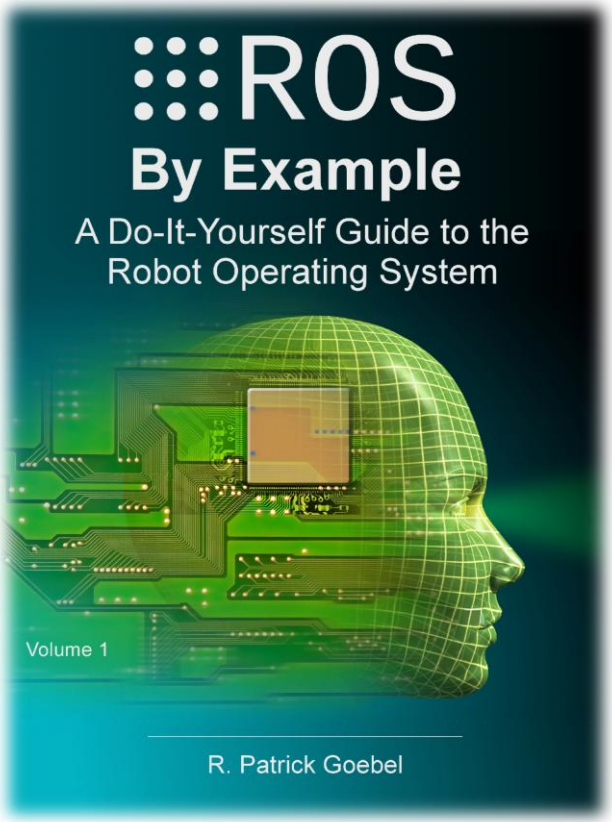
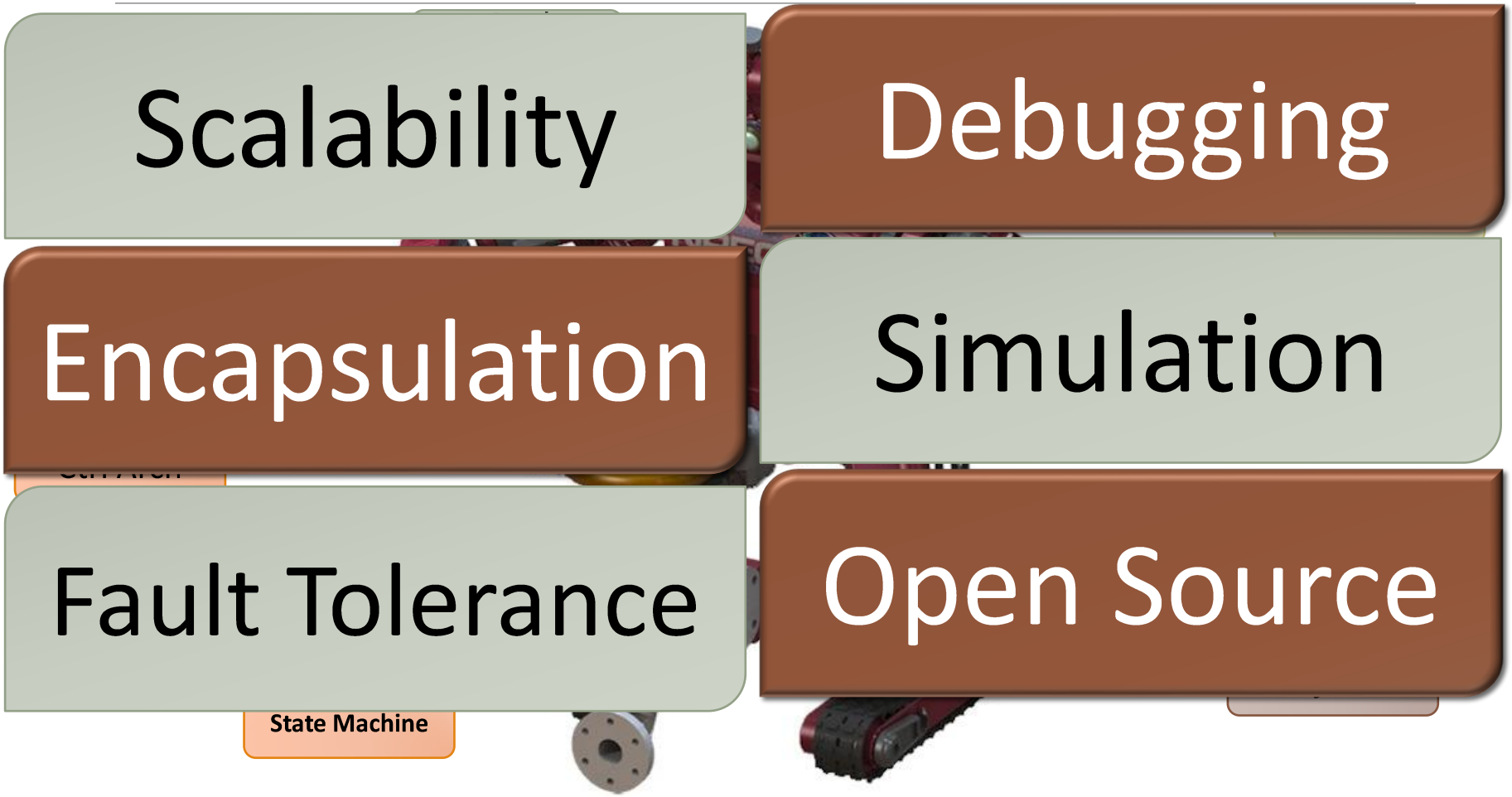


