



An open-source Robot Operating System

Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote,
Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Y. Ng

Computer Science Department, Stanford University
and
Willow Garage, Menlo Park, CA

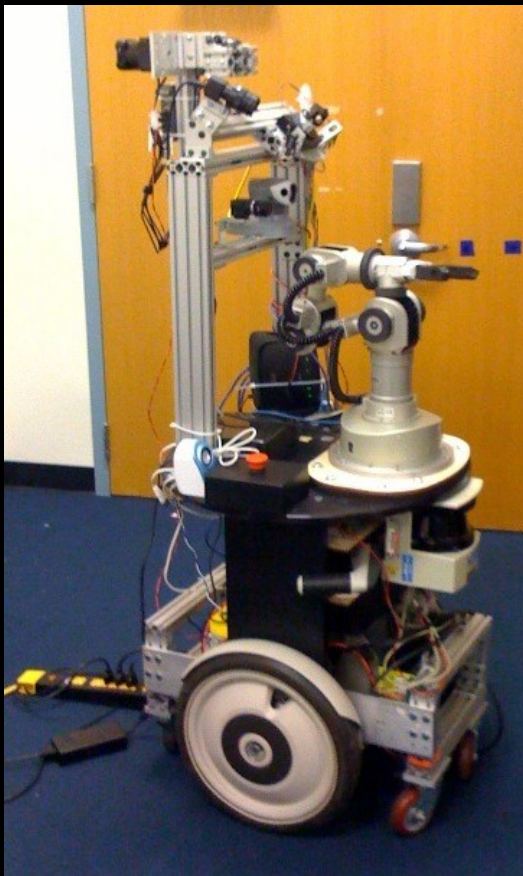
Motivation

- Robotics requires lots of software
- Especially on big robots
- Scale through modularity and abstraction
- Leverage work of others
- Promote replication of results: science!

Historical

- Stanford and Willow Garage are nearby
- We both run large, complex robots
- We both build complex software systems
- We had experience with various systems
 - Various in-house systems at Stanford and others
 - Player
 - ...
- We combined efforts on ROS
 - Meets our needs for large mobile manipulators
 - Hope it will help robotics in general

Target platforms



Design philosophy

- UNIX design:
 - Lots of small programs
 - Programs debugged in isolation
 - Piped or scripted to do complicated things
 - Spectacularly successful
- ROS hopes to do this for robotics

What is ROS?

- Sits on top of POSIX: “meta operating system”
- UNIX provides filesystem, networking, etc.
- ROS provides transforms, vision, nav, etc.
- Modularity via easy, fast P2P messaging
- Tools for collaborative development
- Hackable at all levels
- BSD license

Work cycle

- Coding time:
 - Write small programs
 - Debug them in isolation
- Run time:
 - Launch a bunch of small programs
 - ROS helps them find each other
 - They form a (local, legal, friendly) botnet

ROS overview

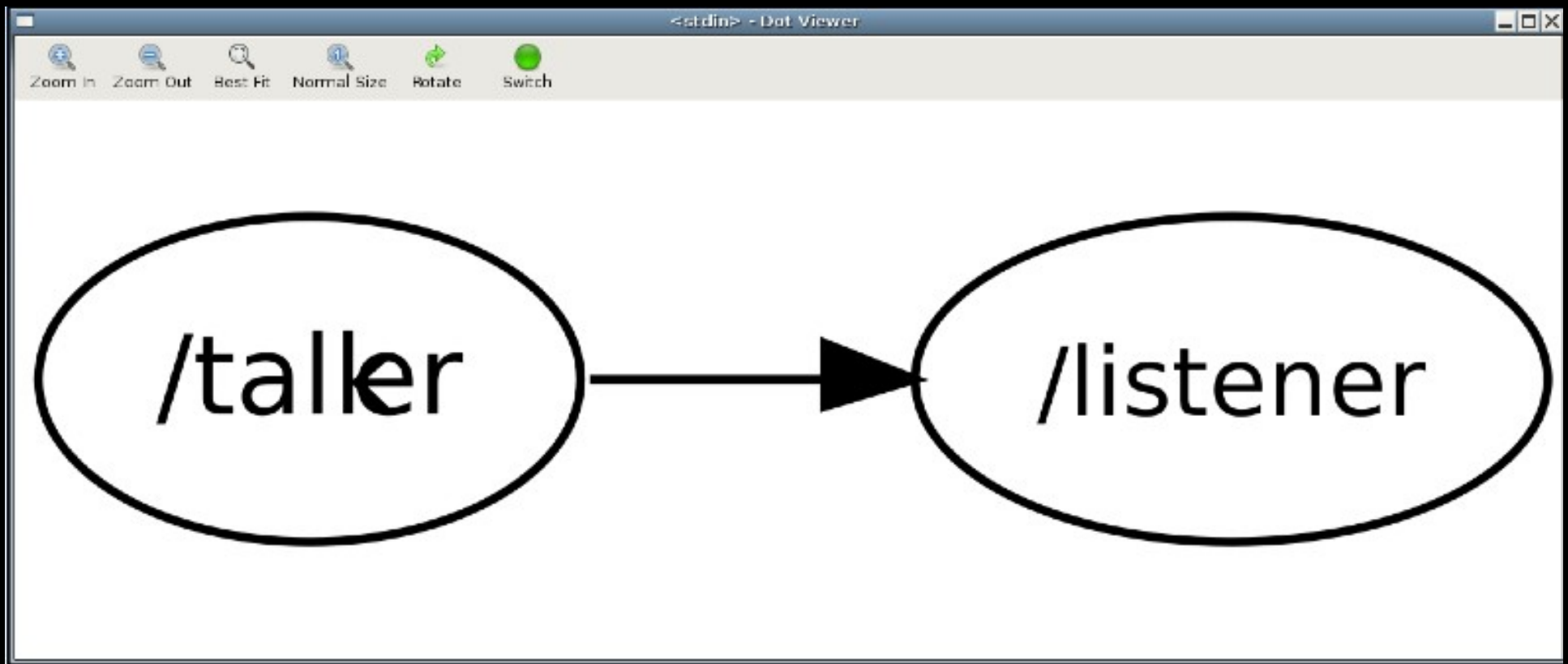
- At runtime, ROS botnet is a graph
 - Nodes are programs using the ROS libraries
 - Edges are P2P communications between nodes
- Nodes communicate in two ways
 - Anonymous publish/subscribe: “laser”
 - RPC to named services: “classify_image”
- Message code autogenerated from simple IDL
- Master node: directory service, no data

