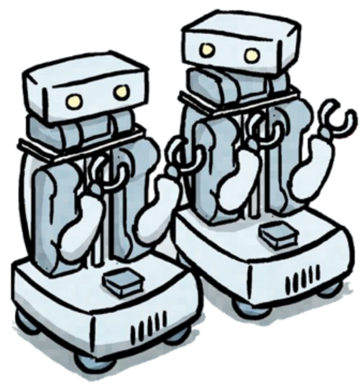


Motion Planning with ROS and OMPL

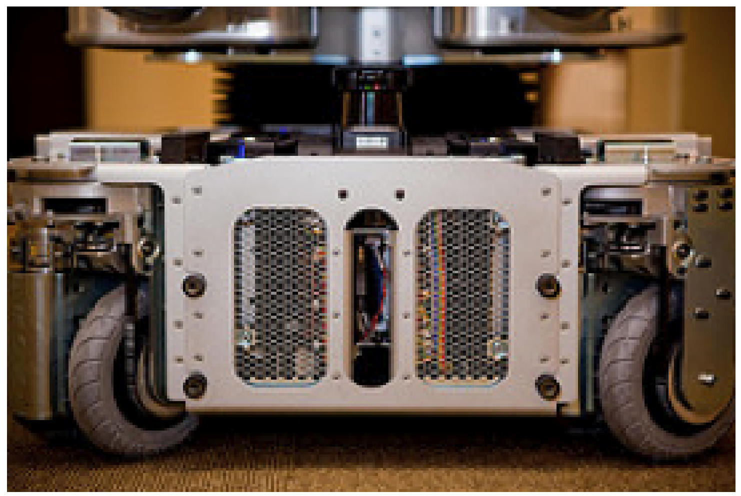
Sachin Chitta

Outline

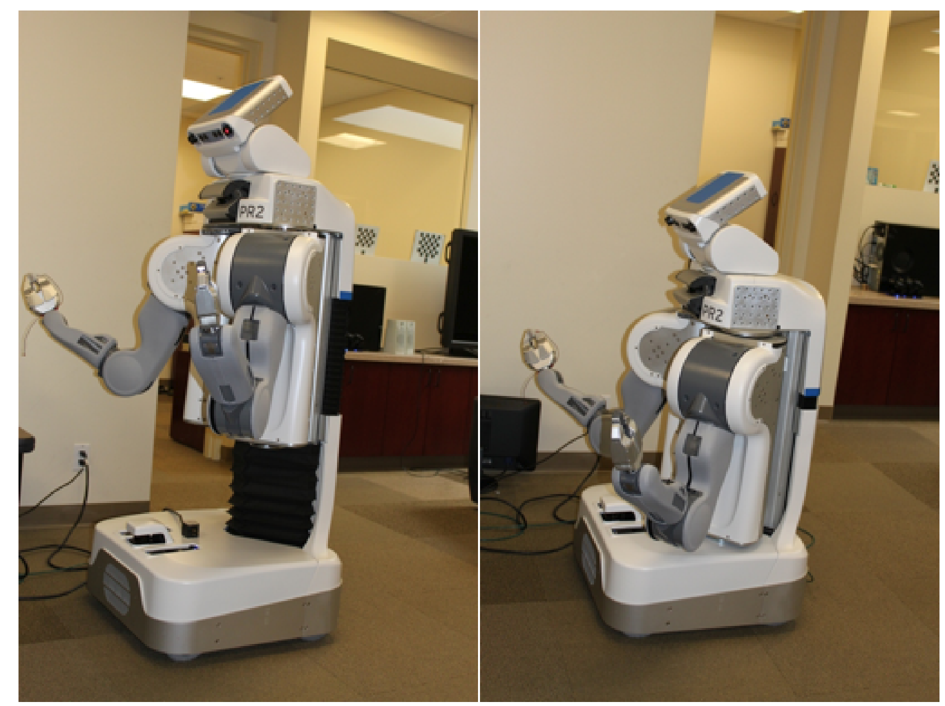
- PR2
- ROS
  - Simulation
  - Motion Planning
    - Sensing for motion planning
    - Collision Environment
    - ROS Interface to OMPL
- Applications

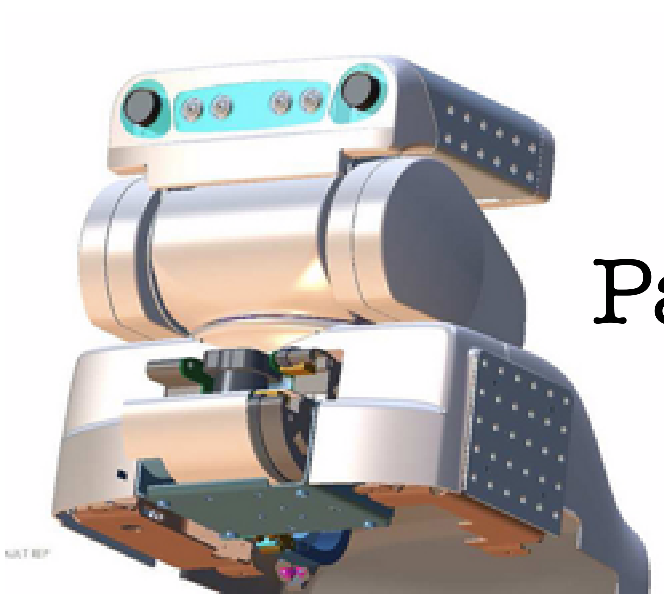
PR2

Omnidirectional base



Telescoping spine

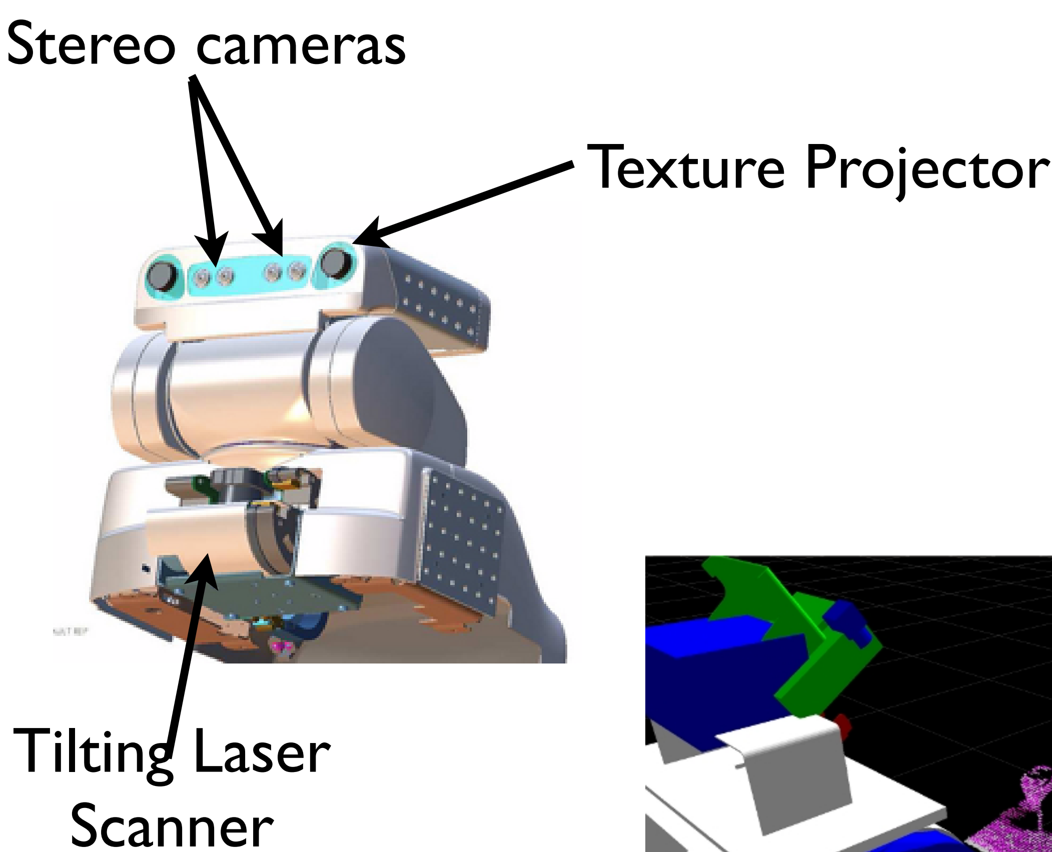


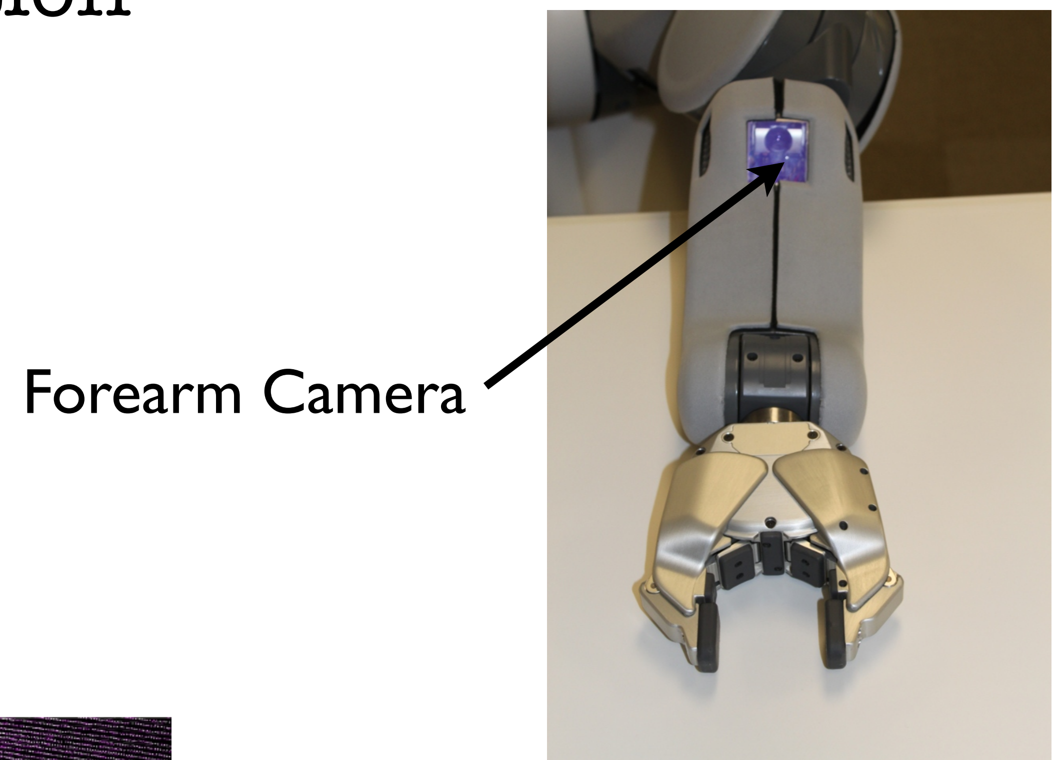


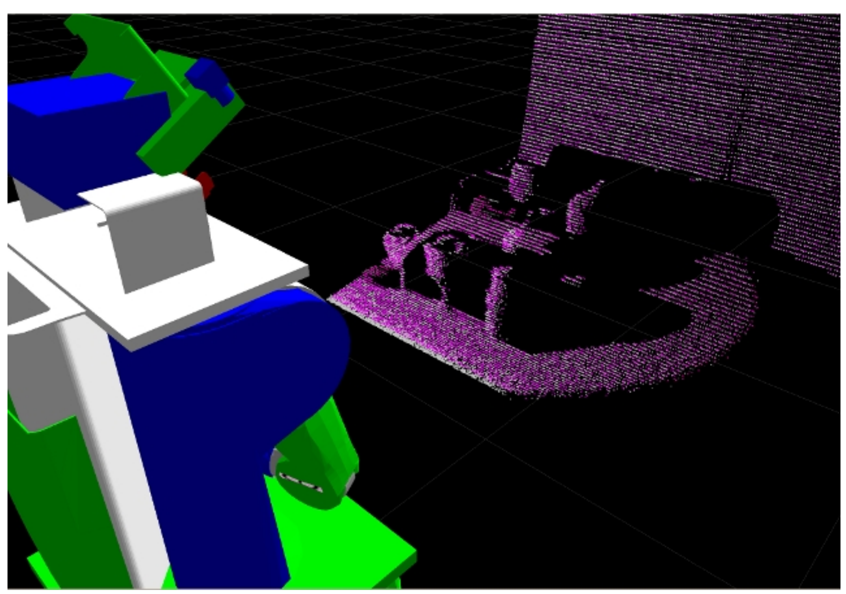
Pan and Tilt head

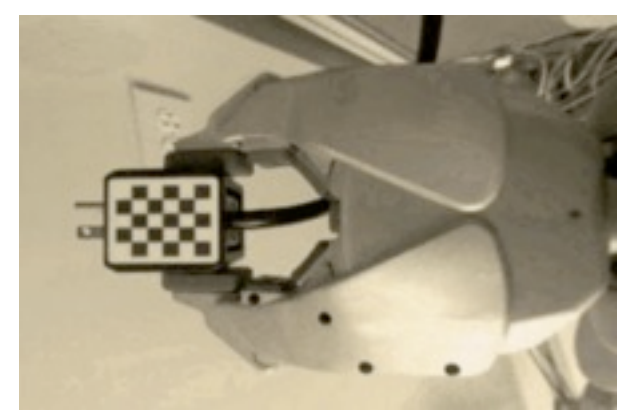
Sensing on the PR2

- Sensors for manipulation









Computing power

- 2 servers
  - 8 cores each
  - 20 GB Memory
  - External hard drives for data storage

