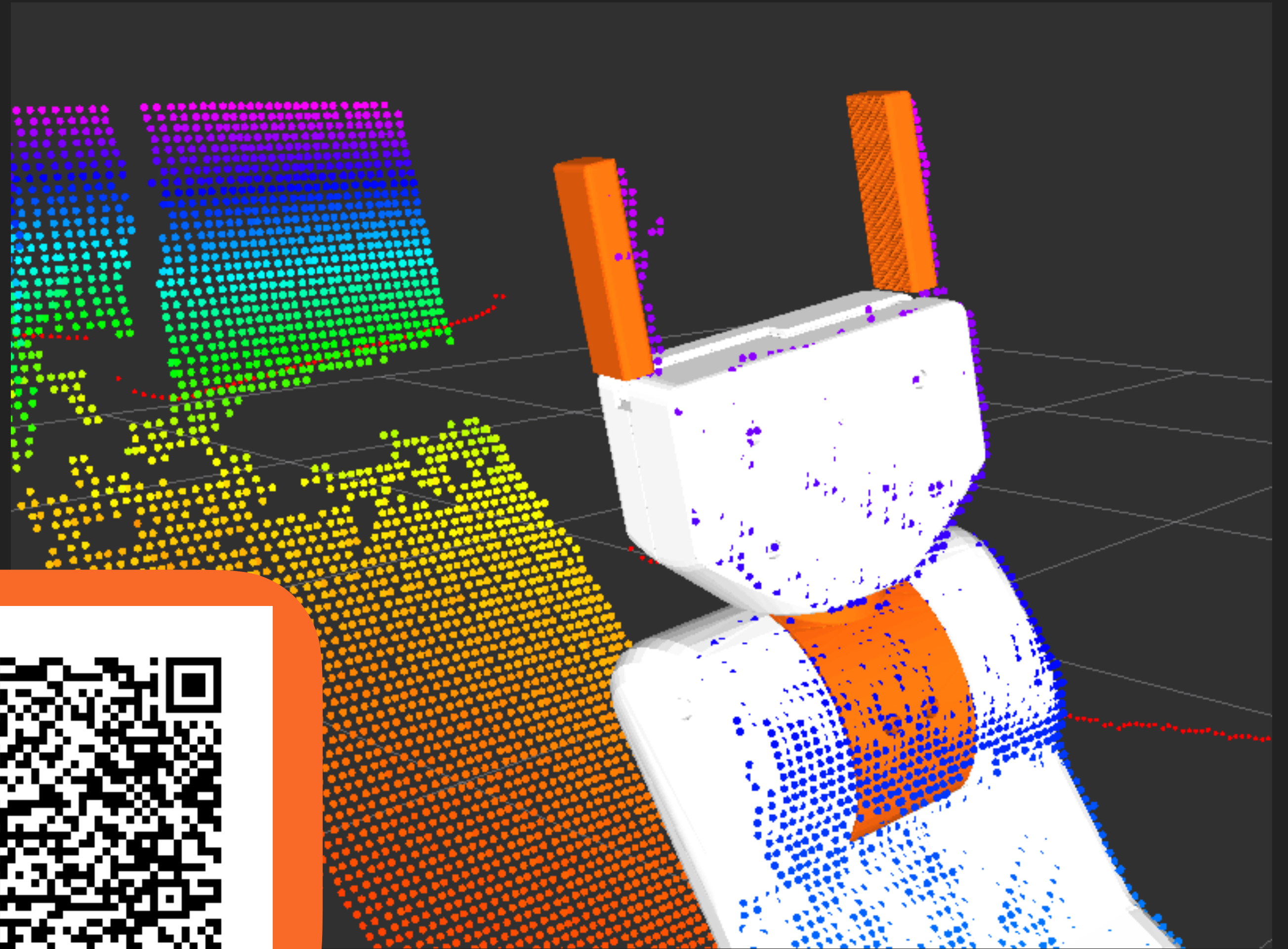


LAUNCH FILES

- ▶ Python-based launch files:
 - ▶ Awesome for complex robots!
 - ▶ Can be overly verbose.
 - ▶ Somewhat poorly documented...