ROS overview II

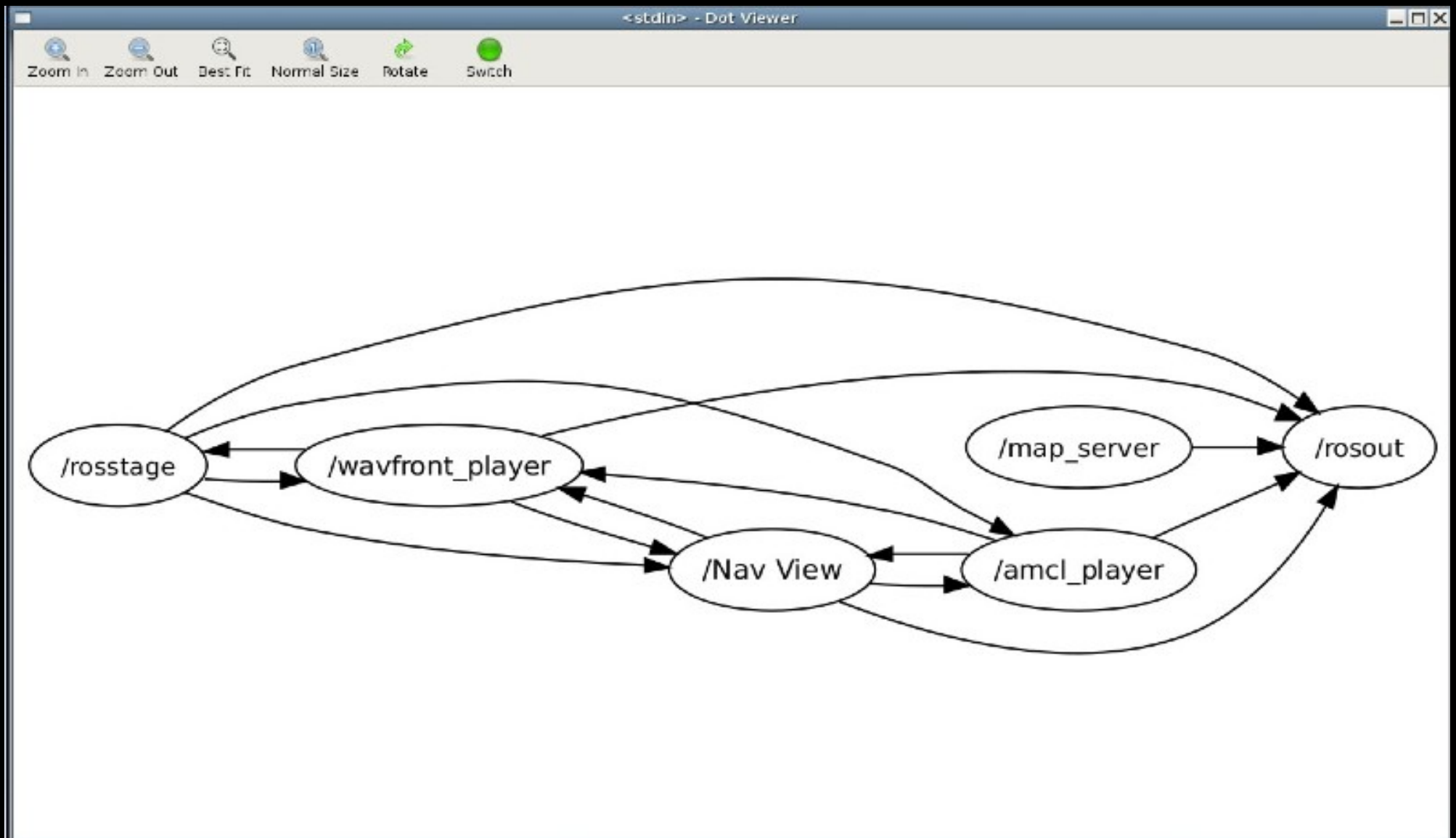
- Console-friendly tools, easily scriptable
- Loosely defined package manager
 - Package requirements: an XML file
 - Packages: hardware drivers, vision, numerics, controllers, nav, planners, messages, datasets, ...
 - Recursive build system
- Recursive dependency tracker / translator
- Packages wrap other open-source projects
 - OpenCV, Player, OGRE, Bullet, ...

