

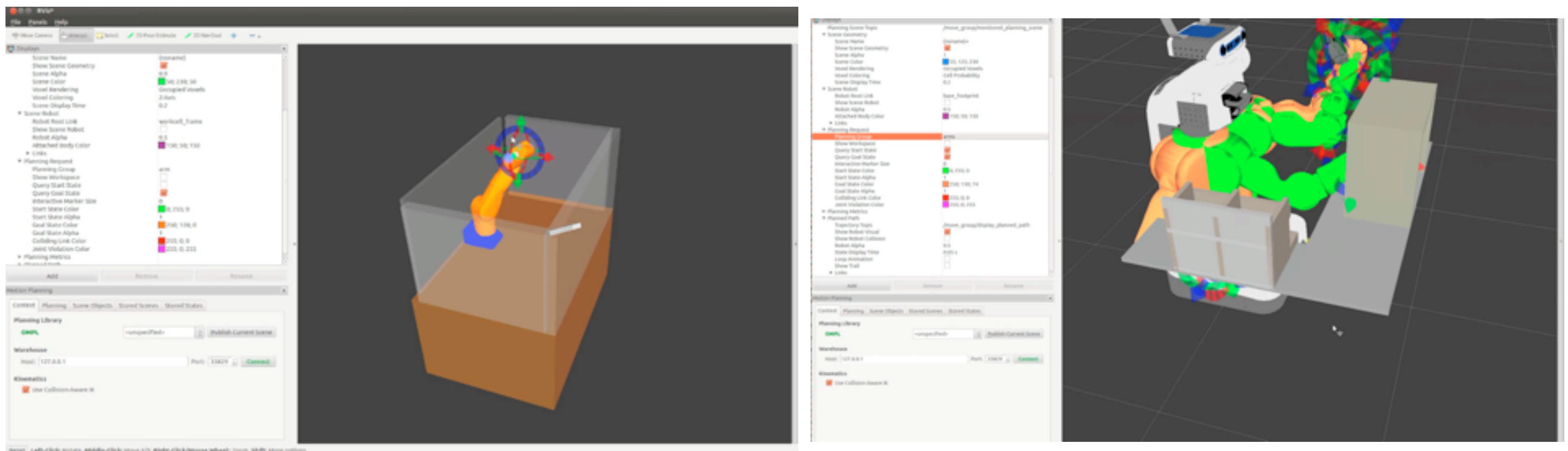


Sachin Chitta, Ioan Sucan, Acorn Pooley

with contributions from: Dave Coleman, Suat Gedikli, Mario Prats,
Matei Ciocarlie, Kaijen Hsiao, Jon Binney, Adam Leeper, Julius
Kammerl, David Gossow, Vincent Rabaud, Dave Hershberger and the
ROS and PR2 communities

What is MoveIt!

- MoveIt!- Software for building mobile manipulation applications
 - ❖ Motion Planning, Kinematics, Collision Checking integrated with Perception, Grasping, Control and Navigation for Mobile Manipulation



Motivation

- Build state of the art software platform for robotics applications and research
- “Simple things should be easy”
 - ❖ Provide out-of-the-box experience
 - easy to setup with new robots - Setup Assistant
 - ❖ Easy to use APIs - C++ and Python
- “Allow users to dive deeper to address harder problems”
 - ❖ Flexible platform - easy to add new components
- Performance
 - ❖ design for high performance