ESSENTIAL MOBILE MANIPULATION TOOLS: CALIBRATION

- ▶ camera_calibration
- ▶ robot_calibration
- ▶ Update all sorts of API when migrating to ROS 2
 - ▶ Parameters had to change format due to XML arrays
 - ▶ See [ros2_cookbook](#):



ESSENTIAL MOBILE MANIPULATION TOOLS: CALIBRATION



ESSENTIAL MOBILE MANIPULATION TOOLS: MAPPING AND LOCALIZATION

- ▶ Some mapping packages not ported to ROS 2 (slam_karto)
- ▶ slam_toolbox does a very good job of mapping
 - ▶ Maps are “transient local” (what was “latched” in ROS 1)
 - ▶ Had to manually adjust the free_thresh for map to look correct
- ▶ Some notes on tuning AMCL on my blog



ESSENTIAL MOBILE MANIPULATION TOOLS: NAVIGATION

- ▶ Nav2 is a major rewrite of the ROS 1 Navigation Stack
- ▶ Behavior Trees allows changing the behavior of what used to be move_base
 - ▶ No longer just plan, control, recover
 - ▶ Newly added features like auto docking (See Steve Macenksi's talk at 4:40)

NAVIGATION: CUSTOM CONTROLLERS

- ▶ UBR-1 uses graceful_controller, porting to ROS 2 involved:
 - ▶ Changes to how parameters are defined, since they used to be loaded via XML arrays.
 - ▶ Updates for different controller API.
- ▶ Fairly straight forward - and new controller has less code thanks to external components like goal checker (leads to less code duplication between controllers).