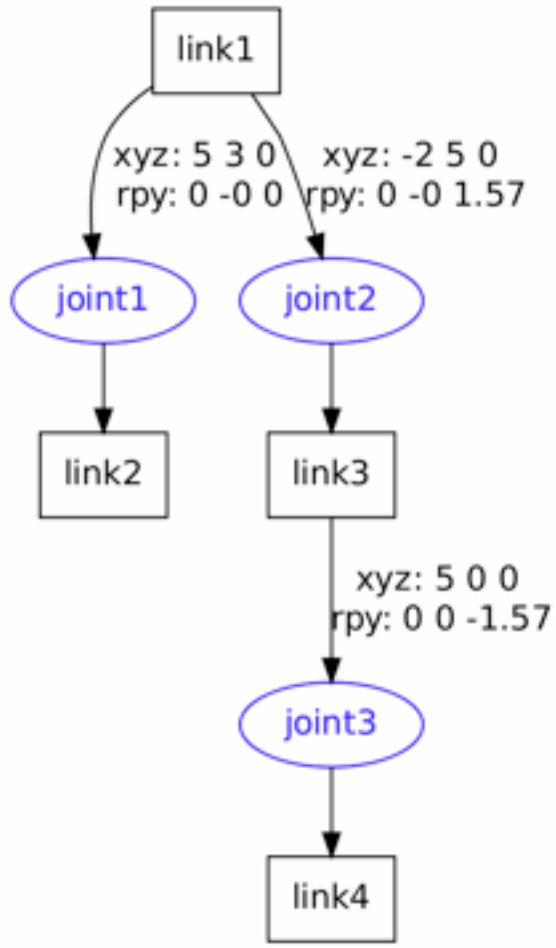
Outline

- PR2
- ROS
  - Simulation
  - Motion Planning
    - Sensing for motion planning
    - Collision Environment
    - ROS Interface to OMPL
- Applications

Robot Description

- URDF (Universal Robot Description Format)
  - ✓ serial manipulators
  - ✓ URDFs already available for different robots

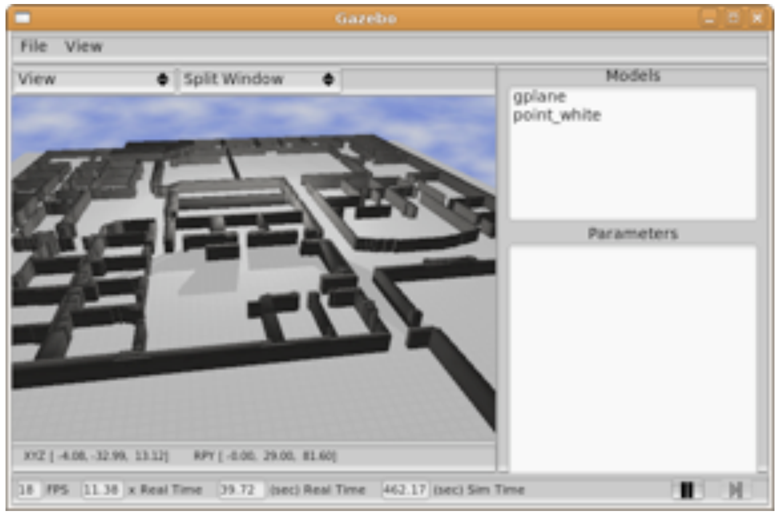
```
<robot name="test_robot">
  <link name="link1" />
  <link name="link2" />
  <link name="link3" />
  <link name="link4" />

  <joint name="joint1" type="continuous">
    <parent link="link1"/>
    <child link="link2"/>
    <origin xyz="5 3 0" rpy="0 0 0" />
    <axis xyz="-0.9 0.15 0" />
  </joint>

  <joint name="joint2" type="continuous">
    <parent link="link1"/>
    <child link="link3"/>
    <origin xyz="-2 5 0" rpy="0 0 1.57" />
    <axis xyz="-0.707 0.707 0" />
  </joint>

  <joint name="joint3" type="continuous">
    <parent link="link3"/>
    <child link="link4"/>
    <origin xyz="5 0 0" rpy="0 0 -1.57" />
    <axis xyz="0.707 -0.707 0" />
  </joint>
</robot>
```

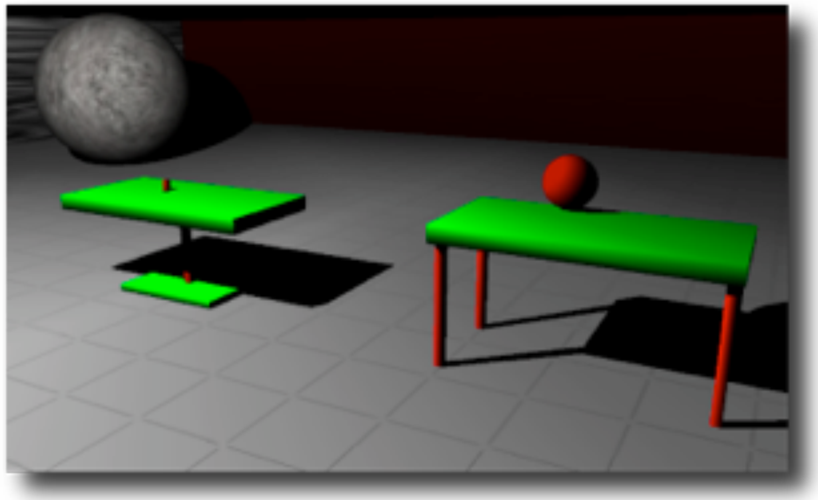


1. www.ros.org/wiki/urdf
2. www.ros.org/wiki/robots

Simulator

- Gazebo
  - ❖ now maintained actively by Willow Garage
  - ❖ using the URDF and ROS or through config files
    - ✓ add robots
    - ✓ add objects
    - ✓ add sensors
    - ✓ add maps

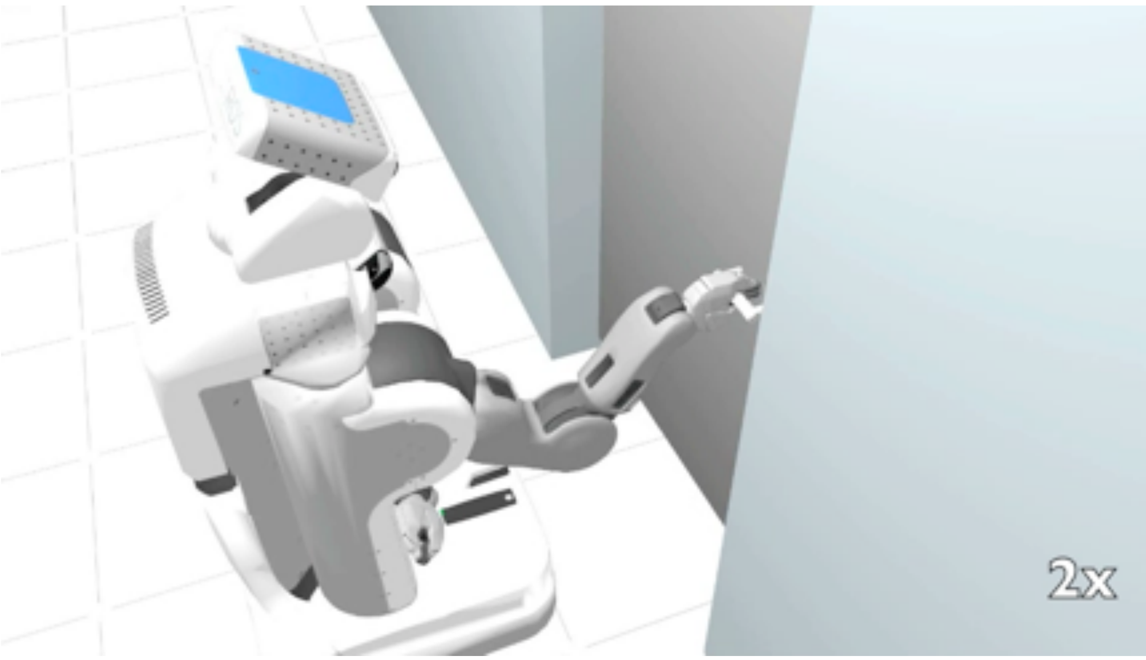


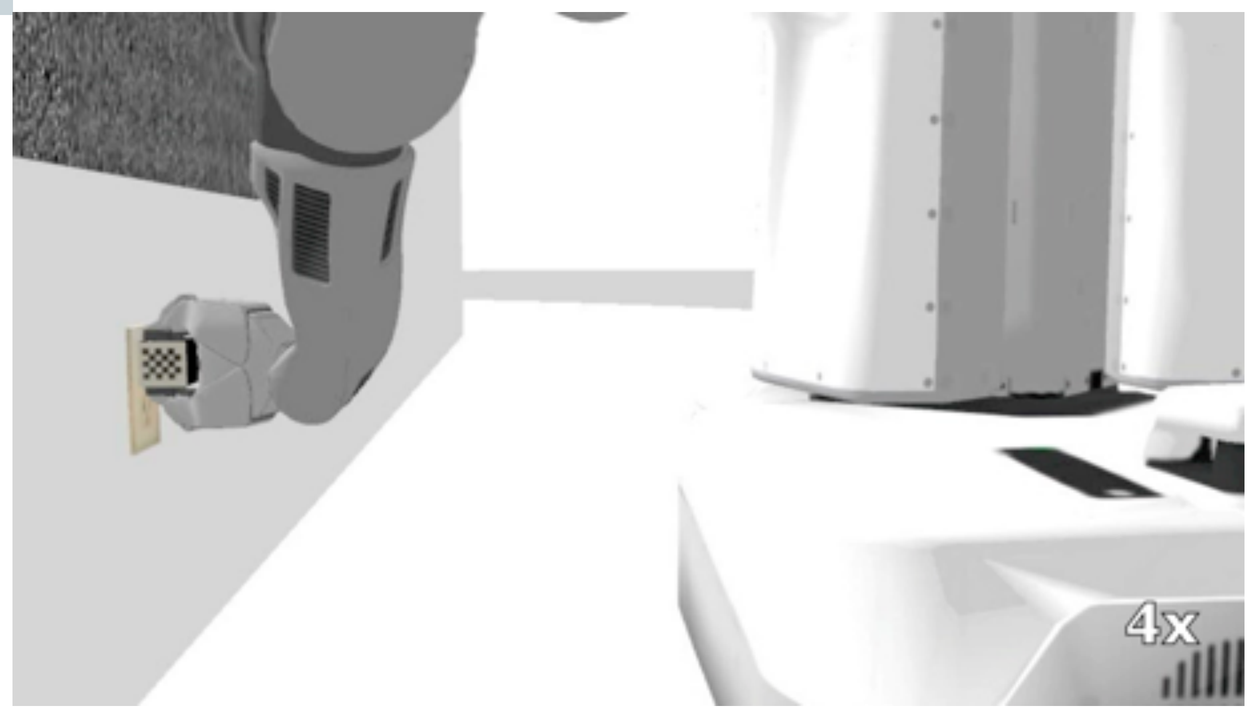


Willow Garage

http://www.ros.org/wiki/simulator_gazebo

Simulator





http://www.ros.org/wiki/pr2_gazebo

Outline

- PR2
- ROS
  - Simulation
  - Motion Planning
    - ❖ Sensing for motion planning
    - ❖ Collision Environment
    - ❖ ROS Interface to OMPL
- Applications