ROS

Overview & Simulation

Dr. Juan Rojas
Sun Yat Sen University 中山大学
School of Software 软件学院

Motivation


Scalability
Debugging
Encapsulation
Simulation
Fault Tolerance
Open Source
State Machine

History

STAIR: STanford Artificial Intelligence Robot

Artificial Intelligence Laboratory, Computer Science Department, Stanford University

Applications Overview

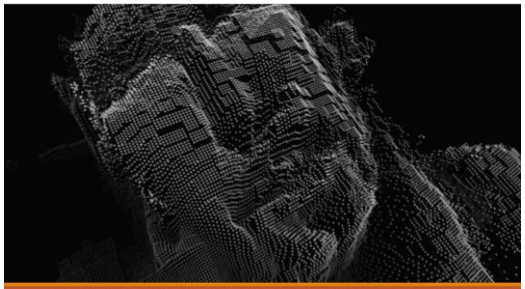
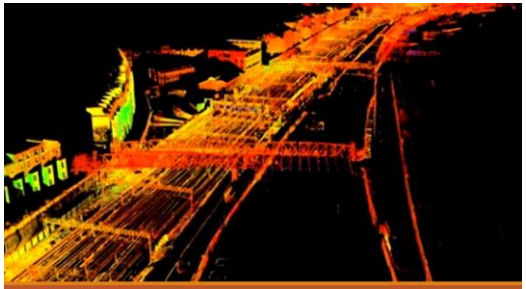