Minimal ROS graph

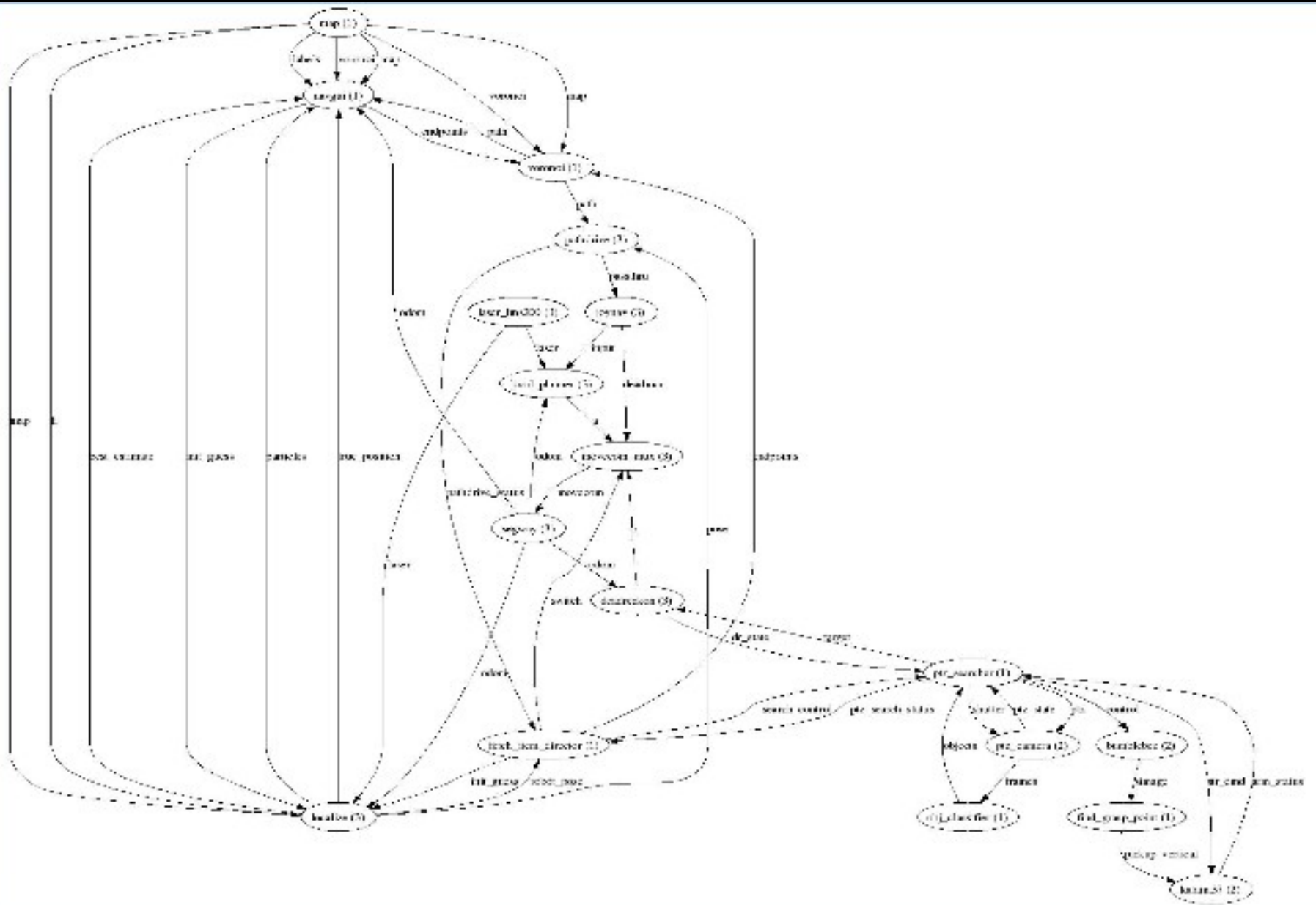
- One node blasts data to another node



2D navigation in simulation



Fetch a stapler



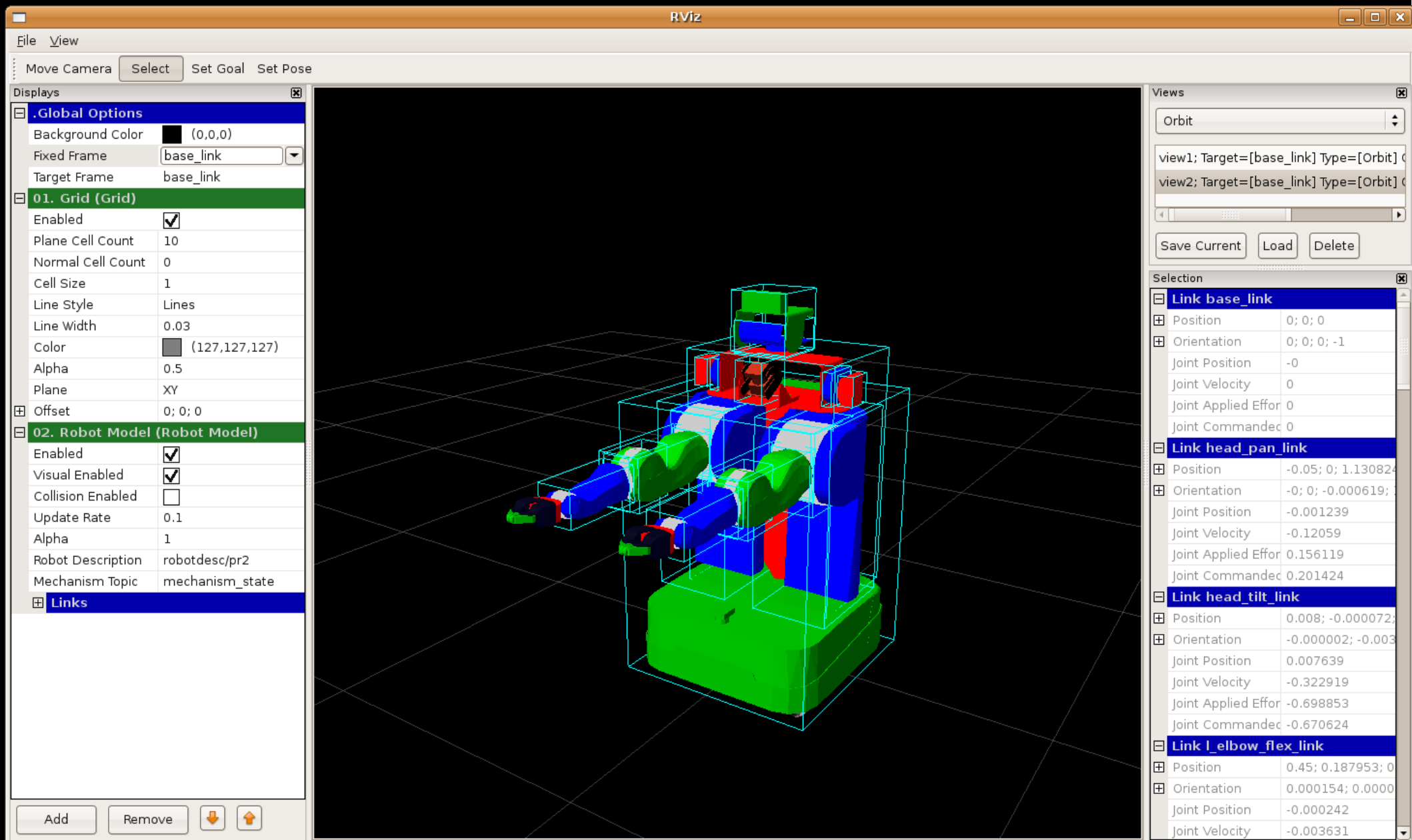
ROS coding philosophy

- Thin
 - Maximize amount of “standalone” code
 - ROS just wraps libraries so they can talk
- Multi-lingual
 - C++, Python, LISP, Octave, more coming
- Scaling via modularity (P2P)
- Tools-based
 - Push things out of the core library, into small tools
- Open!
 - Hosted at Sourceforge