Motion Planning

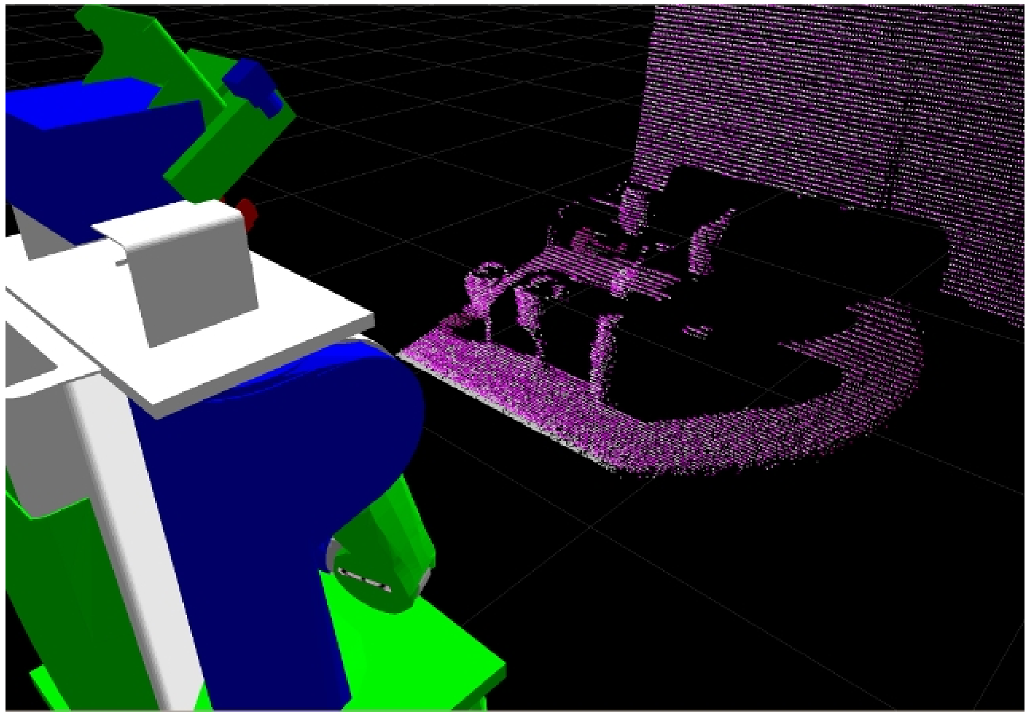
- Motion Planning
  - Sensing for motion planning
  - Collision Space
  - Kinematics
  - ROS Interface to OMPL

Motion Planning

- Motion Planning
  - Sensing for motion planning
  - Collision Space
  - Kinematics
  - ROS Interface to OMPL

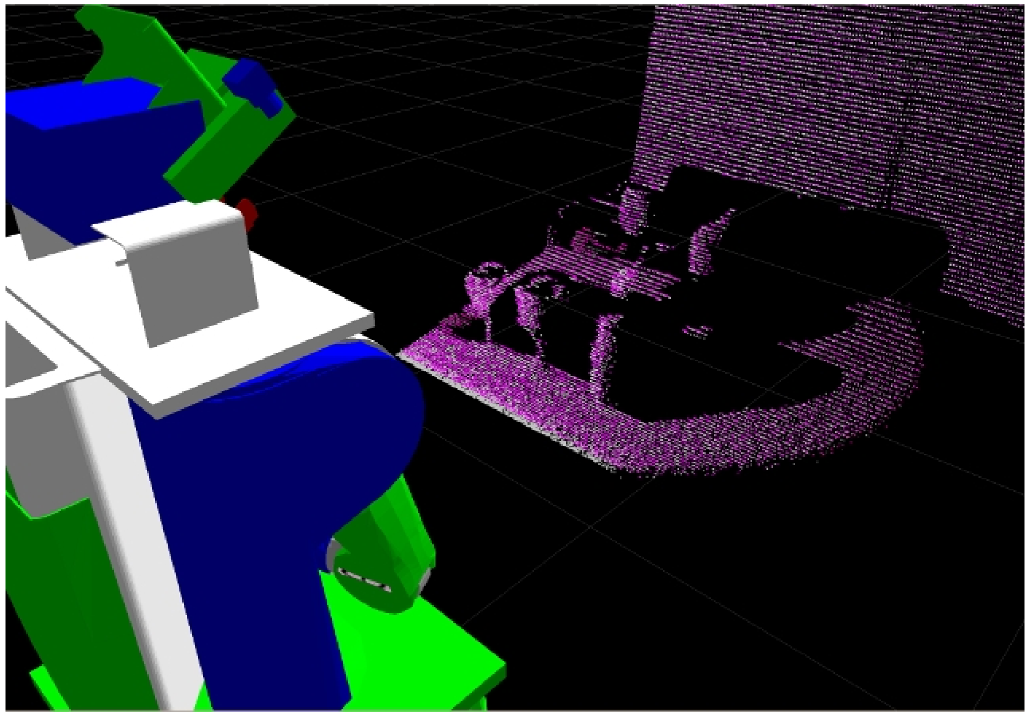
Sensing

- Essential for operation in unstructured environments
- Sensor input
  - point cloud (from stereo, lasers, etc.)
    - ❖ collection of 3D points
    - ❖ can annotate other information
      - ✓ RGB color, intensity values
    - ❖ stereo (20-25,000 points)
    - ❖ laser sensor (5-7,000 points)



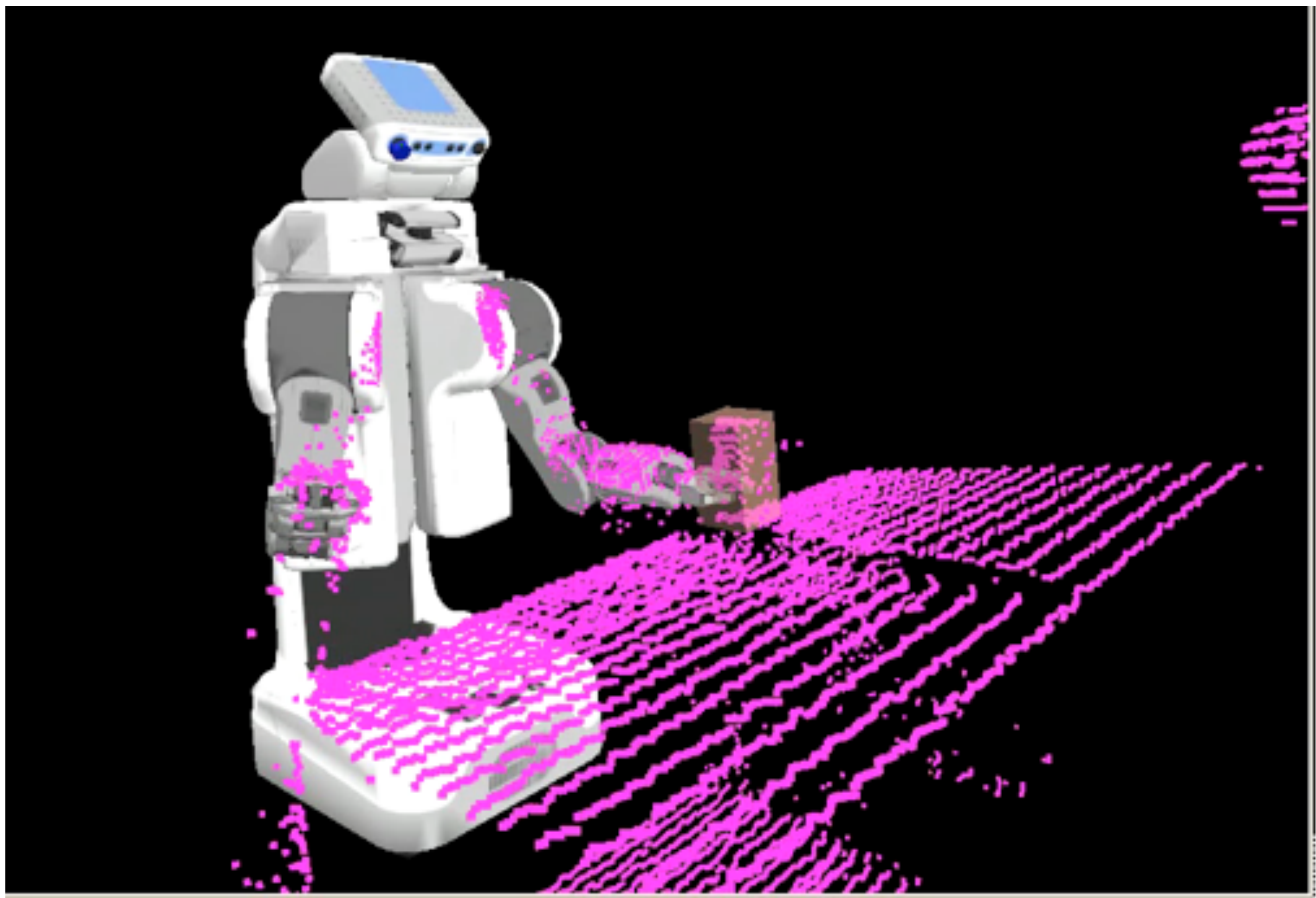
Sensing

- Essential for operation in unstructured environments
- Sensor input
  - point cloud (from stereo, lasers, etc.)
    - ❖ collection of 3D points
    - ❖ can annotate other information
      - ✓ RGB color, intensity values
    - ❖ stereo (20-25,000 points)
    - ❖ laser sensor (5-7,000 points)



Willow Garage

Sensing Pipeline



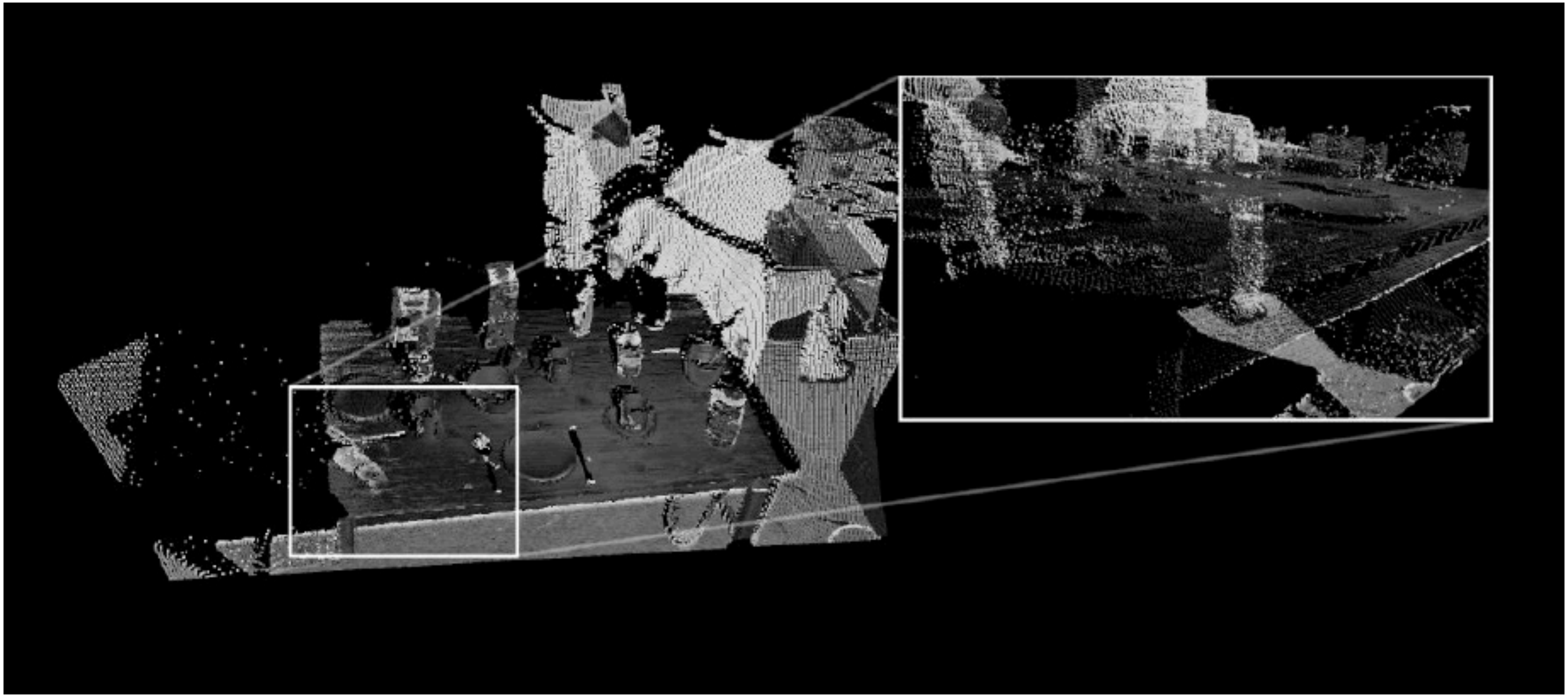
- Shadow filtering
  - filter out noisy sensor data
- Self-filtering
  - filter out body parts and attached objects
- Final filtered scans
  - less noisy + do not contain body parts