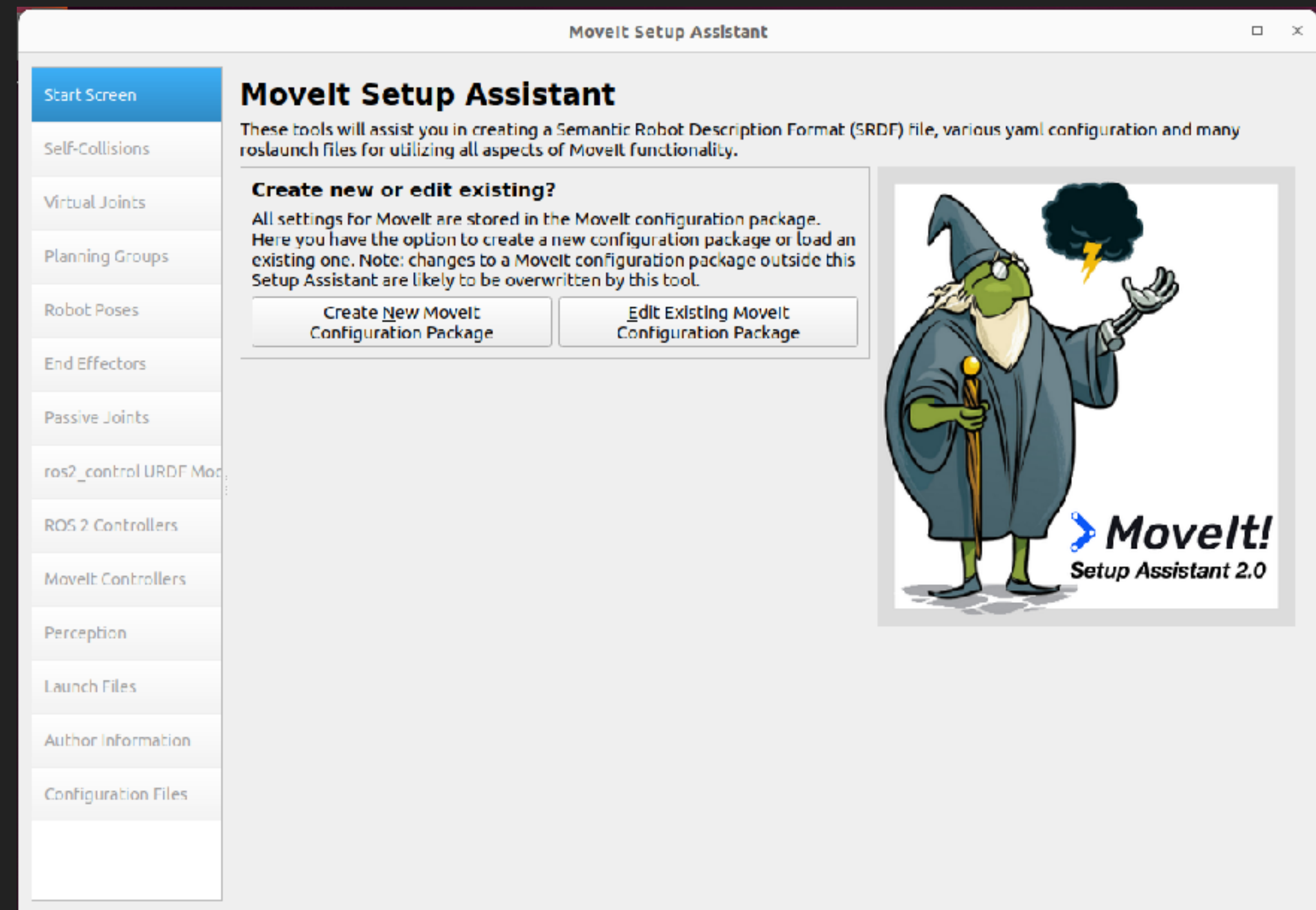
NAVIGATION: CUSTOM COSTMAP LAYERS

- ▶ Mobile manipulator wants to tilt head up and down while navigating in order to get better field of view from sensors
- ▶ Timing isn't always perfectly aligned - need to find the ground plane and remove it for better performance



ESSENTIAL MOBILE MANIPULATION TOOLS: MANIPULATION

- ▶ MoveIt2 - heavy changes to API from ROS 1
- ▶ Pick & Place => MTC
- ▶ Setup Assistant now available!



ESSENTIAL TOOLS: MOVEIT SETUP ASSISTANT

Start Screen

Self-Collisions

Virtual Joints

Planning Groups

Robot Poses

End Effectors

Passive Joints

ros2_control URDF Moc...

ROS 2 Controllers

MoveIt Controllers

Perception

Launch Files

Author Information

Configuration Files

Optimize Self-Collision Checking

This searches for pairs of robot links that can safely be disabled from collision checking, decreasing motion planning time. These pairs are disabled when they are always in collision, never in collision, in collision in the robot's default position, or when the links are adjacent to each other on the kinematic chain. Sampling density specifies how many random robot positions to check for self collision.

Sampling Density: Low

High 10000

Min. collisions for "always"-colliding pairs: 95%

Generate Collision Matrix

	Link A	Link B	Disabled	Reason to Disab
1	base_laser...	bellows_link	✓	Never in ...
2	base_laser...	elbow_Flex...	✓	Never in ...
3	base_laser...	estop_link	✓	Never in ...
4	base_laser...	fixed_bello...	✓	Never in ...
5	base_laser...	fixed_torso...	✓	Never in ...
6	base_laser...	head_came...	✓	Never in ...
7	base_laser...	head_pan_l...	✓	Never in ...
8	base_laser...	head_tilt_link	✓	Never in ...
9	base_laser...	left_grippe...	✓	Never in ...
10	base_laser...	right_grippe...	✓	Never in ...
11	base_laser...	sho...	✓	Never in ...
12	base_laser...	sho...	✓	Never in ...
13	base_laser...	tors...	✓	Never in ...
14	base_laser...	upp...	✓	Never in ...

link name filter

Define Planning Groups

Create and edit 'joint model' groups for your robot based on joint collections, link collections, kinematic chains or subgroups. A planning group defines the set of (joint, link) pairs considered for planning and collision checking. Define individual groups for each subset of the robot you want to plan for. Note: when adding a link to the group, its parent joint is added too and vice versa.