Image Processing

Controlling a Mobile Robot

Manipulation



SLAM and Navigation

Combining Vision and Navigation

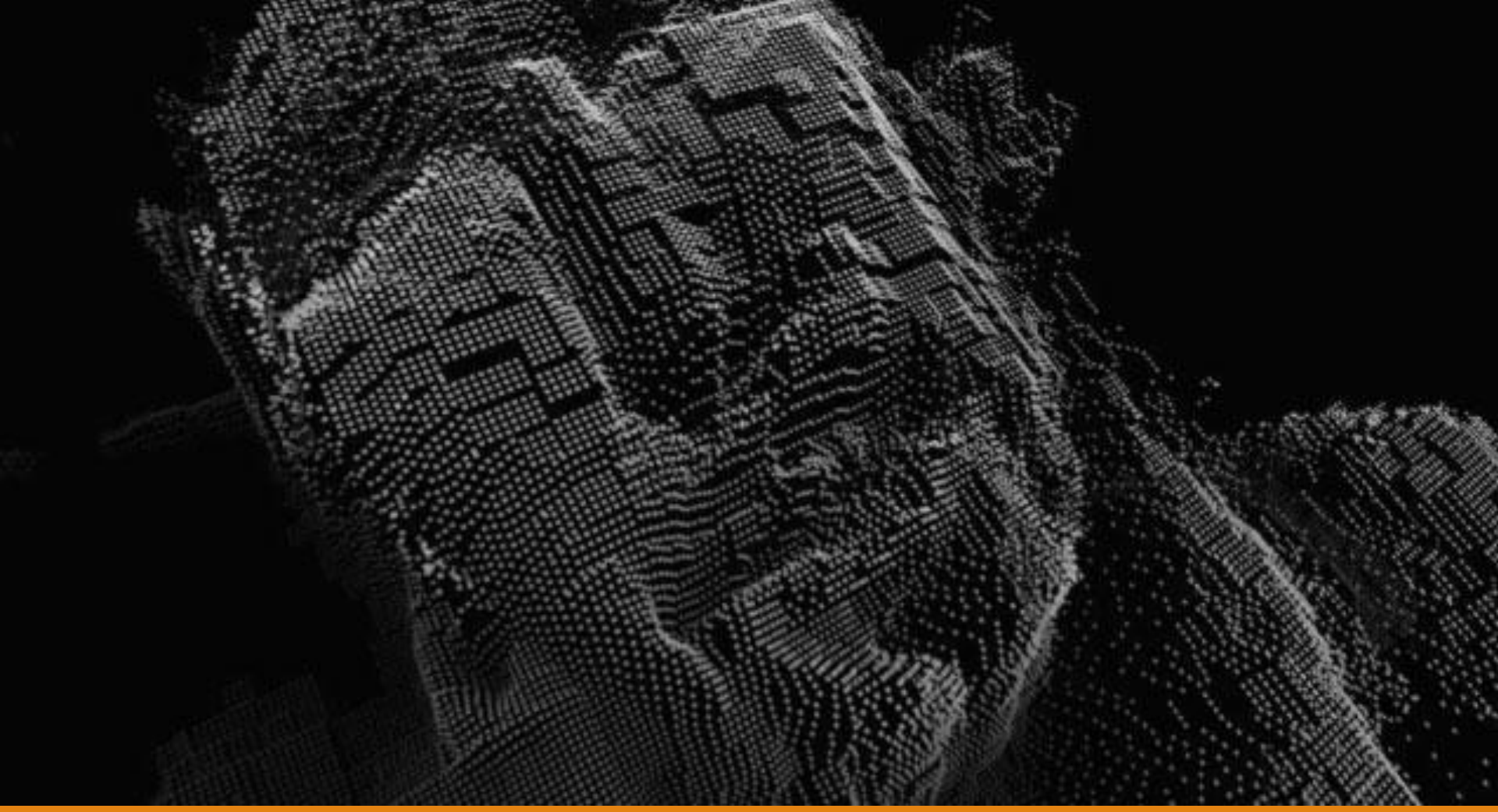
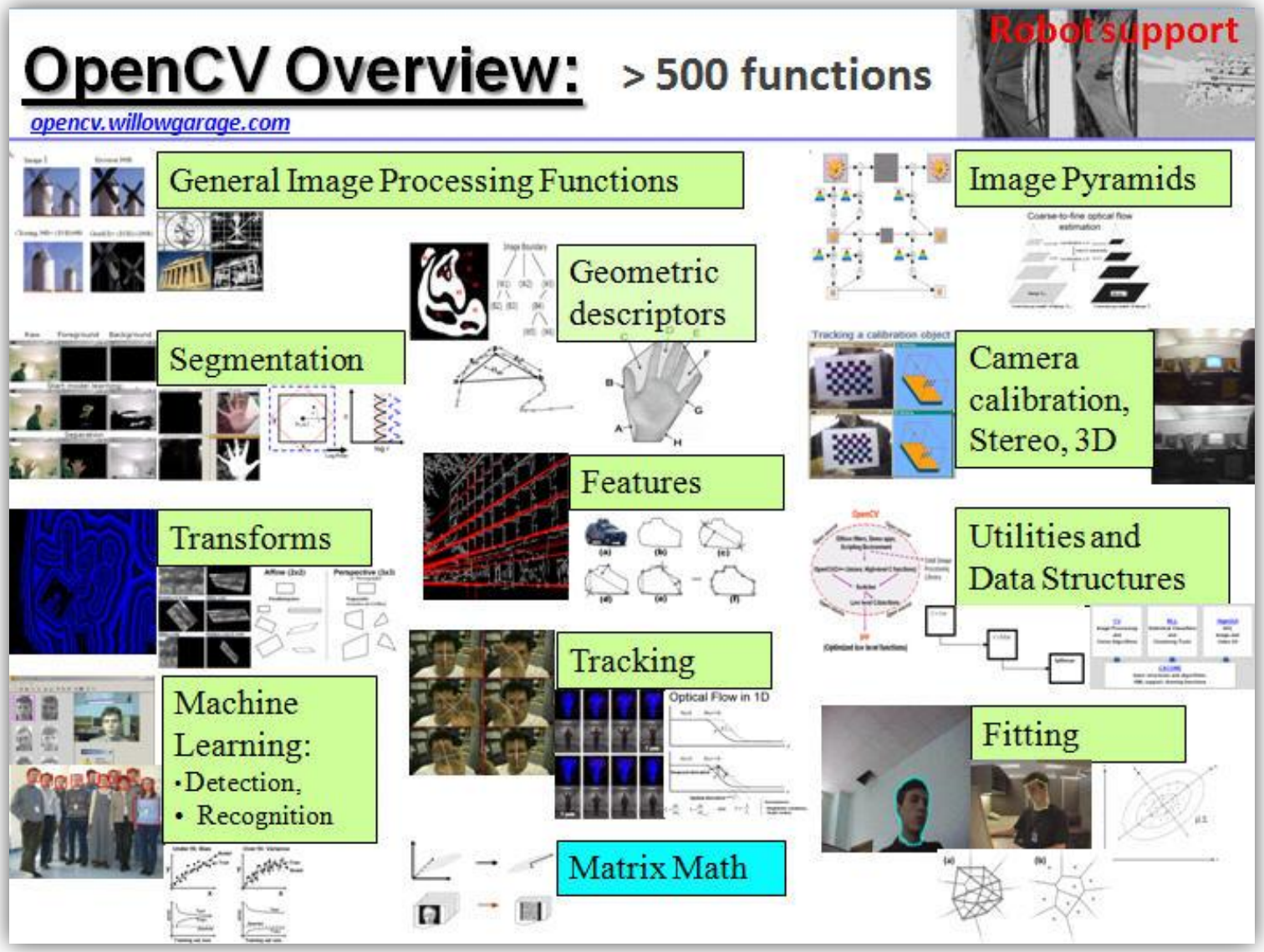
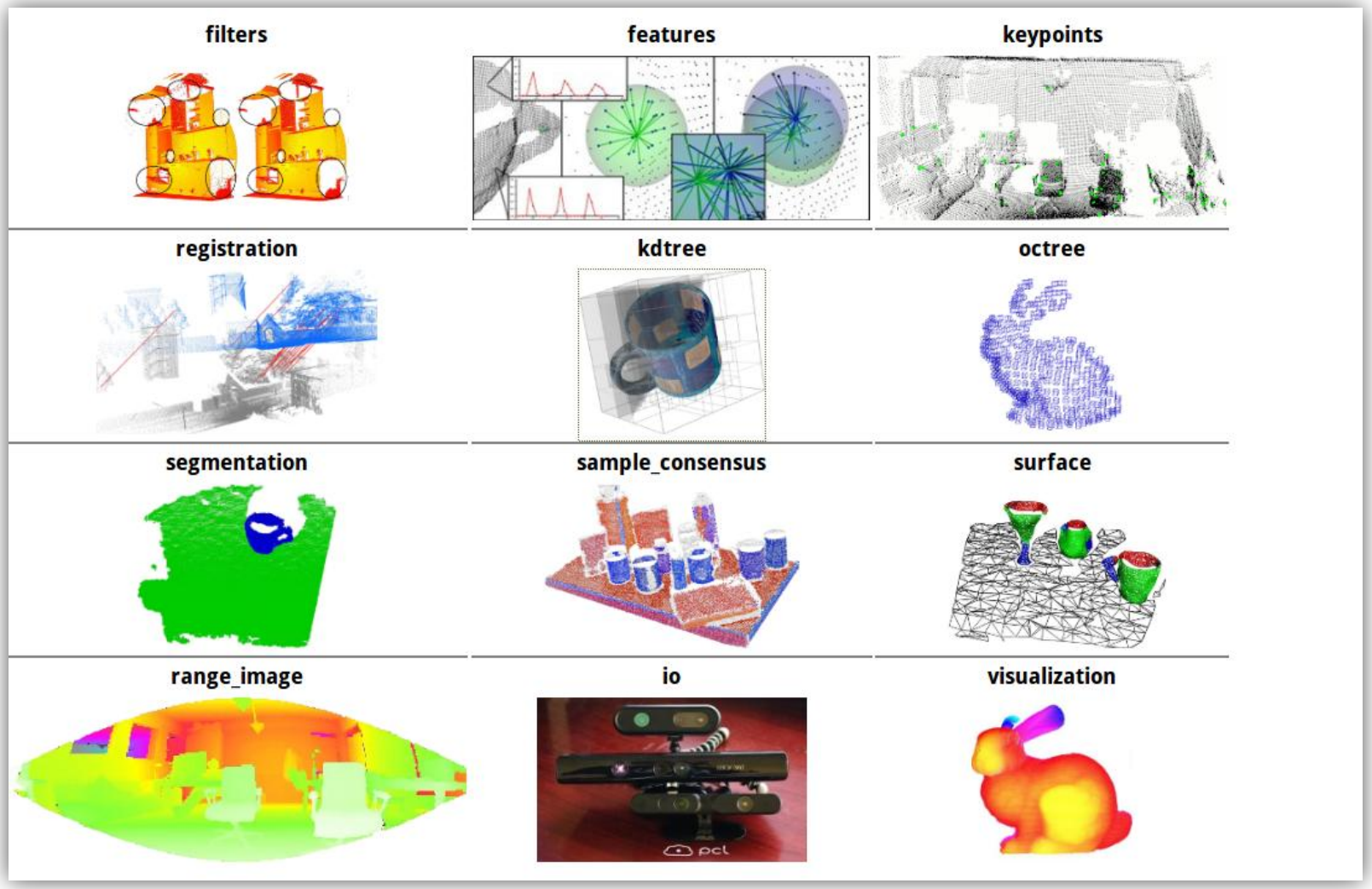