Environment Representation

- Two parts
  - Semantic perception
    - ❖ represent known objects
    - ❖ e.g. tables, objects recognized using object detectors
  - 3D grid representation
    - ❖ more raw representation
    - ❖ essential for complete representation
      - ✓ all parts of environment not part of semantic representation

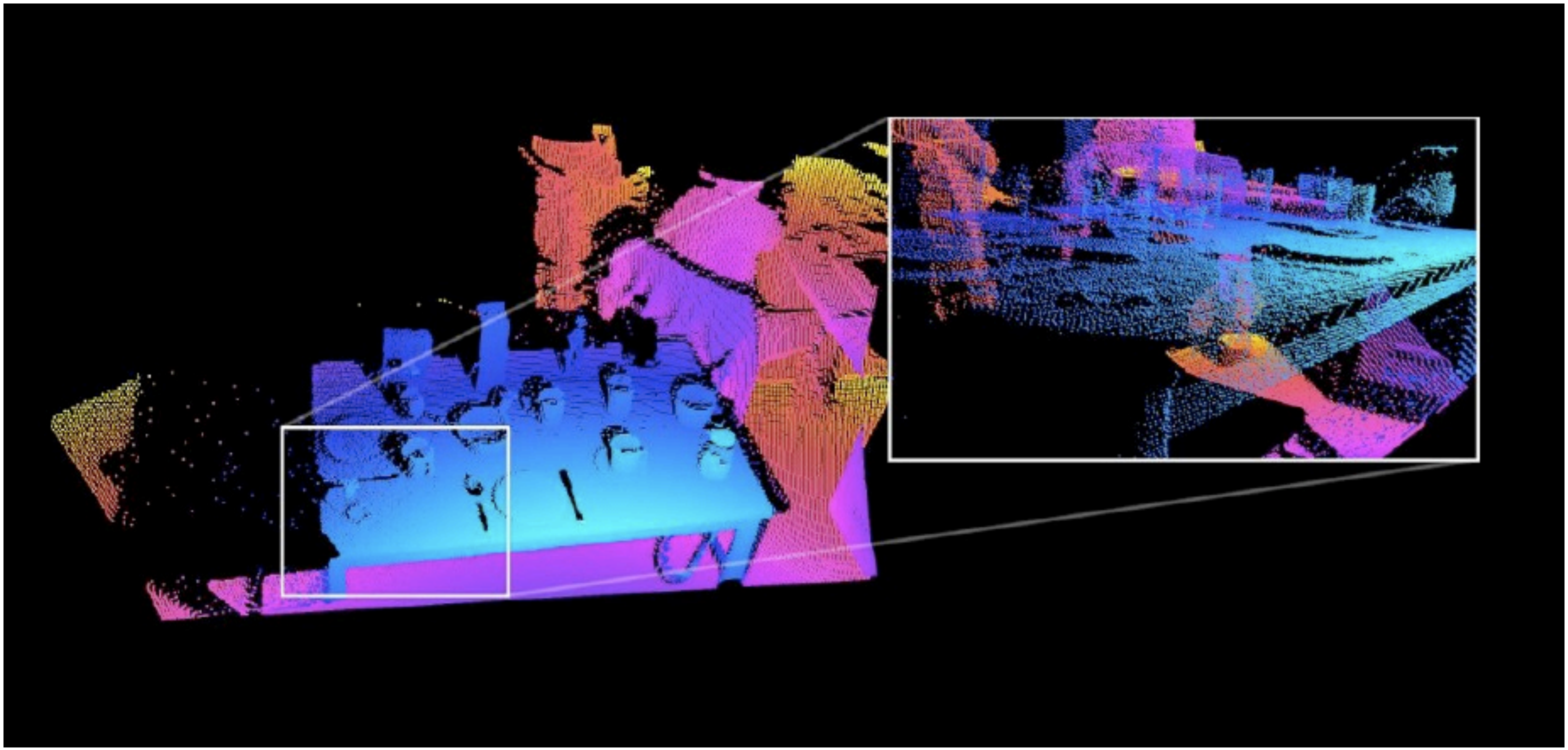
Semantic Perception

Extracting semantic information from point clouds



Semantic Perception

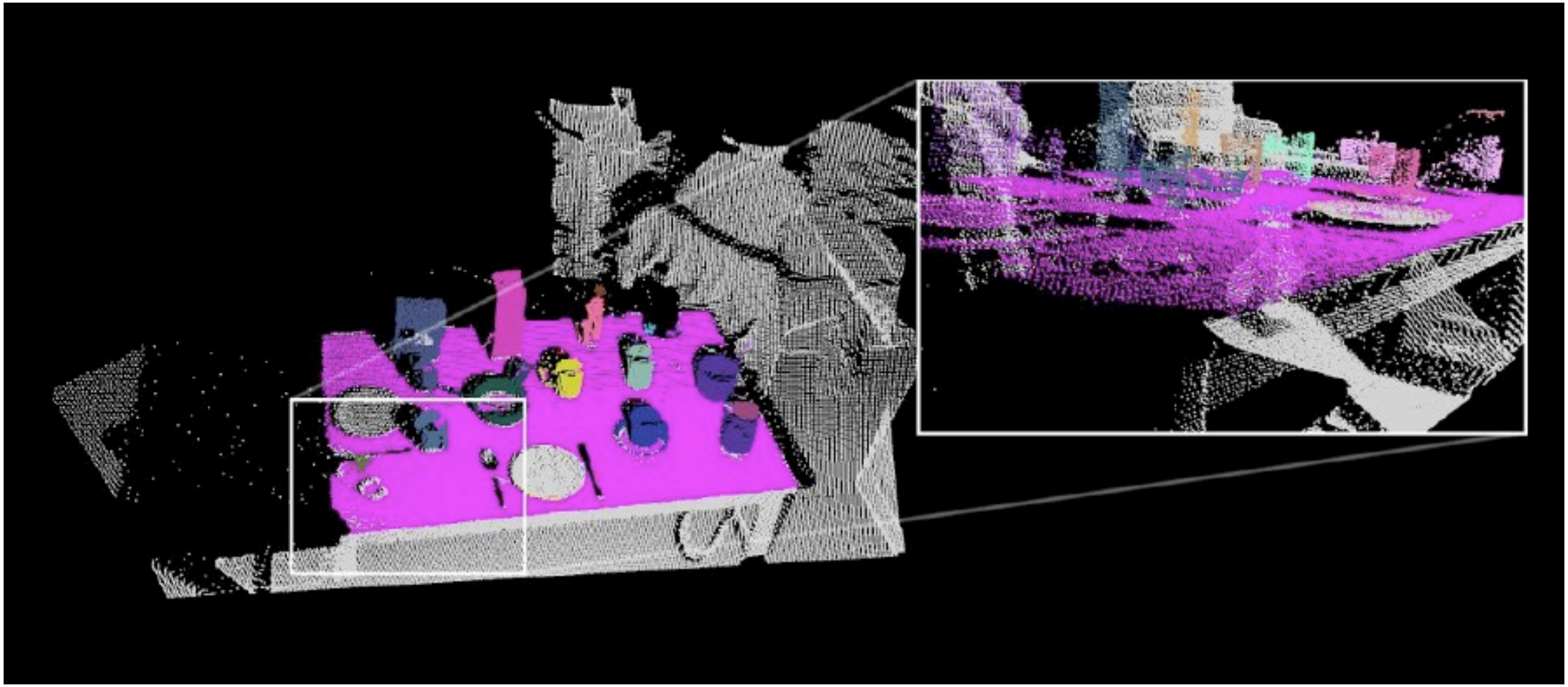
Extracting semantic information from point clouds



Willow Garage

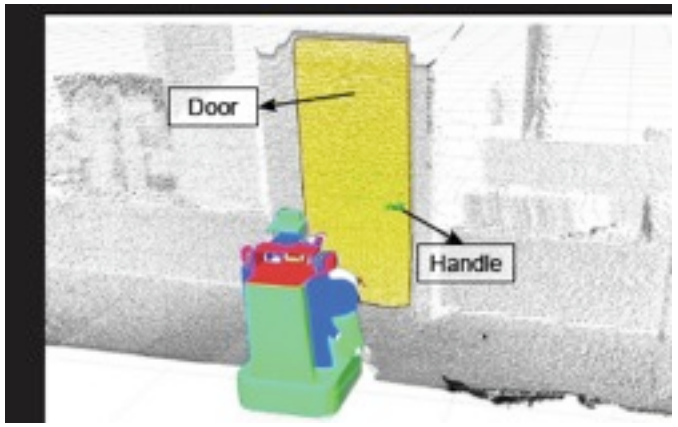
Semantic Perception

Extracting semantic information from point clouds

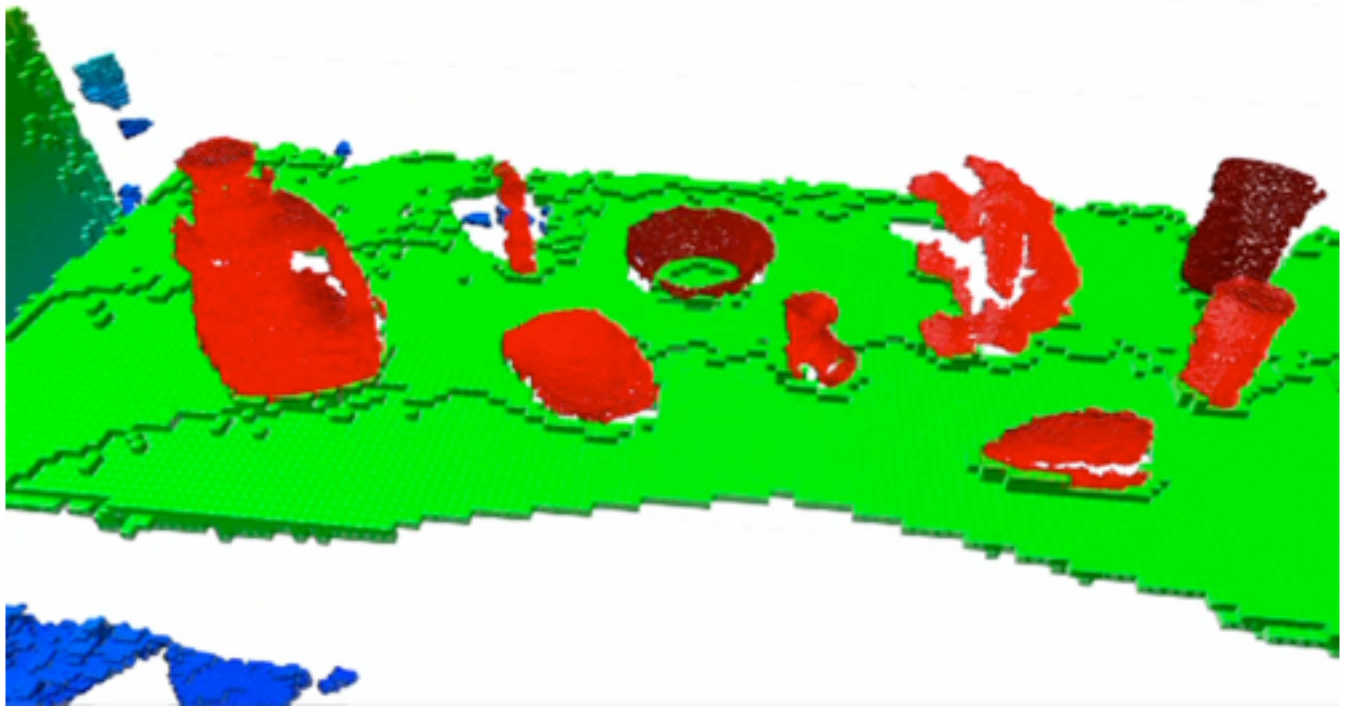


Semantic Perception

- Helps in
  - object recognition
  - object representation for collision checking
  - place recognition
  - task planning
    - ✓ place/pickup locations for objects
    - ✓ constrained planning