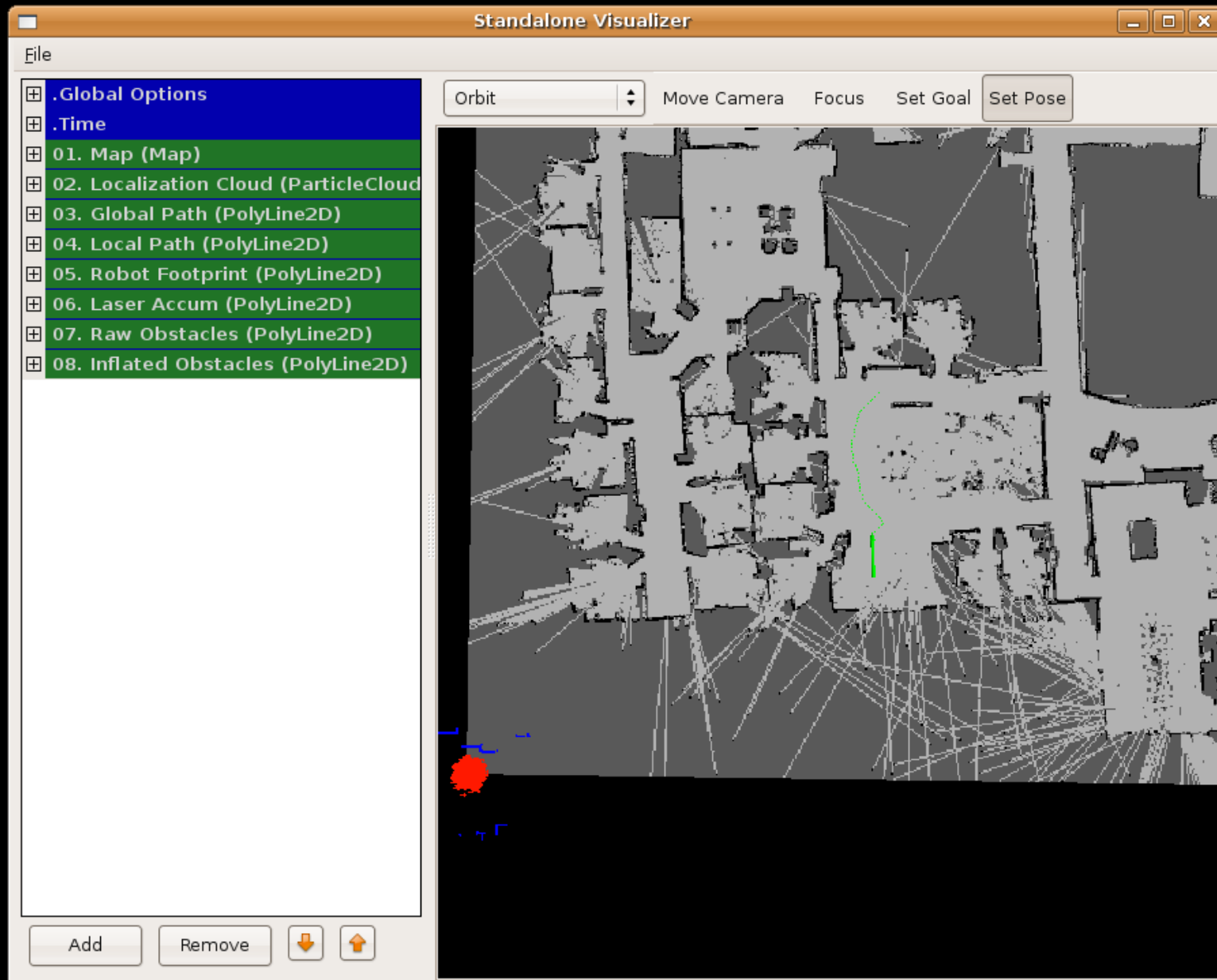
System Monitoring

- Distributed systems can be tough to monitor
- Lots of state in various places
- ROS graphs are dynamic
- Insert visualizers or console dumpers
- Script / pipe UNIX text tools on console dumps
- Plugin architecture to write visualizer panels