Image Processing

OpenCV Overivew



OpenPCL


filters
features
keypoints
registration
kdtree
octree
segmentation
sample_consensus
surface
range_image
io
visualization
pcl

Edge Detection

- Kinnect Drivers:

```
roslaunch rbx1_vision openni_node.launch
```

- Edge Detection:

```
rosrun rbx1_vision cv_bridge_demo.py
```

Advanced Image Processing

Face Features & Key-Point Detection & Optical Flow

- Kinect: launch the OpenNI Driver:

```
roslaunch rbx1_vision openni_node.launch
```

- Run the face tracker:

```
roslaunch rbx1_vision face_tracker2.launch
```

Face Velocity Tracking

- Online Face Tracker Analysis

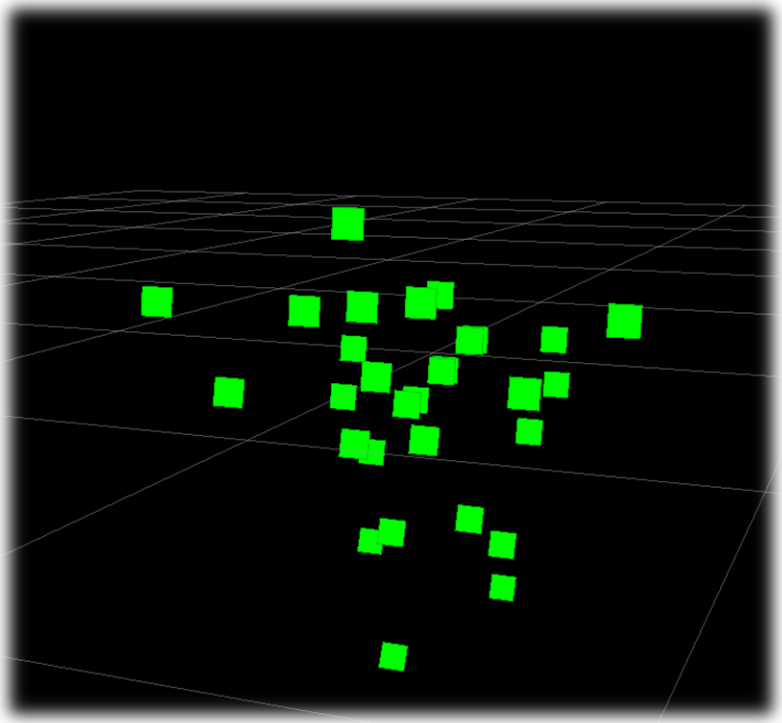
1. Launch: Face tracker.
2. Launch: ROI tracker
3. Plot: Results in rxplot

```
roslaunch rbx1_apps object_tracker.launch
```

```
rxplot –p 15 /cmd_vel/angular/z
```