Octomap



- Octree-based representation of the environment
  - Probabilistic, flexible, and compact 3D mapping
  - Optimized for online operation

Octomap - http://octomap.sf.net
Armin Hornung, Kai Wurm, et. al., Freiburg

Motion Planning

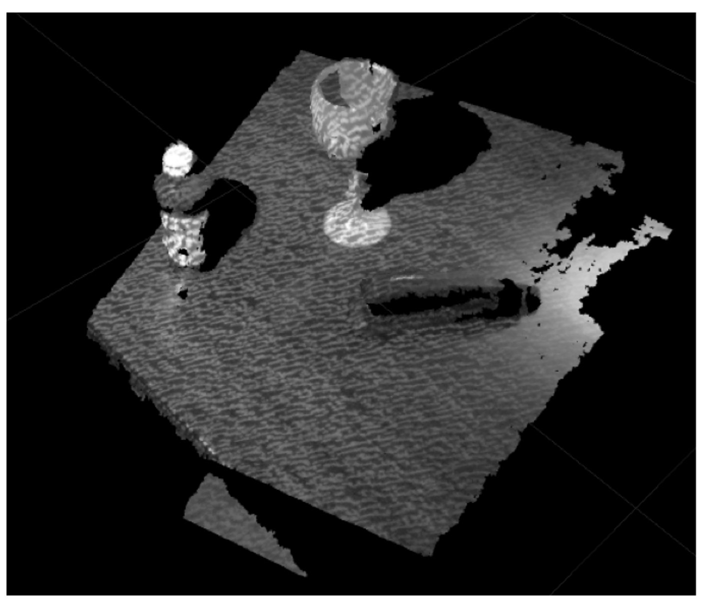
- Motion Planning
  - Sensing for motion planning
  - Collision Space
  - Kinematics
  - ROS Interface to OMPL

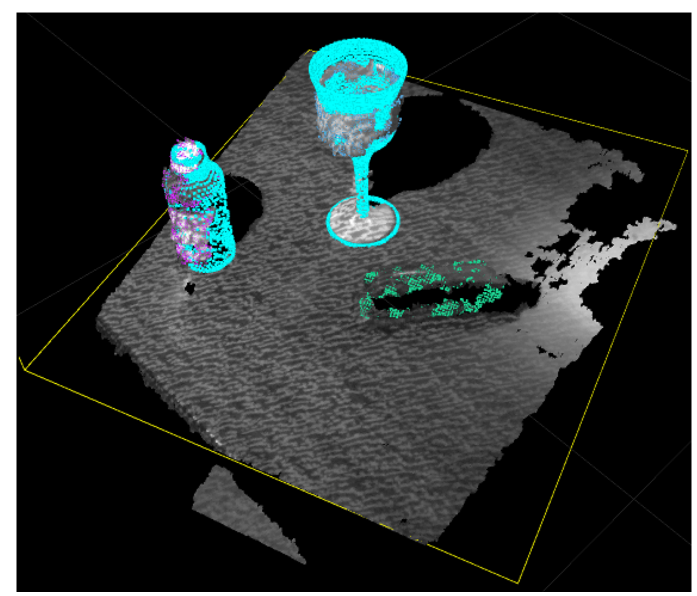
Motion Planning

- Motion Planning
  - Sensing for motion planning
  - Collision Space
  - Kinematics
  - ROS Interface to OMPL

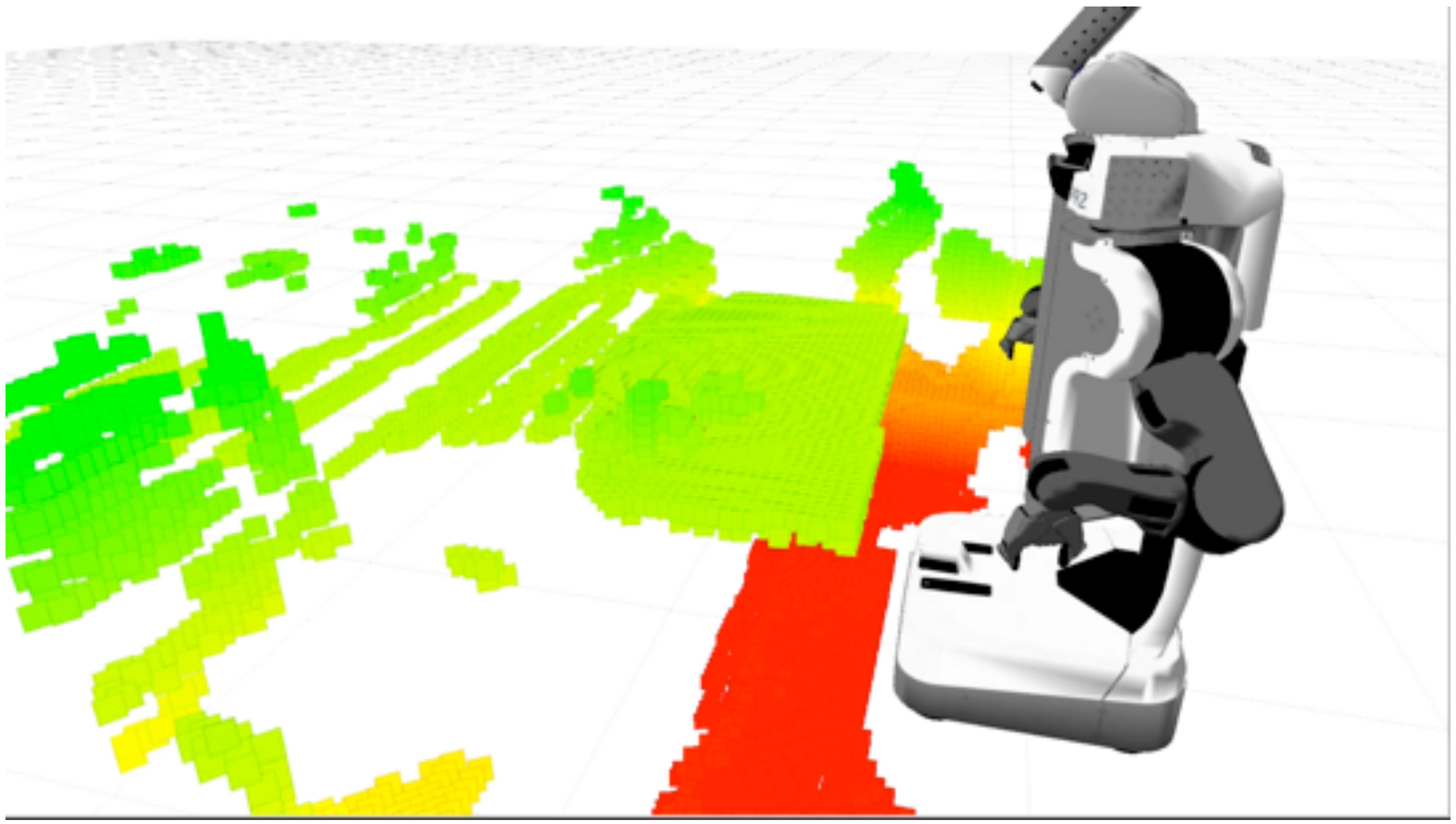
Collision Space

- From point clouds to collision representation
  - meshes for known objects
  - unmodeled parts of environment represented by collision map





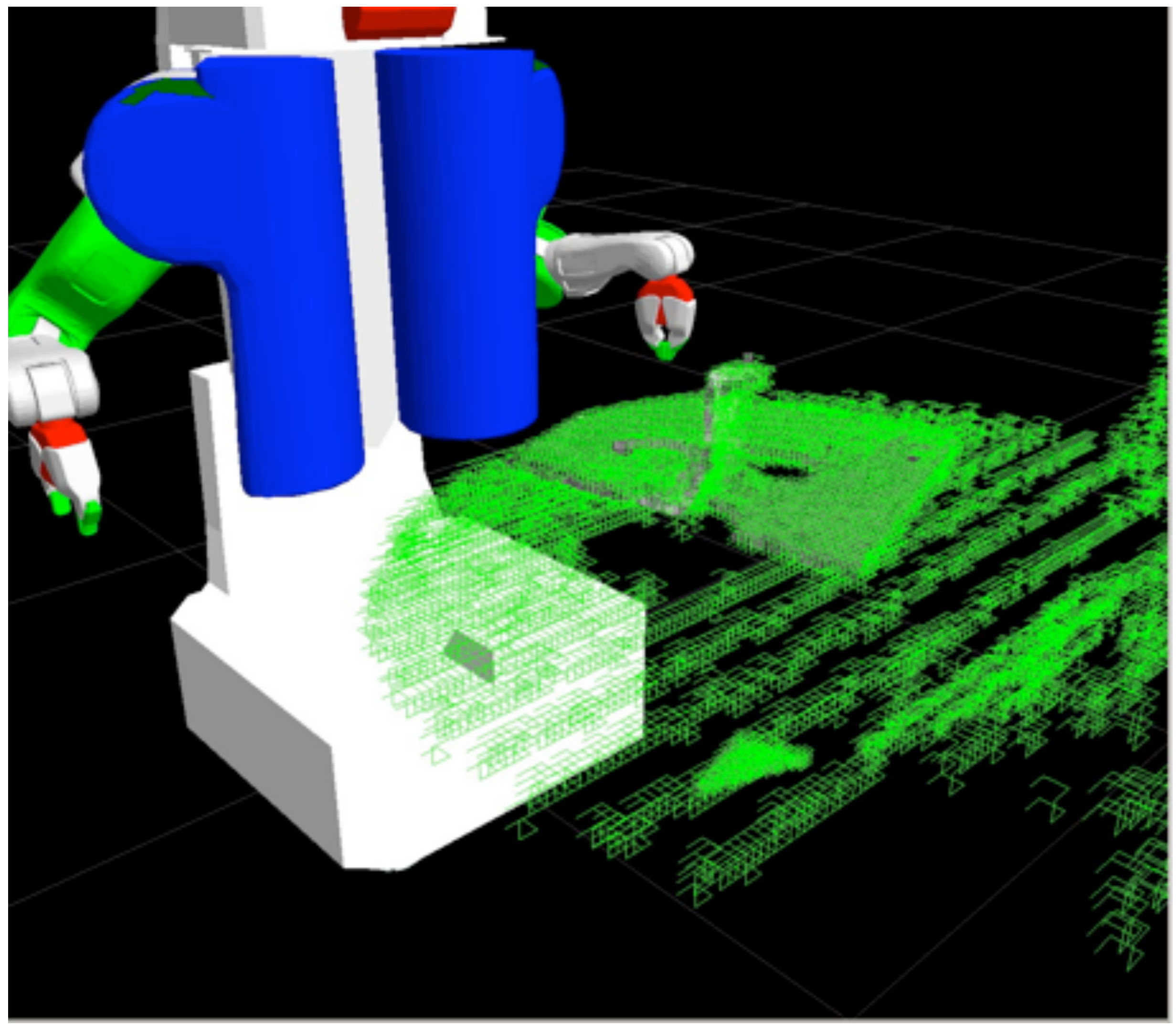
Collider



- Convert octree-based representation to collision map
  - axis aligned boxes
  - extensively used for dealing with unmodeled parts of environment
  - future plans to use notion of uncertainty in sensing
  - probabilistic representation possibly more robust to sensor noise, errors