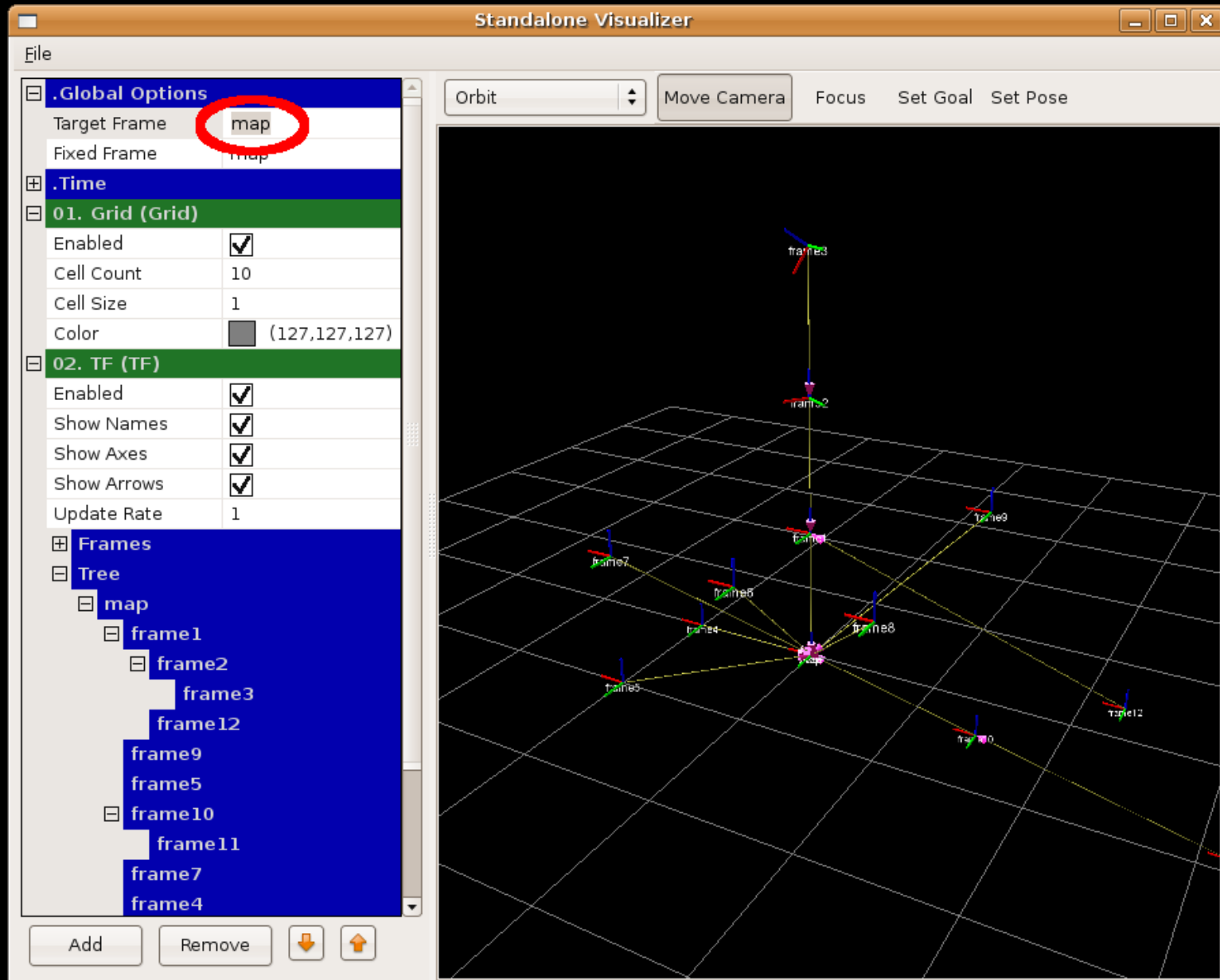
Visualizers



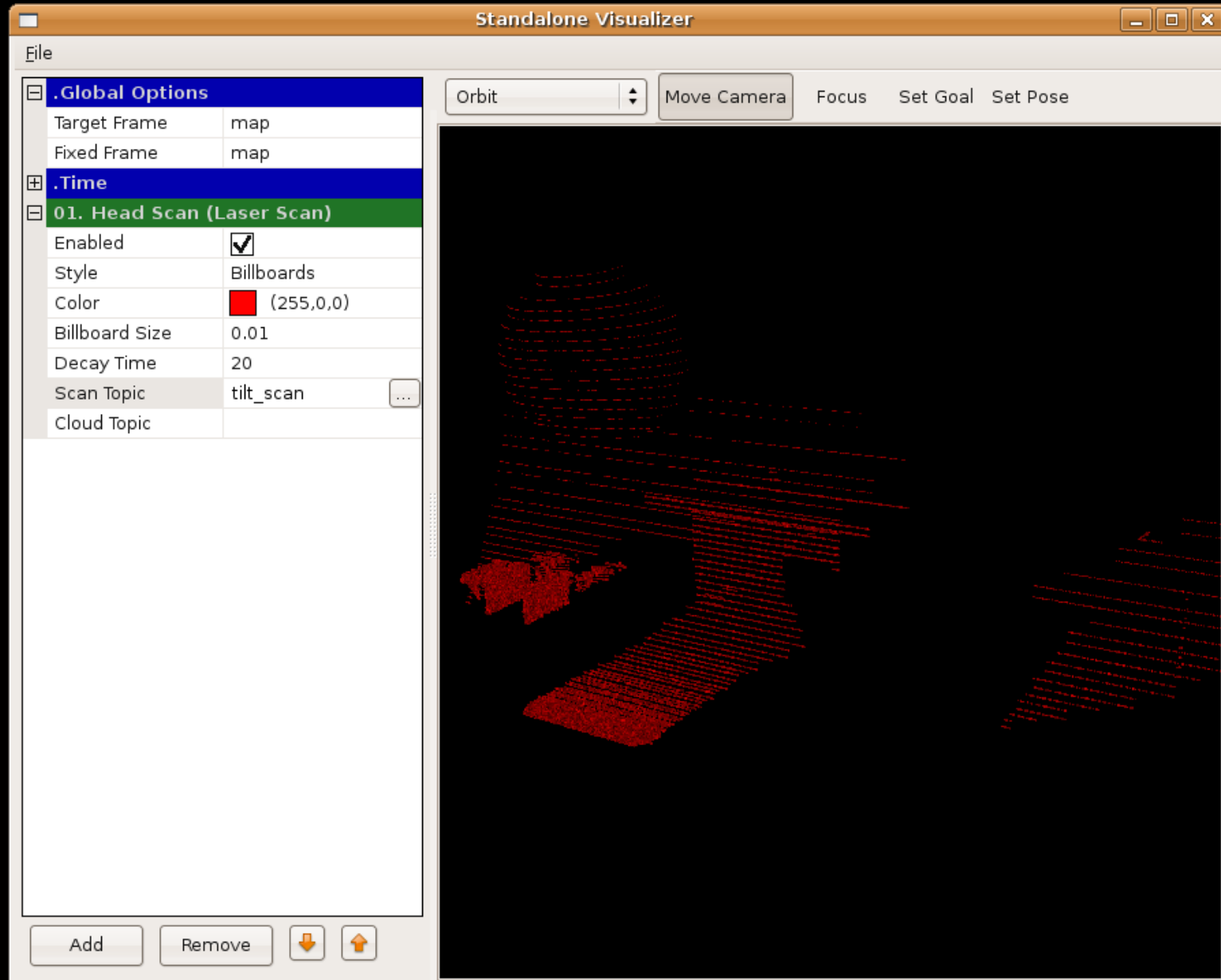
Visualizers



Visualizers



Visualizers



Visualization video

Computation

- Complex robots need lots of CPU time
 - Sensing, planning, etc.
- ROS: trivial to shift computation around
 - Several machines onboard robot
 - Cluster offboard
 - VM's in a cloud somewhere?
- Automated launching of processing nodes
- Monitoring tools for distributed logs

Robot descriptions

- Dynamics and kinematics encoded in XML
- Inspired by PR2, not tied to it
- Populates tree-based transform system
- “Put nodding laser data in map frame”
 - Nodding laser is on pan unit
 - Pan unit is on torso
 - Torso can twist and be raised / lowered
 - Torso sits on an omnidirectional base
 - Base is localized to map using nav system
- Transform system handles this transparently

File

+	Global Options
+	Time
+	01. Grid (Grid)
+	02. Origin Axes (Axes)
+	03. Visualization Markers (Markers)
-	04. Robot Model (Robot Model)