Skeleton Markers

- Follow Body Motion.
- Useful:
  Learning from Demonstration.



```
roslaunch rbx1_vision openni_node.launch
roslaunch skeleton_markers markers_from_tf_groovy.launch
rosrun rviz rviz -d `rospack find \skeleton_markers`/markers_from_tf_groovy.rviz
```

Face Velocity Tracking

Online Face Tracker Analysis

1. Launch: Face tracker.
2. Launch: ROI tracker
3. Plot: Results in rxplot

```
roslaunch rbx1_apps object_tracker.launch
```

```
rxplot -p 15 /cmd_vel/angular/z
```

Controlling a Mobile Robot

Teleoperating your Robot

- ROS move-base controller subscribes to /cmd_vel and tells drivers how to move.


```
roslaunch rbx1_bringup fake_pi_robot.launch
```

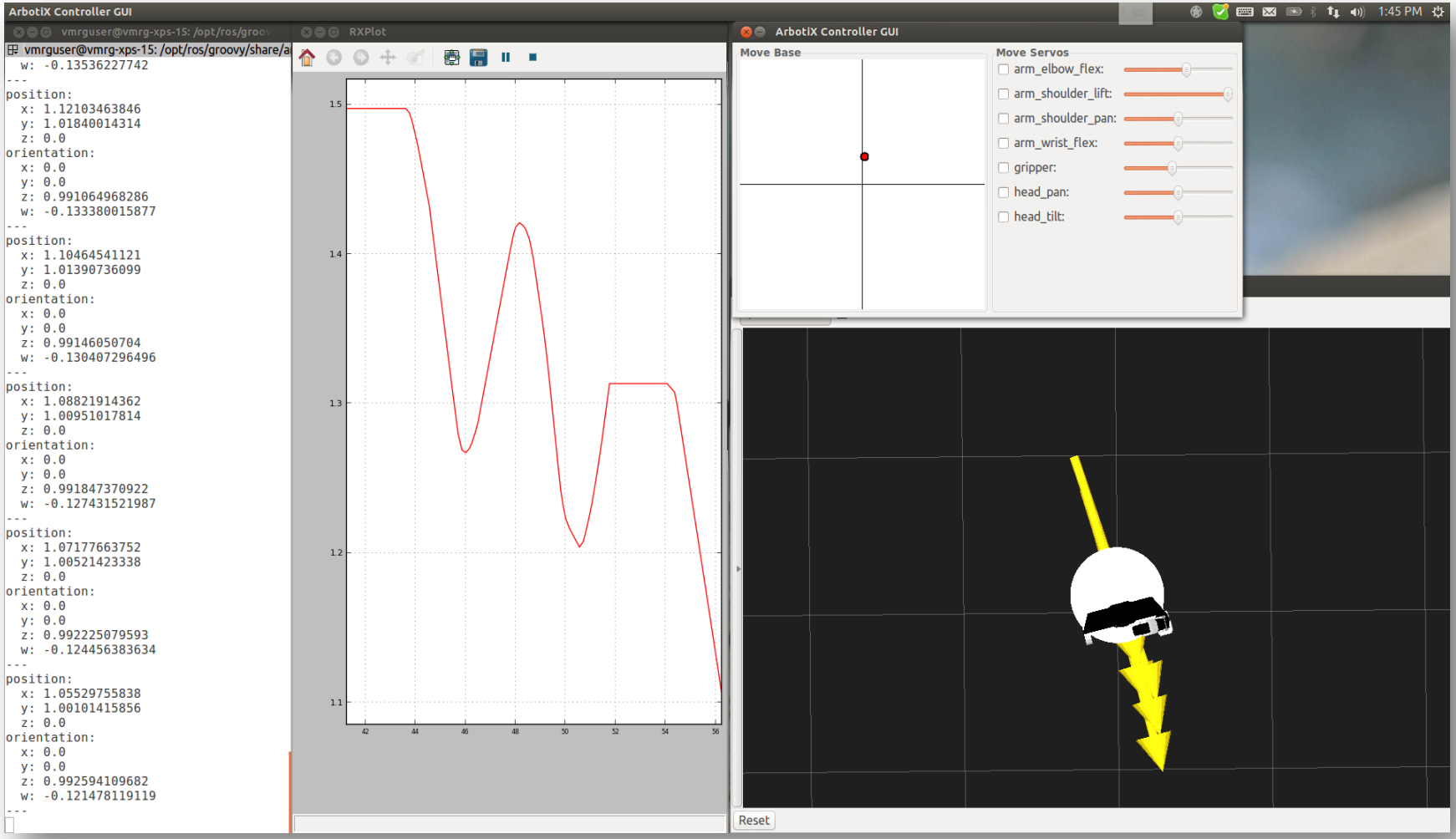
```
rosrun rviz rviz -d `rospack find rbx1_nav`/sim.rviz
```

- Teleoperate with keyboard, joystick, GUI.