Collider -http://www.ros.org/wiki/collider

Collision Space



Collision map -http://www.ros.org/wiki/collision_map
Collision space - http://www.ros.org/wiki/collision_space

Motion Planning

- Motion Planning
  - Sensing for motion planning
  - Collision Space
  - Kinematics
  - ROS Interface to OMPL

Motion Planning

- Motion Planning
  - Sensing for motion planning
  - Collision Space
  - Kinematics
  - ROS Interface to OMPL

Kinematics

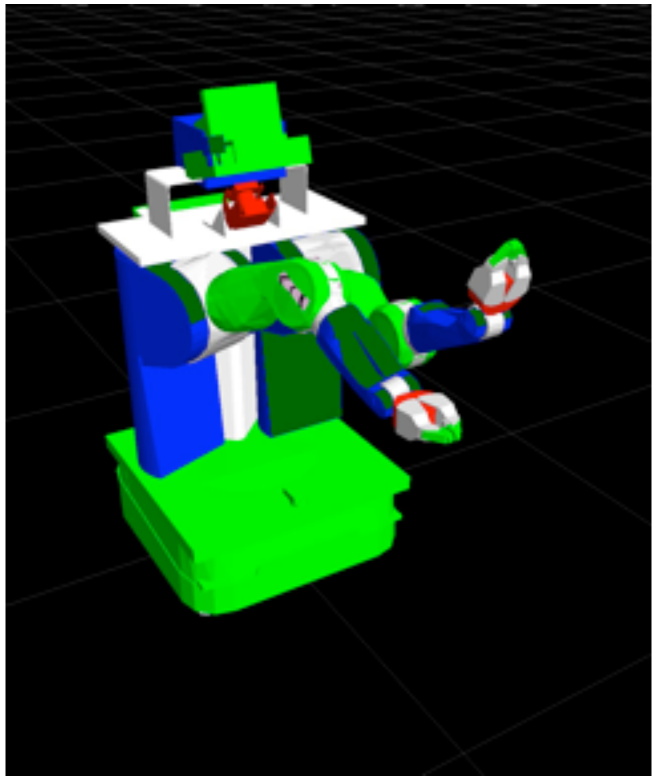
- Robot specific kinematics
  - e.g. pr2_kinematics package - fast custom solvers

- Constraint, collision aware kinematics
  - search for collision free configurations that satisfy constraints for general arms

PR2 custom kinematics -http://www.ros.org/wiki/pr2_arm_kinematics
General kinematics solvers - http://www.ros.org/wiki/arm_kinematics_constraint_aware

Kinematics

- Robot specific kinematics
  - e.g. pr2_kinematics package - fast custom solvers



- Constraint, collision aware kinematics
  - search for collision free configurations that satisfy constraints for general arms

PR2 custom kinematics -http://www.ros.org/wiki/pr2_arm_kinematics
General kinematics solvers - http://www.ros.org/wiki/arm_kinematics_constraint_aware

Kinematics

- Kinematics plugin interface
  - YAML configuration
  - you can implement your own kinematics implementation and configure it using YAML
    - ✓ Fast custom kinematics for the PR2


```
right_arm:
  tip_name: r_wrist_roll_link
  root_name: torso_lift_link
  kinematics_solver: pr2_arm_kinematics/PR2ArmKinematicsPlugin
```


    - ✓ KDL based generic kinematics for any arm


```
right_arm:
  tip_name: r_wrist_roll_link
  root_name: torso_lift_link
  kinematics_solver: arm_kinematics_constraint_aware/KDLArmKinematicsPlugin
```

Motion Planning

- Motion Planning
  - Sensing for motion planning
  - Collision Space
  - Kinematics
  - ROS Interface to OMPL

Motion Planning

- Motion Planning
  - Sensing for motion planning
  - Collision Space
  - Kinematics
  - ROS Interface to OMPL

ROS - Motion Planners

- Sampling based planners (OMPL)
  - ❖ interfaced to ROS through the ompl_ros_interface package

http://www.ros.org/wiki/ompl
http://www.ros.org/wiki/ompl_ros_interface

Willow Garage

OMPL

- Sampling-based planners
  - RRT, RRT-Connect, EST, KPIECE, SBL and more
  - new planner - RRT*
- Very fast planning times
- Extensive documentation

OMPL - http://www.ros.org/wiki/ompl
ROS interface to OMPL - http://www.ros.org/wiki/ompl_ros_interface

ROS Interface to OMPL

- YAML based configuration
  - setup planners for each planning group
  - easily configure all planners available in OMPL
  - modify parameters for OMPL planners
    - e.g. range, projection evaluator

OMPL - http://www.ros.org/wiki/ompl
ROS interface to OMPL - http://www.ros.org/wiki/ompl_ros_interface

ROS Interface to OMPL

- Input - ROS Motion Planning Request (MPR)
  - ❖ specify goal as regions in space
    - ✓ joint space goal - nominal joint angle + tolerance
    - ✓ position goal - nominal position + region of space
    - ✓ orientation goal - nominal orientation + tolerance (in roll, pitch and yaw)

Move arm -http://www.ros.org/wiki/move_arm
Interface definition -http://www.ros.org/wiki/move_arm_msgs

ROS Interface to OMPL

- Input - ROS Motion Planning Request (MPR)
  - ❖ Joint goals
  - ❖ Pose goals
    - ✓ spawns a separate thread to sample IK solutions
      - uses the kinematics plugin interface so you can use your own kinematics implementations as well

Move arm -http://www.ros.org/wiki/move_arm
Interface definition -http://www.ros.org/wiki/move_arm_msgs

ROS Interface to OMPL

- Input - ROS Constraint Specification
  - keep the glass of water level - less freedom in roll and pitch space
  - constraint specification:

```
orientation_constraint.header.frame_id = "torso_lift_link";
orientation_constraint.header.stamp = ros::Time::now();
orientation_constraint.link_name = "r_wrist_roll_link";
orientation_constraint.orientation.x = 0.0;
orientation_constraint.orientation.y = 0.0;
orientation_constraint.orientation.z = 0.0;
orientation_constraint.orientation.w = 1.0;
orientation_constraint.type = motion_planning_msgs::OrientationConstraint::HEADER_FRAME;
orientation_constraint.absolute_roll_tolerance = 0.1;
orientation_constraint.absolute_pitch_tolerance = 0.1;
orientation_constraint.absolute_yaw_tolerance = M_PI;
```

ROS Interface to OMPL

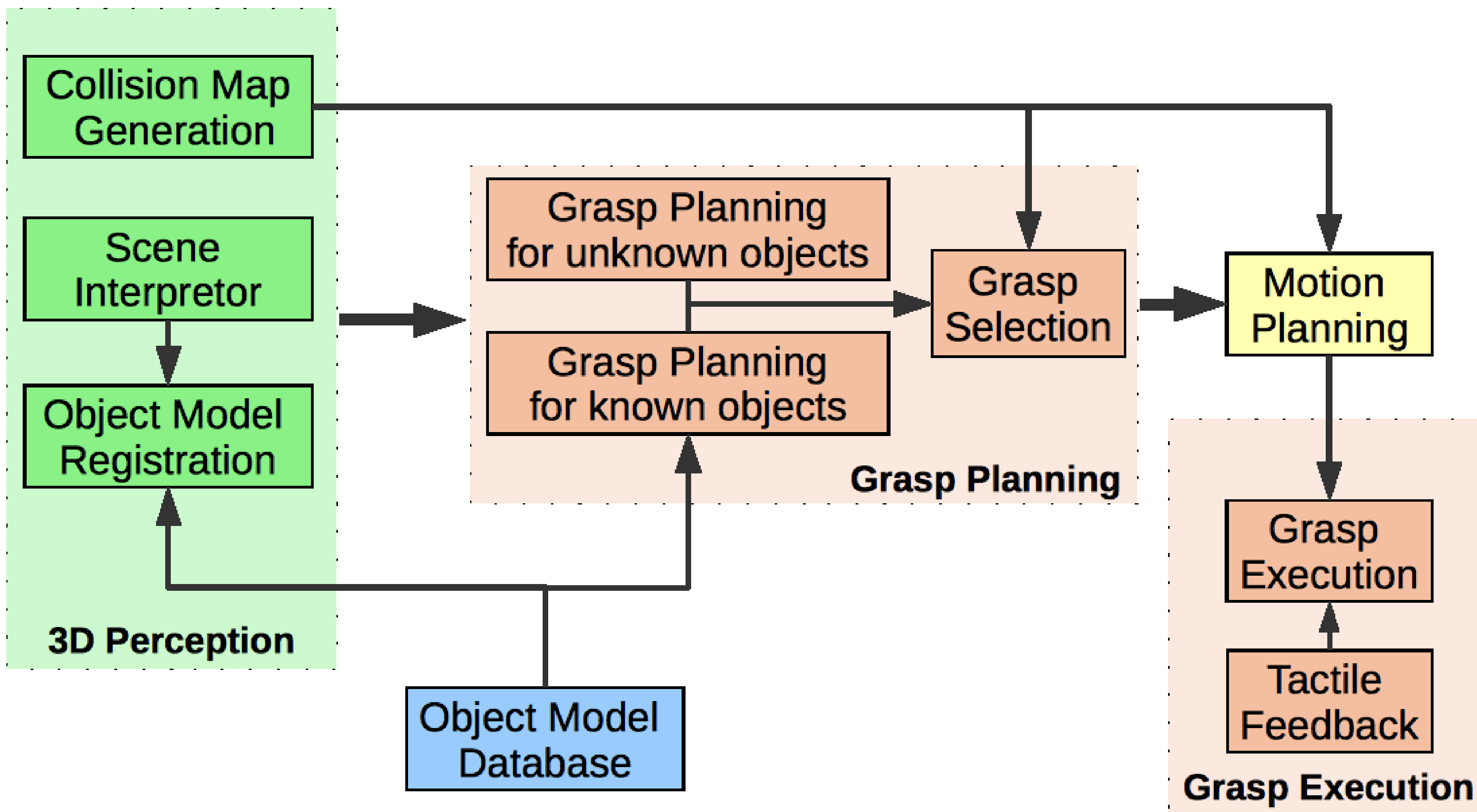
- Full integration of collision_space
  - uses the ompl state_validity_checker interface
    - specify and check constraints
    - fully integrated collision checking
  - plans for the future*
    - integrate proximity information
    - integrate with continuous collision checking

Smoothing

- Sampling based planners can generate jerky paths
  - ❖ smoothing required before plans can be sent to controllers
  - ❖ cubic spline shortcuts used to smoothen paths
  - ❖ also impose maximum velocity and acceleration constraints