Current Groups

arm

joints

Links

Chain

torso_lift_link -> wrist_roll_link

Subgroups

arm_with_torso

joints

Links

Chain

base_link -> wrist_roll_link

Subgroups

gripper

joints

left_gripper_joint - Prismatic

right_gripper_joint - Prismatic

Links

Chain

Subgroups

MoveIt Setup Assistant

Start Screen

Self-Collisions

Virtual Joints

Planning Groups

Robot Poses

End Effectors

Passive Joints

ros2_control URDF Moc...

ROS 2 Controllers

MoveIt Controllers

Perception

Launch Files

Author Information

Configuration Files

Define Planning Groups

Create and edit 'joint model' groups for your robot based on joint collections, link collections, kinematic chains or subgroups. A planning group defines the set of (joint, link) pairs considered for planning and collision checking. Define individual groups for each subset of the robot you want to plan for. Note: when adding a link to the group, its parent joint is added too and vice versa.

Edit Planning Group 'arm'

Kinematics

Group Name:

arm

Kinematic Solver:

kd/_kinematics_plugin/KDLKinematicsPlugin

Kin. Search Resolution:

0.005

Kin. Search Timeout (sec):

0.005

Kin. parameters file:

...

OMPL Planning

Group Default Planner:

None

Delete Group

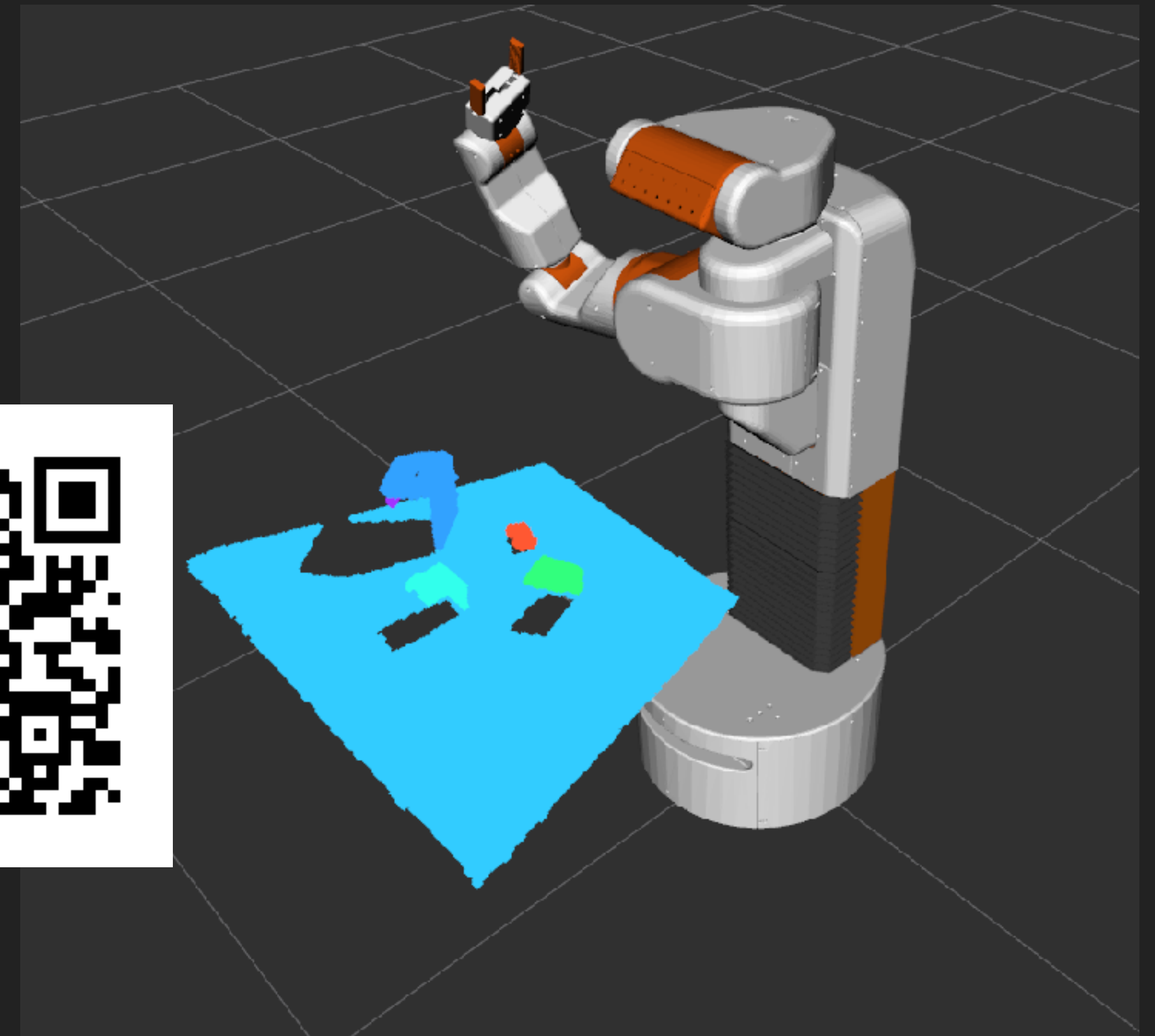
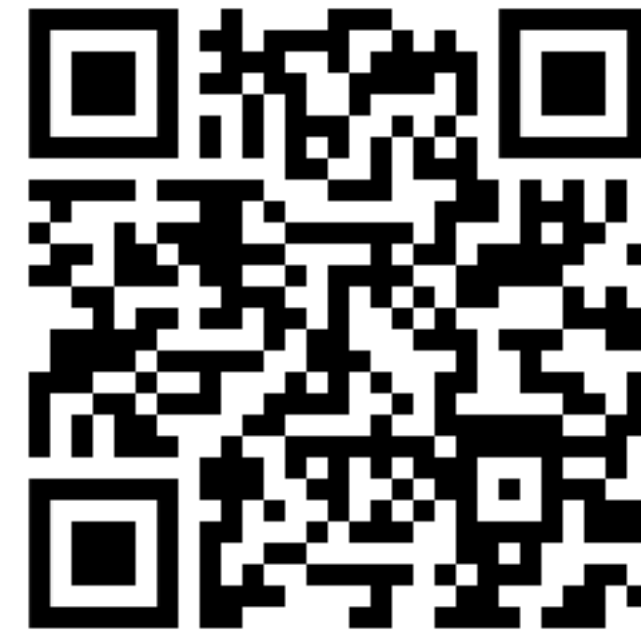
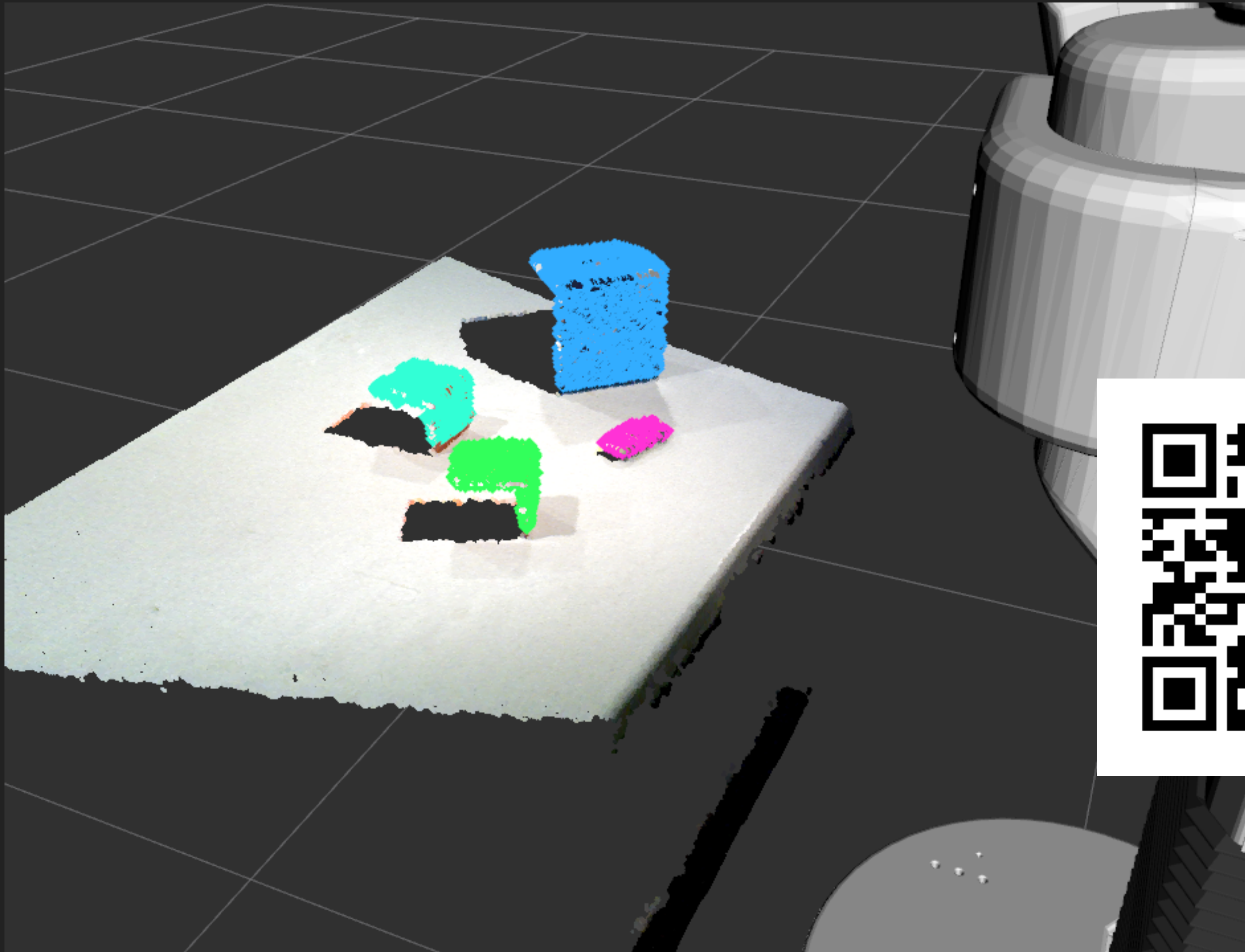
Save