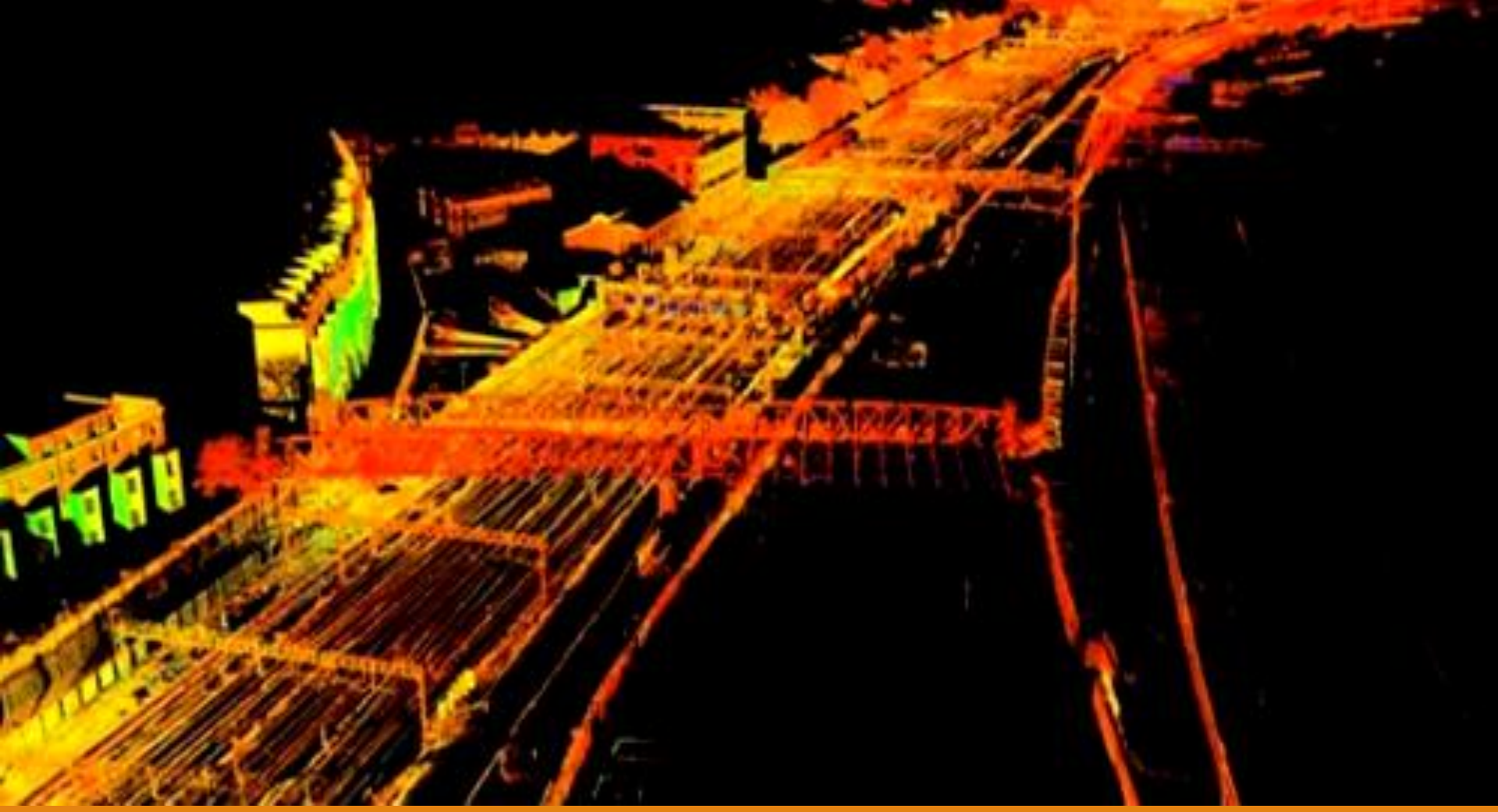
```
Keyboard:        roslaunch rbx1_nav keyboard_teleop.launch
```

```
GUI:             arbotix_gui
```

Analysis


ArbotiX Controller GUI
1:45 PM
vmrguser@vmrg-xps-15: /opt/ros/groovy/share/a
w: -0.13536227742
---
position:
x: 1.12103463846
y: 1.01840014314
z: 0.0
orientation:
x: 0.0
y: 0.0
z: 0.991064968286
w: -0.133380015877
---
position:
x: 1.10464541121
y: 1.01390736099
z: 0.0
orientation:
x: 0.0
y: 0.0
z: 0.99146050704
w: -0.130407296496
---
position:
x: 1.08821914362
y: 1.00951017814
z: 0.0
orientation:
x: 0.0
y: 0.0
z: 0.991847370922
w: -0.127431521987
---
position:
x: 1.07177663752
y: 1.00521423338
z: 0.0
orientation:
x: 0.0
y: 0.0
z: 0.992225079593
w: -0.124456383634
---
position:
x: 1.05529755838
y: 1.00101415856
z: 0.0
orientation:
x: 0.0
y: 0.0
z: 0.992594109682
w: -0.121478119119