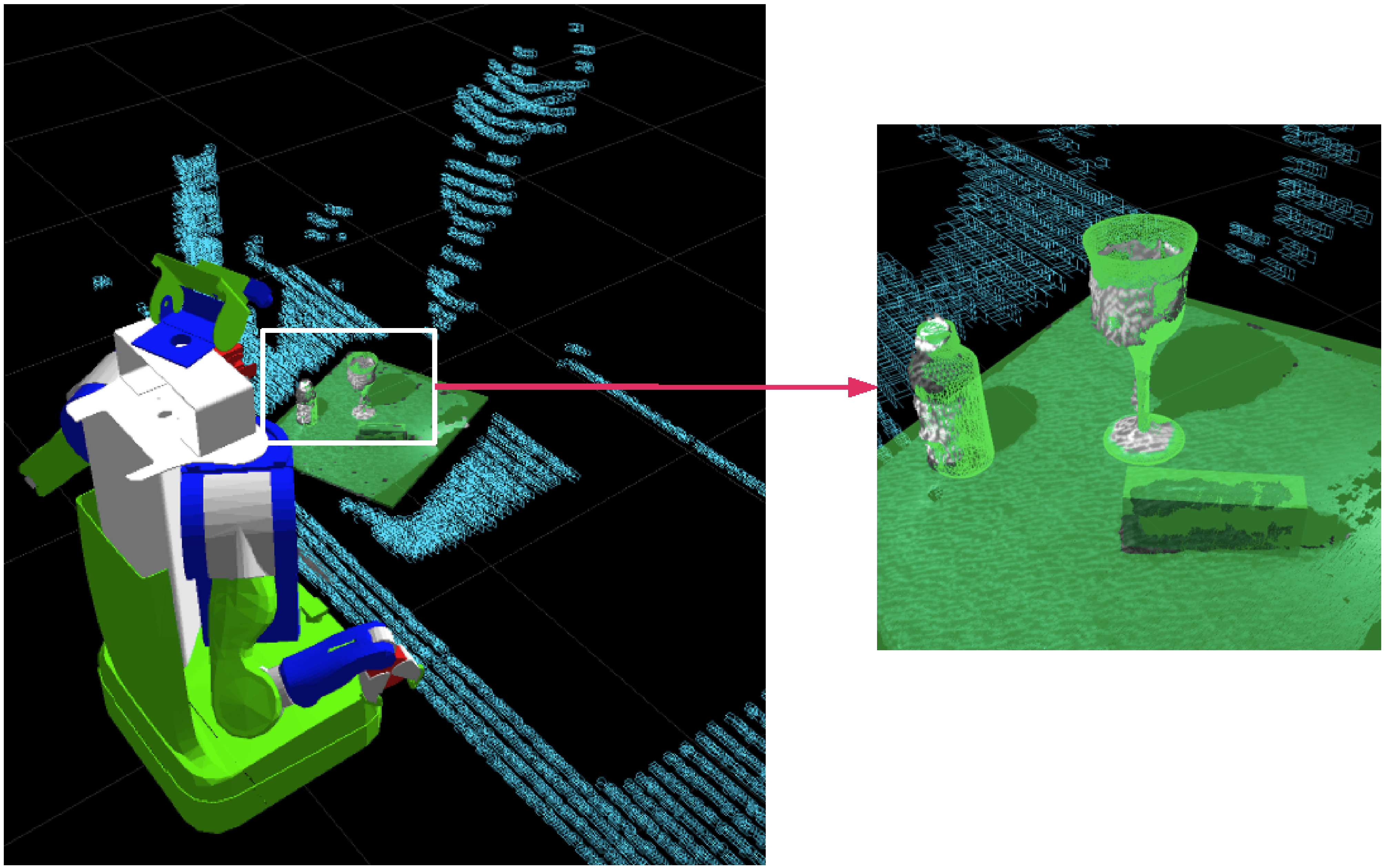
Cubic spline smoother - http://www.ros.org/wiki/constraint_aware_spline_smoother

Application - Grasping

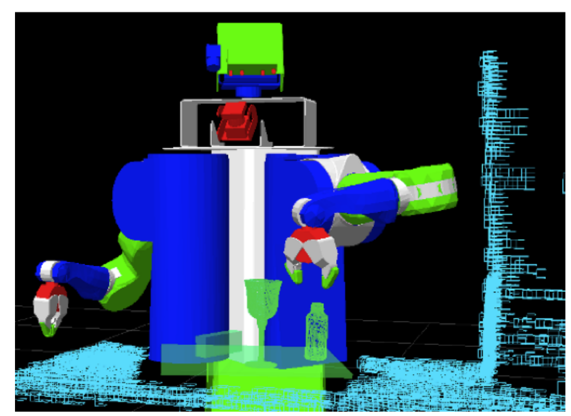


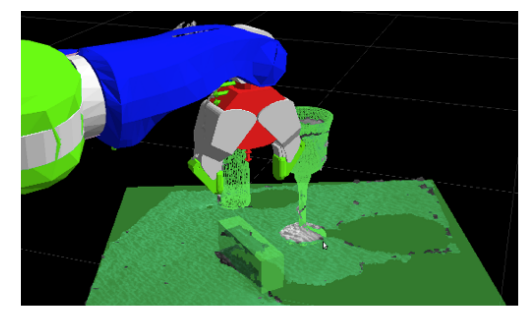
Willow Garage

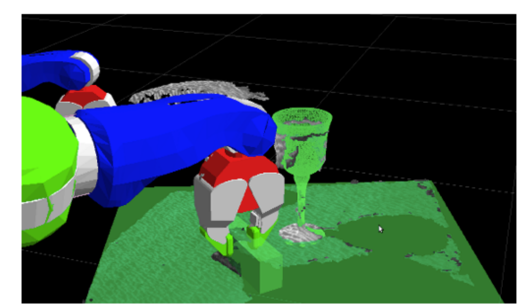
Grasping Pipeline



Arm Navigation and Manipulation

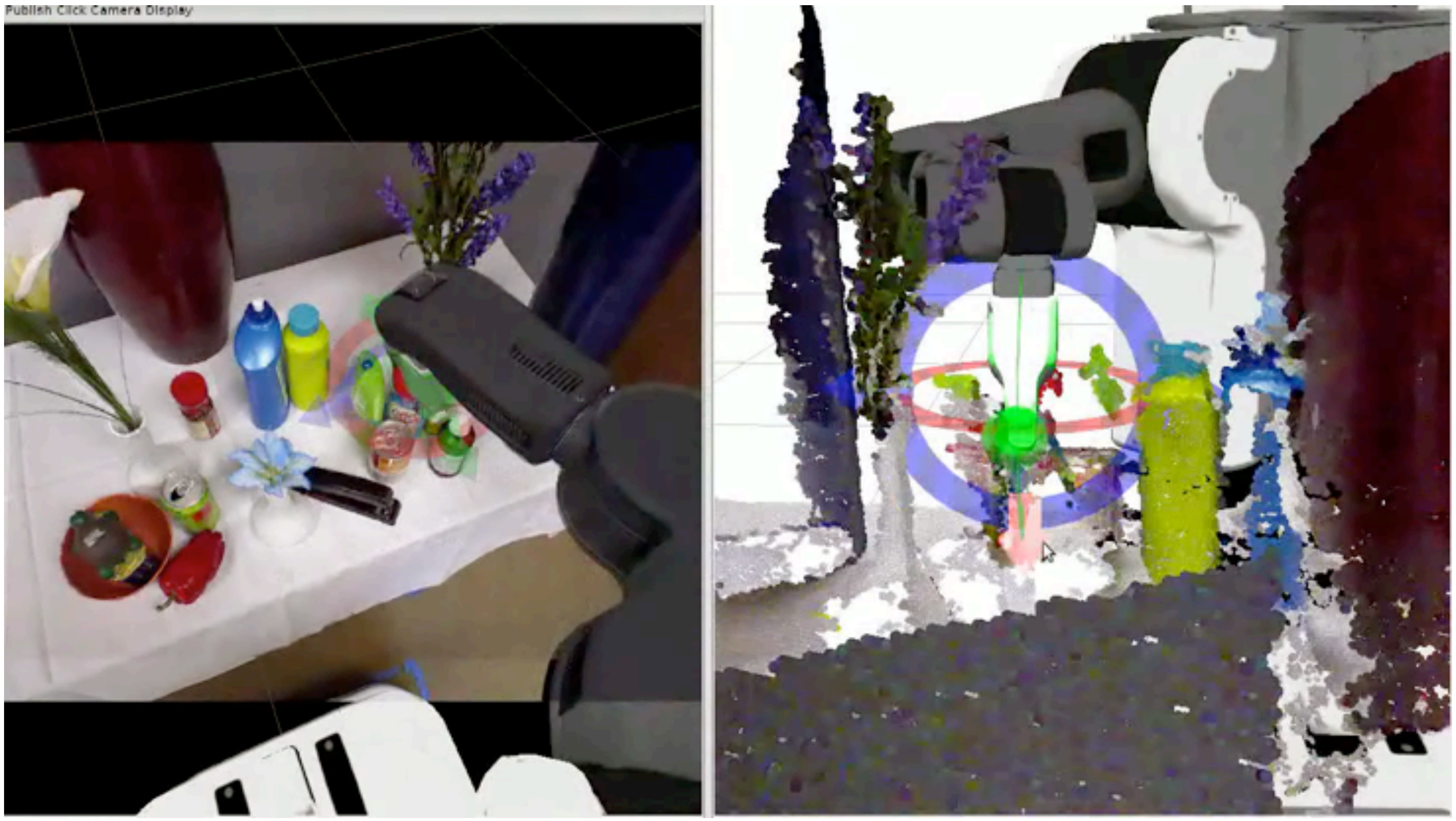




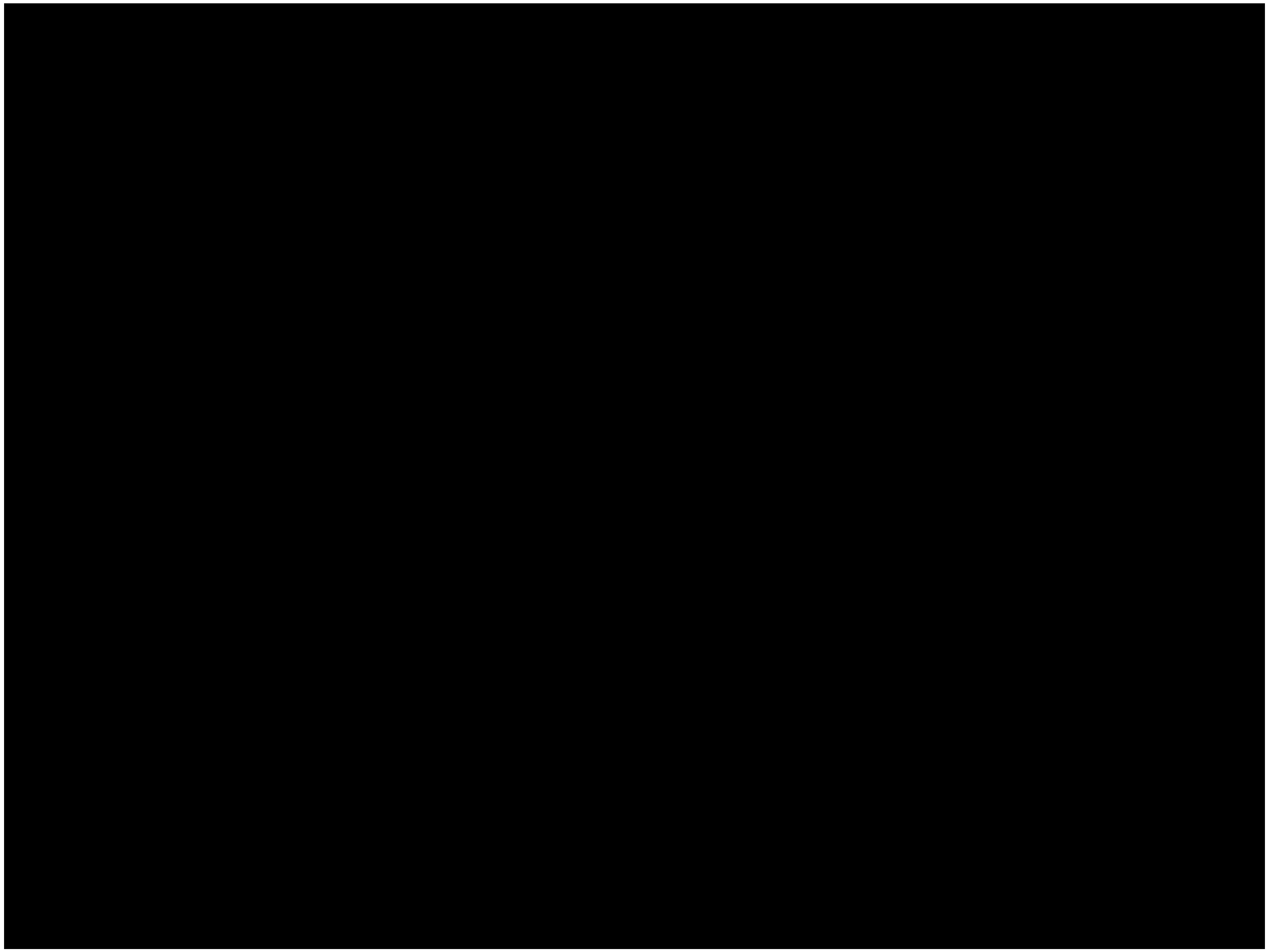


http://www.ros.org/wiki/arm_navigation
http://www.ros.org/wiki/object_manipulation
http://www.ros.org/wiki/pr2_object_manipulation