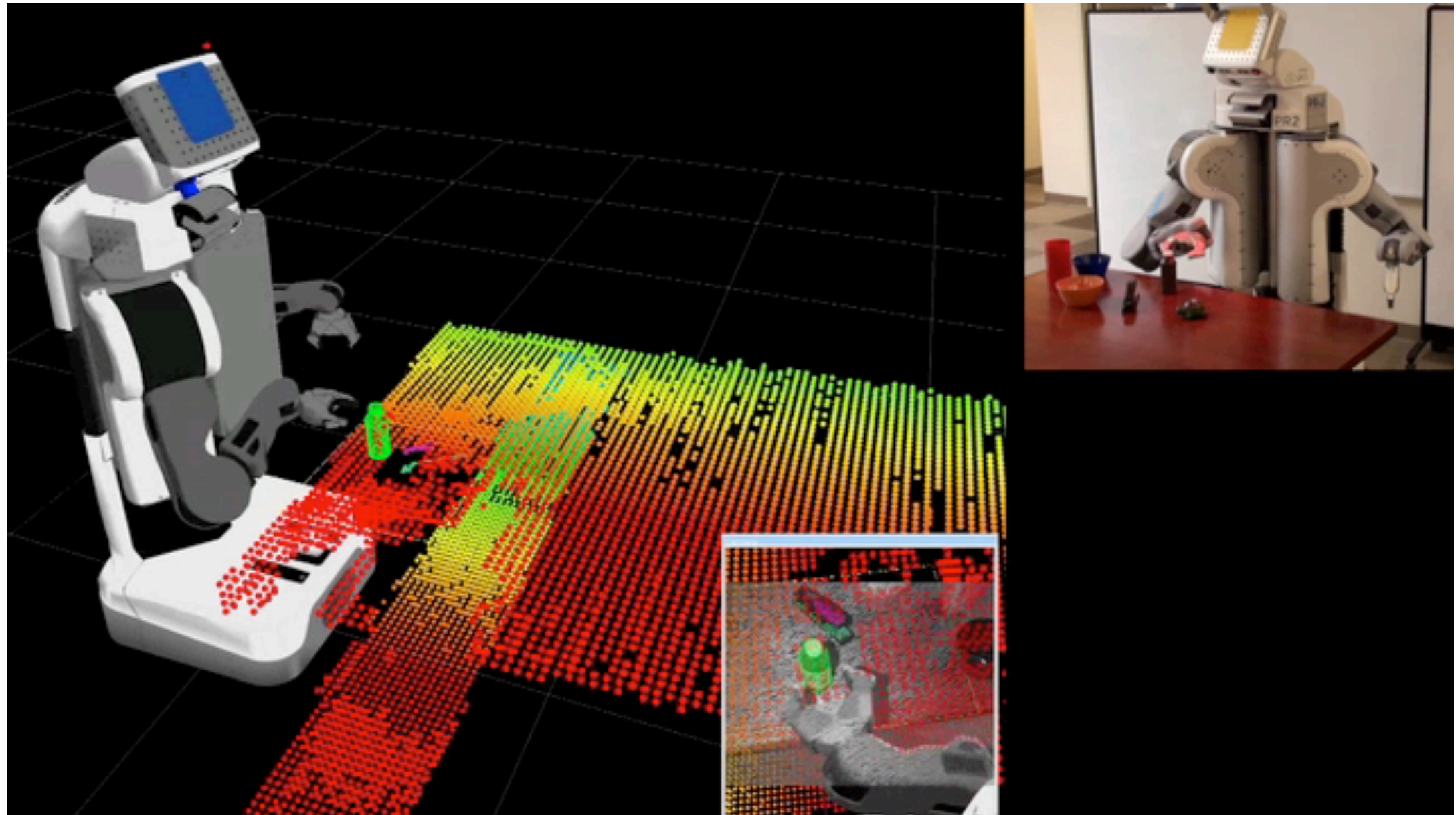
Motivation

- Developing new higher-level capabilities is time-consuming
 - ❖ building every capability from scratch is a waste of effort
- Environment awareness critical for new applications
 - ❖ Integrated 3D perception can provide situational awareness, improving safety, especially in human-robot collaborative tasks
- Motion Planning important for dynamic changing environments
 - ❖ essential for maintaining safety and performance in human-robot collaborative tasks
- Constrained manipulation tasks are hard to solve
 - ❖ increasingly complex types of constraints need to be handled

MoveIt!: Evolution

- MoveIt! has evolved from the arm_navigation and grasping pipeline set of stacks
 - ❖ essentially a rewrite from scratch
 - ❖ ROS API almost the same
 - ❖ incorporates lessons learnt