RXPlot
1.5
1.4
1.3
1.2
1.1
42
44
46
48
50
52
54
56
ArbotiX Controller GUI
Move Base
Move Servos
arm_elbow_flex:
arm_shoulder_lift:
arm_shoulder_pan:
arm_wrist_flex:
gripper:
head_pan:
head_tilt:
Reset



SLAM and Navigation

Navigation

1. Move to goal with empty map

```
roslaunch rbx1_bringup fake_turtlebot.launch
```

2. Move-Base Controller.

```
roslaunch rbx1_nav fake_move_base_blank_map.launch
```

3. See Simulation

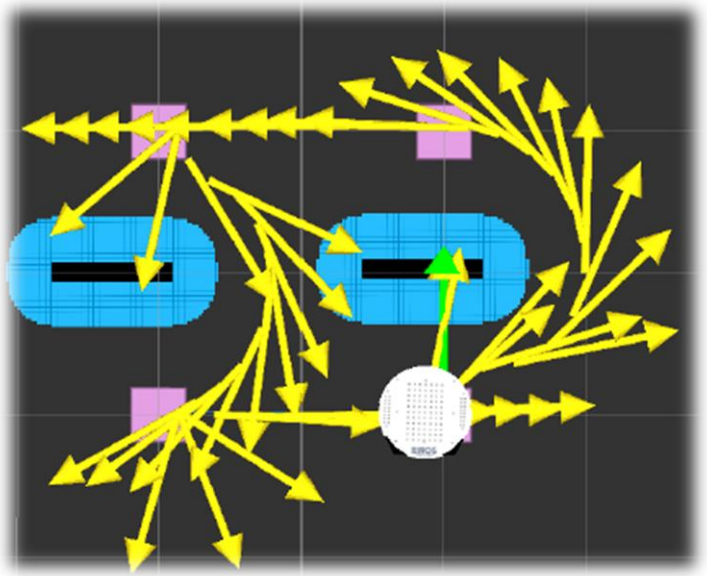
```
rosrun rviz rviz —d `rospack find rbx1_nav`/nav.rviz
```

4. Pass Commands

```
rostopic pub -1 /cmd_vel geometry_msgs/Twist '{linear: {x: 0.2, y: 0, z: 0}, angular: {x: 0, y: 0, z: 0}}'; rostopic pub -r 10 /cmd_vel geometry_msgs/Twist '{linear: {x: 0.2, y: 0, z: 0}, angular: {x: 0, y: 0, z: 0.5}}'
```

Collision Avoidance

- Costmaps recomputed to find optimal path.



```
roslaunch rbx1_bringup fake_turtlebot.launch
```

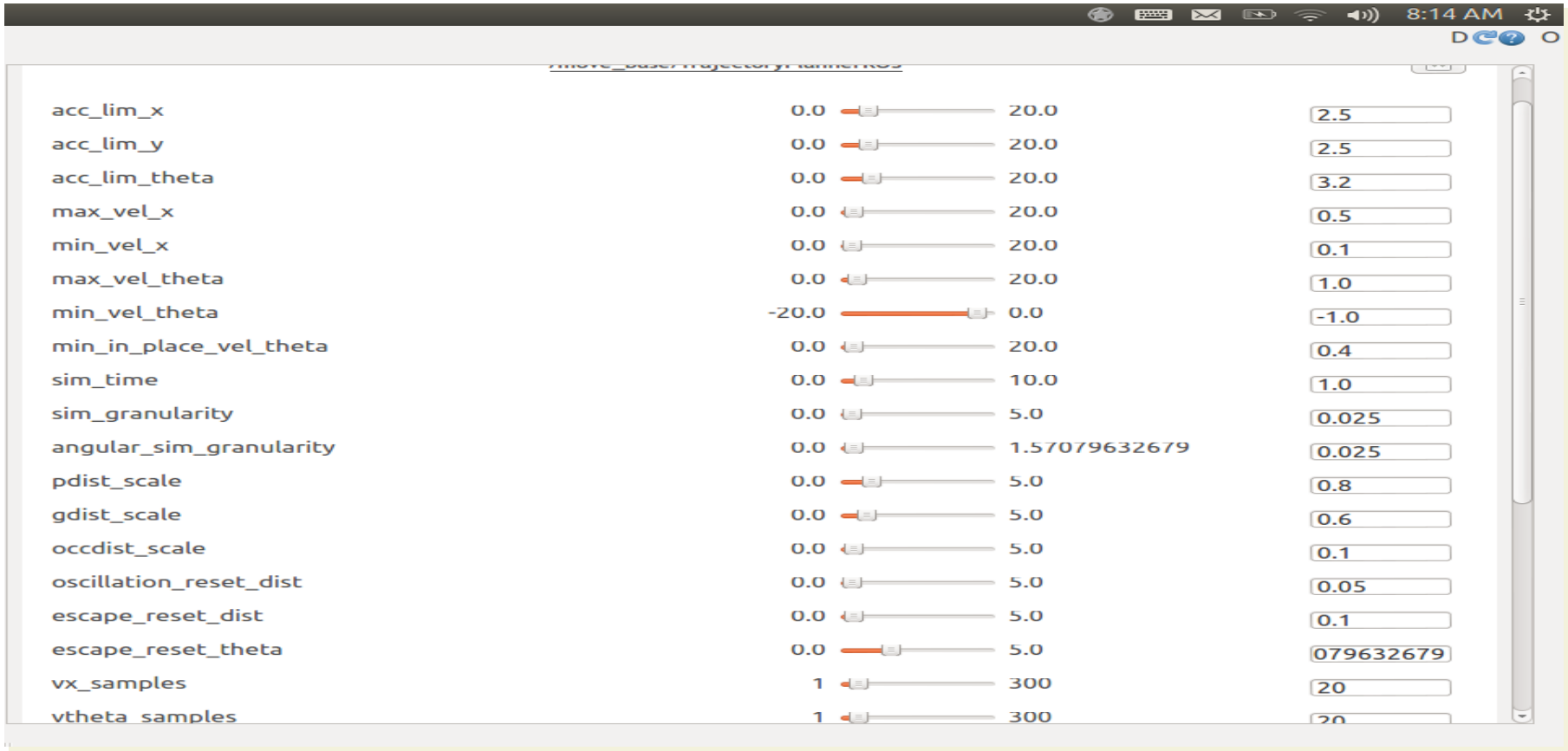
```
roslaunch rbx1_nav fake_move_base_obstacle.launch
```

```
rosrun rviz rviz —d `rospack find rbx1_nav`/nav.rviz
```

```
rosrun rbx1_nav move_base_square.py
```