Applications

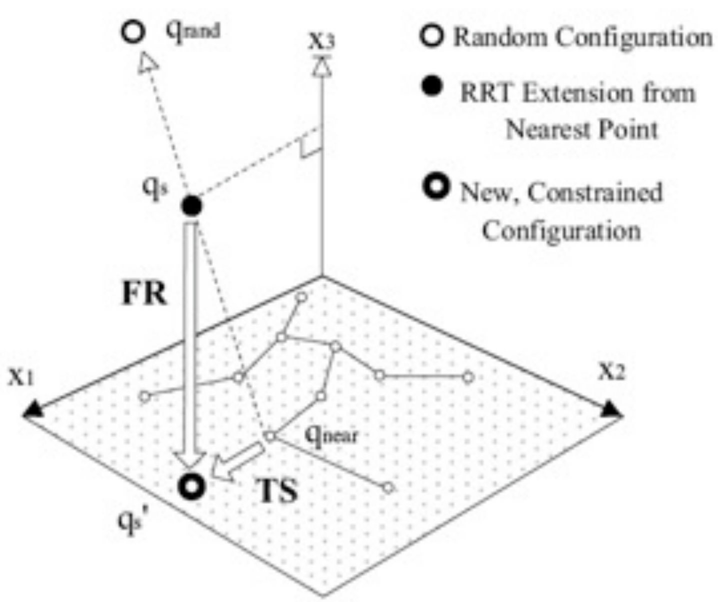


Application - Busbot



Dealing with constraints

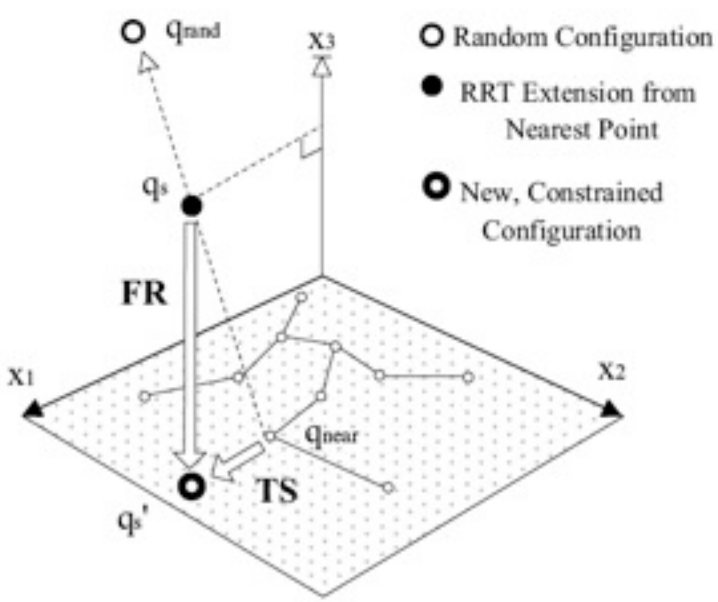
- In joint space - the constraint manifold is complicated
- sampling based planners sample in joint space
  - ✓ far away from constrained manifold
  - ✓ need to project samples back onto constraint manifold



Picture from Stilman M., “Task Constrained Motion Planning in Robot Joint Space”, in IROS 2007

Dealing with constraints

- In joint space - the constraint manifold is complicated
- sampling based planners sample in joint space
  - ✓ far away from constrained manifold
  - ✓ need to project samples back onto constraint manifold



- Most constraints are expressed more naturally in cartesian space
  - ✓ e.g. holding water level => restricting roll and pitch of the end-effector

Picture from Stilman M., "Task Constrained Motion Planning in Robot Joint Space", in IROS 2007

Constrained Planning

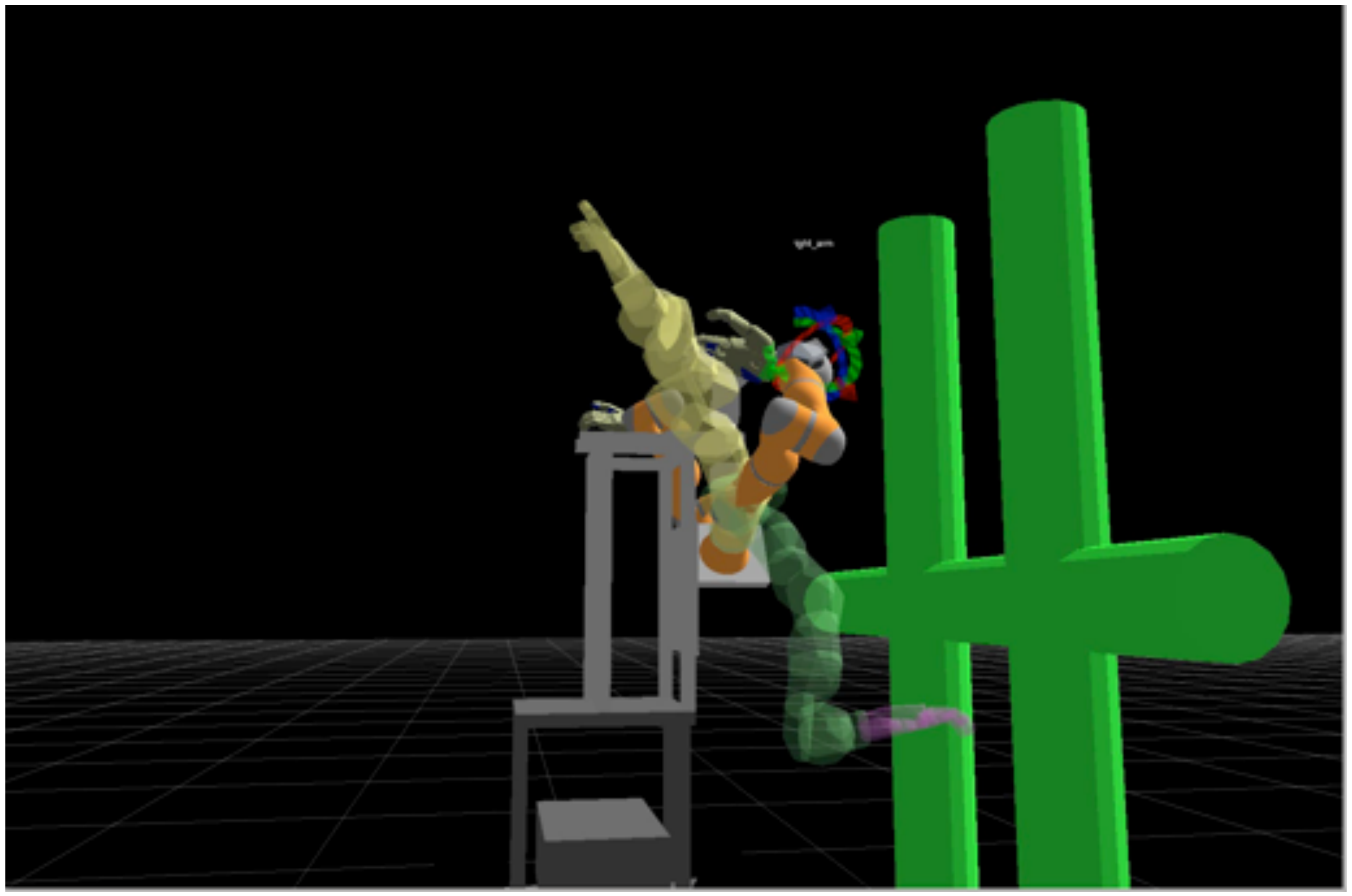
- Exploit geometry of the PR2
  - joint limits imply only 1 IK solution branch exists most of the time
  - plan in cartesian end-effector coordinates + redundancy (shoulder roll joint angle)
  - no need to do constraint projections in joint space
    - ✓ just shrink region available for sampling!

Willow Garage

Constrained Planning

What's next?

- New tools
  - go from URDF to complete planning in a few quick steps
  - visualize everything along the way
  - interactive GUI to test out planning



Topic of next talk by Gil Jones

Thank you!

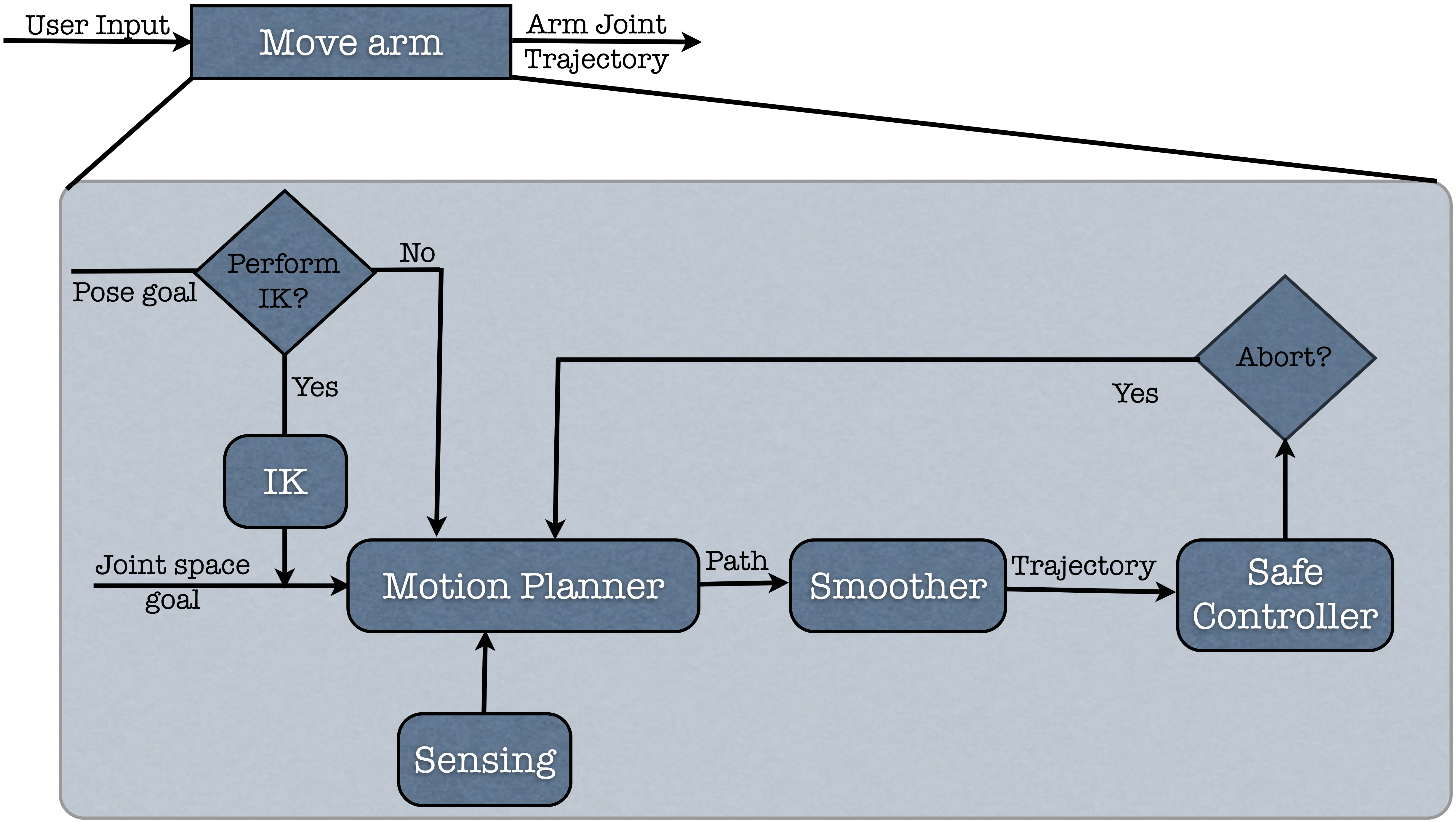
More resources:

1. http://www.ros.org/wiki
2. http://answers.ros.org
3. ros-users mailing lists
4. http://www.ros.org/wiki/arm_navigation

Willow Garage

Extra Slides

System Architecture



System Architecture