Cancel

3D View

SEE TUTORIAL

ESSENTIAL MOBILE MANIPULATION TOOLS: SIMPLE GRASPING



ESSENTIAL MOBILE MANIPULATION TOOLS: MOVEIT2

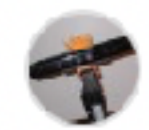


ROS 2: CONTINUED STRUGGLES

- ▶ QoS and DDS Reliability
 - ▶ Maybe Zenoh? Help test it!
- ▶ Documentation is scattered/sparse



MIGRATING A MOBILE MANIPULATOR TO ROS 2



ros2_cookbook

Public

Pin

Unwatch 21

Fork 93

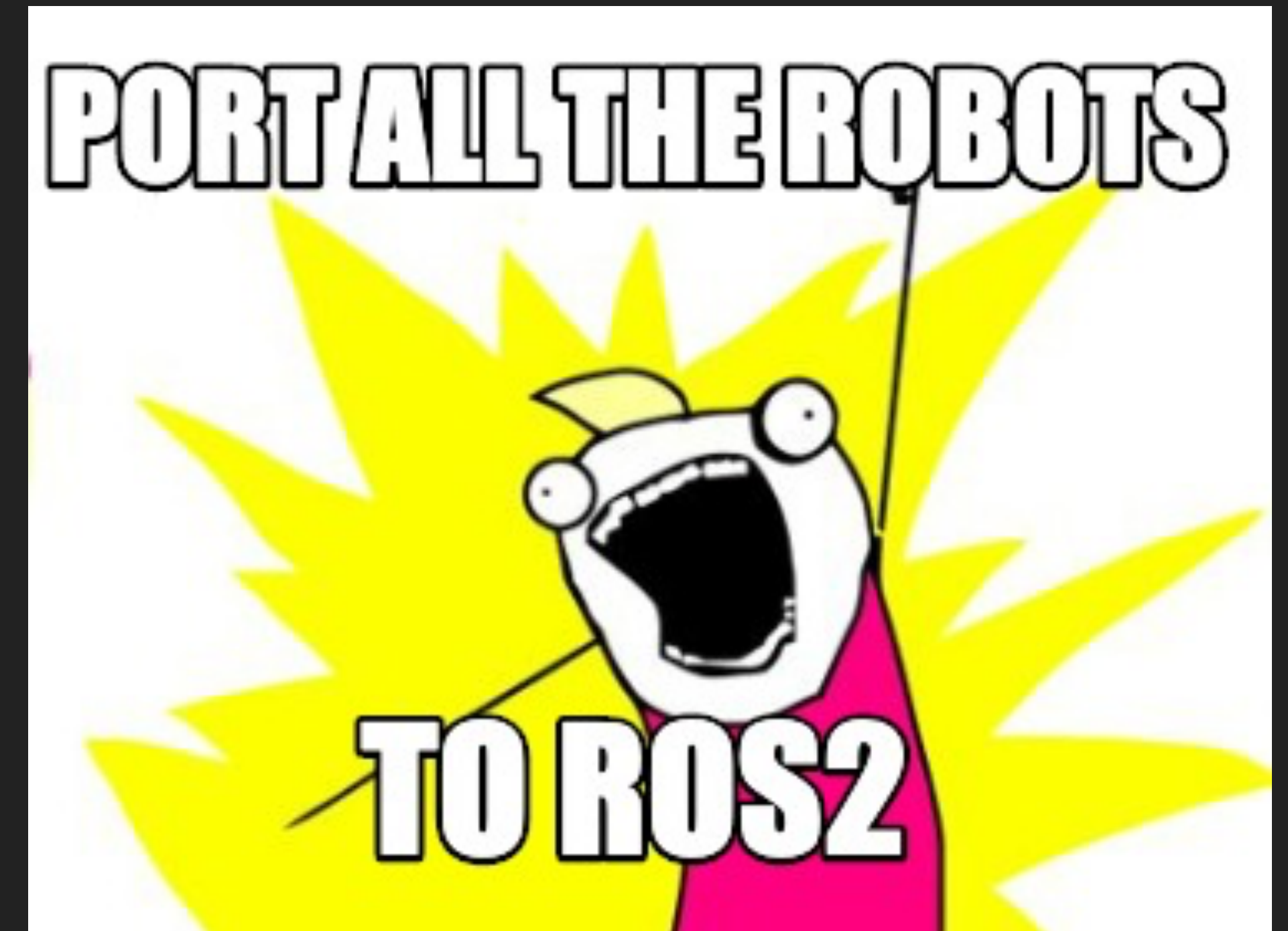
Star 816

- Client Libraries
 - rclcpp [API](#)
 - [Logging](#)
 - [Nodes and Components](#)
 - [Parameters](#)
 - [Point Clouds](#)
 - [Time](#)
 - [TF2](#)
 - [Workarounds](#)
 - rclpy [API](#)
 - [Nodes](#)
 - [Parameters](#)
 - [Time](#)
 - [TF2](#)
- [ros2launch](#)
 - [Python-Based Launch Files](#)
 - [Making a Launch File Executable](#)
 - [Loading Parameters From a File](#)



FUTURE WORK

- ▶ UBR-1 MoveIt2 Blog Post: Posted Now!
- ▶ UBR-1 on Jazzy: In Progress
- ▶ New Gazebo Simulation
- ▶ Actual mobile AND manipulation AT THE SAME TIME





RESOURCES



<http://www.robotandchisel.com/roscon24>

Michael Ferguson