	Enabled <input checked="" type="checkbox"/>
	Visual Enabled <input checked="" type="checkbox"/>
	Collision Enabled <input type="checkbox"/>
	Update Rate 0.1
	Robot Description robotdesc/pr2
	Mechanism Topic mechanism_state
+	Links
-	05. 2D Pose: Odom (RobotBase2D)
	Enabled <input type="checkbox"/>
	Color ■ (255,25,0)
	Topic odom
	Position Tolerance 0.1
	Angle Tolerance 0.1
-	06. 2D Pose: Localized (RobotBase2D)
	Enabled <input checked="" type="checkbox"/>
	Color ■ (0,25,204)
	Topic localizedpose
	Position Tolerance 0.1
	Angle Tolerance 0.1
+	07. Map (Map)
+	08. Planning (Planning)
+	09. Stereo Full Cloud (Point Cloud)
+	10. Head Full Cloud (Point Cloud)
+	11. World 3D Map (Point Cloud)
-	12. Head Scan (Laser Scan)
	Enabled <input checked="" type="checkbox"/>
	Style Billboards
	Color ■ (255,0,0)
	Billboard Size 0.03
	Decay Time 100
	Scan Topic tilt_scan
	Cloud Topic
+	13. Floor Scan (Laser Scan)
+	14. Localization Cloud (ParticleCloud)
+	15. Global Path (Global Path)