8.2 Varying Navigation Parameters

- Online Parameter Adjustment



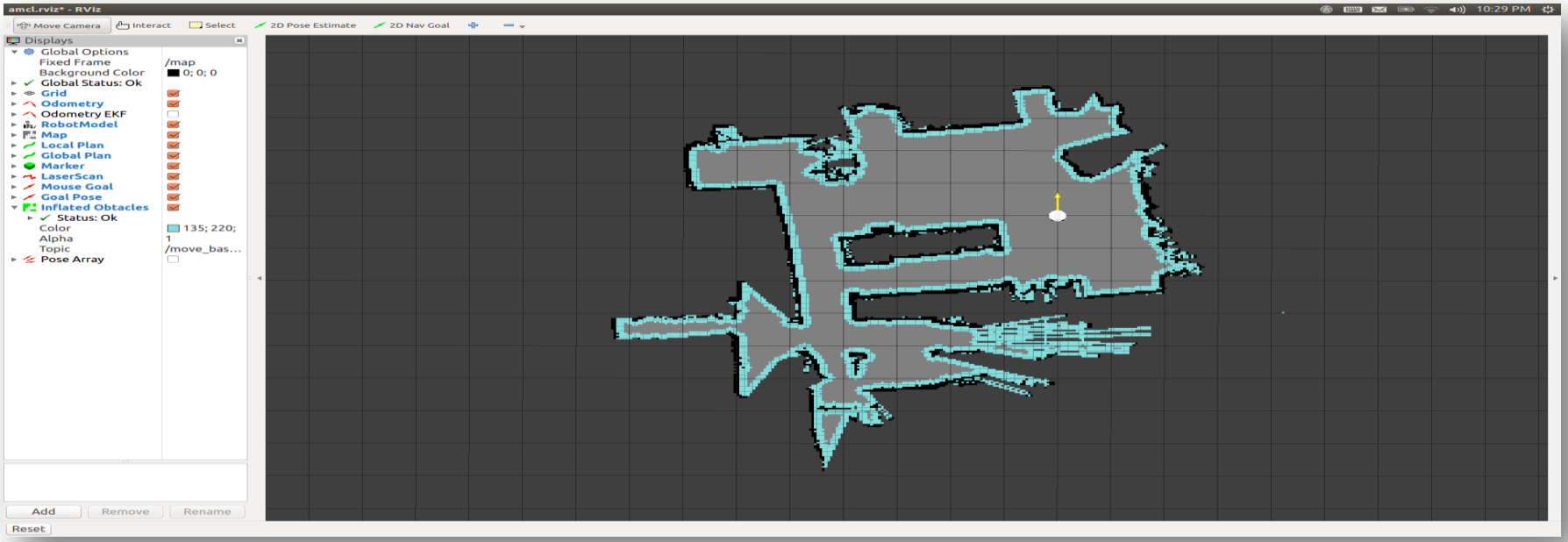
SLAM & Navigation

- Map. Localize. Motion Planning.

```
roslaunch rbx1_bringup fake_turtlebot.launch
```

```
roslaunch rbx1_nav fake_amcl.launch map:=test_map.yaml
```

```
rosrun rviz rviz -d `rospack find rbx1_nav`/amcl.rviz
```



Combining Vision and Navigation

Person Follower

- Turtlebot_follower node
  - Does not recognize person
  - Looks for moving objects within a certain distance
  - Tells robot how close to follow
  - Program commands robot to move so as to follow blob

```
roslaunch rbx1_vision openni_node.launch
```

```
roslaunch rbx1_apps follower.launch
```

```
roslaunch rbx1_bringup fake_turtlebot.launch
```

```
rosrun rviz rviz -d `rospack find rbx1_nav`/sim.rviz
```

Manipulation

PR2 Manipulation Simulation

- Solve Kinematics and Dynamics Easily
- Avoid Collisions
- Easy GUI
- Planning & Executing Motions
- Goal Acquisition through Sensor Integration

- Launch the PR2 in Gazebo

```
roslaunch pr2_gazebo pr2_empty_world.launch
```

- Launch the move it plan and execute file:

```
roslaunch pr2_moveit_config moveit_planning_execution.launch
```