Orbit

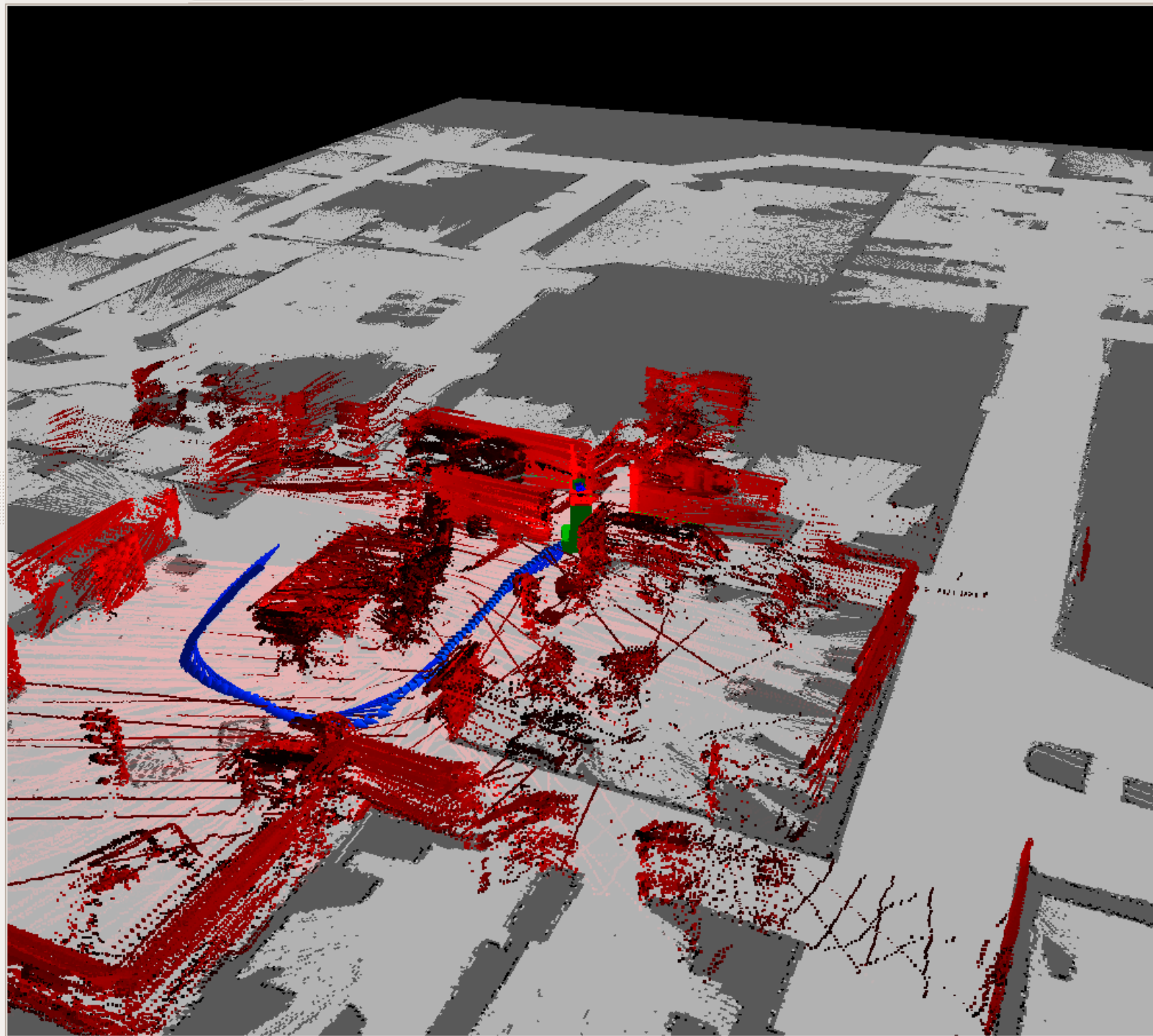


Move Camera

Focus

Set Goal

Set Pose



Add

Remove



Door video

Inventory video

ROS-ified algorithms

- Large and growing library of packages
- 2.5D navigation stack
 - Extensive testing underway (multiple km / day)
- Manipulation stack
 - Sampling-based planning
 - Grasp planning (various approaches)
- Perception
 - Visual odometry
 - 3d perception
- Many more (400+ packages currently)

Known weaknesses

- Overkill for really small robots
- No Windows port (yet)
- No IDE (yet); need to be comfortable at console
- Can't talk to other large codebases
 - Player, OROCOS, Orca, OpenRTM
 - Future: automated tools to bridge systems?

Summary

- ROS hopes to scale through modularity:
 - Peer-to-peer messaging
 - Hierarchical sets of packaged functionality: “nav”
- Tools for collaborative development
 - Simple package manager
 - Many package repositories
 - Wrapping other open-source projects
 - In use at >10 laboratories, wildly varying platforms
- <http://ros.sourceforge.net>



An open-source Robot Operating System

Morgan Quigley, Brian Gerkey, Ken Conley, Josh Faust, Tully Foote,
Jeremy Leibs, Eric Berger, Rob Wheeler, Andrew Y. Ng

Computer Science Department, Stanford University
and
Willow Garage, Menlo Park, CA