Mobile Manipulation: State of the art



Chitta, Jones, Ciocarlie, Hsiao, Sucas, 2011

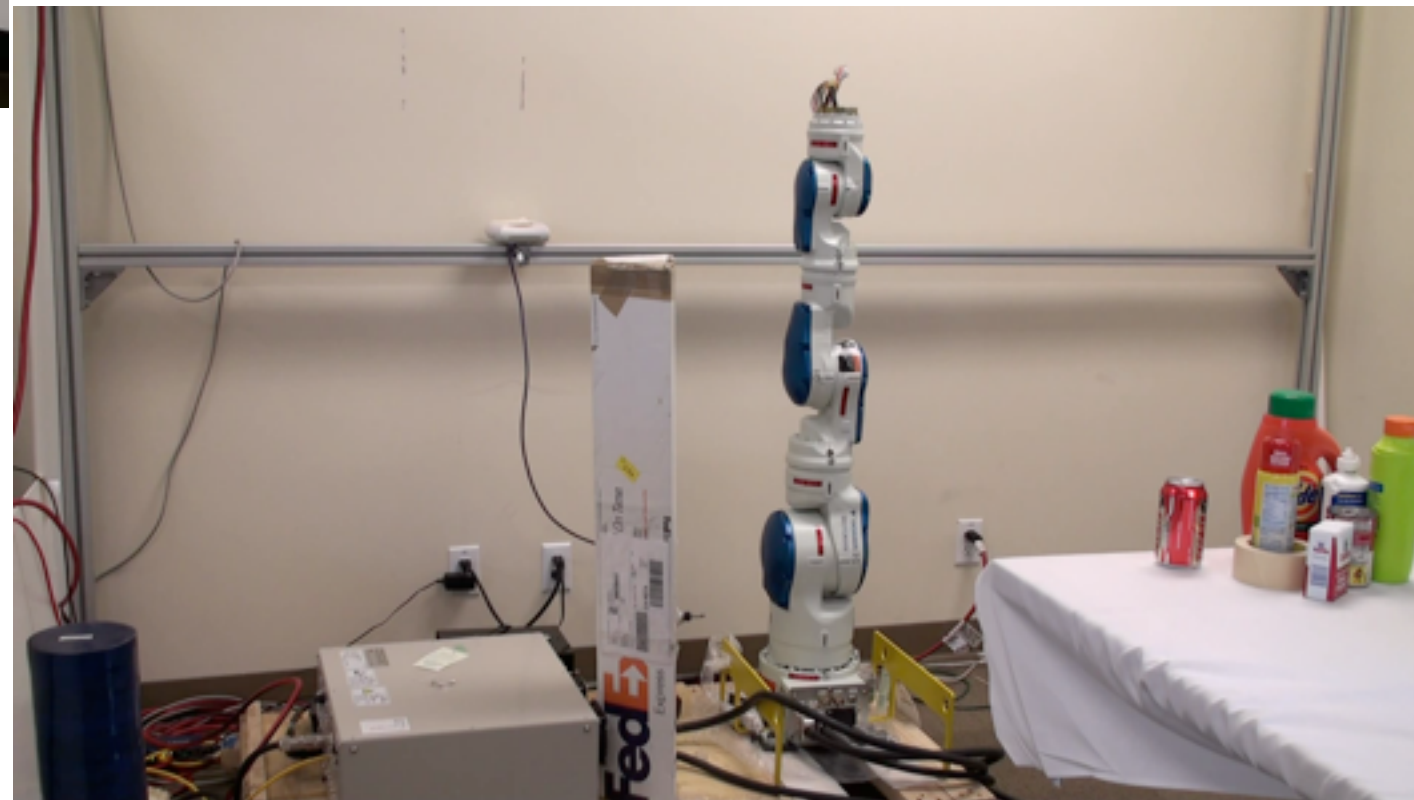
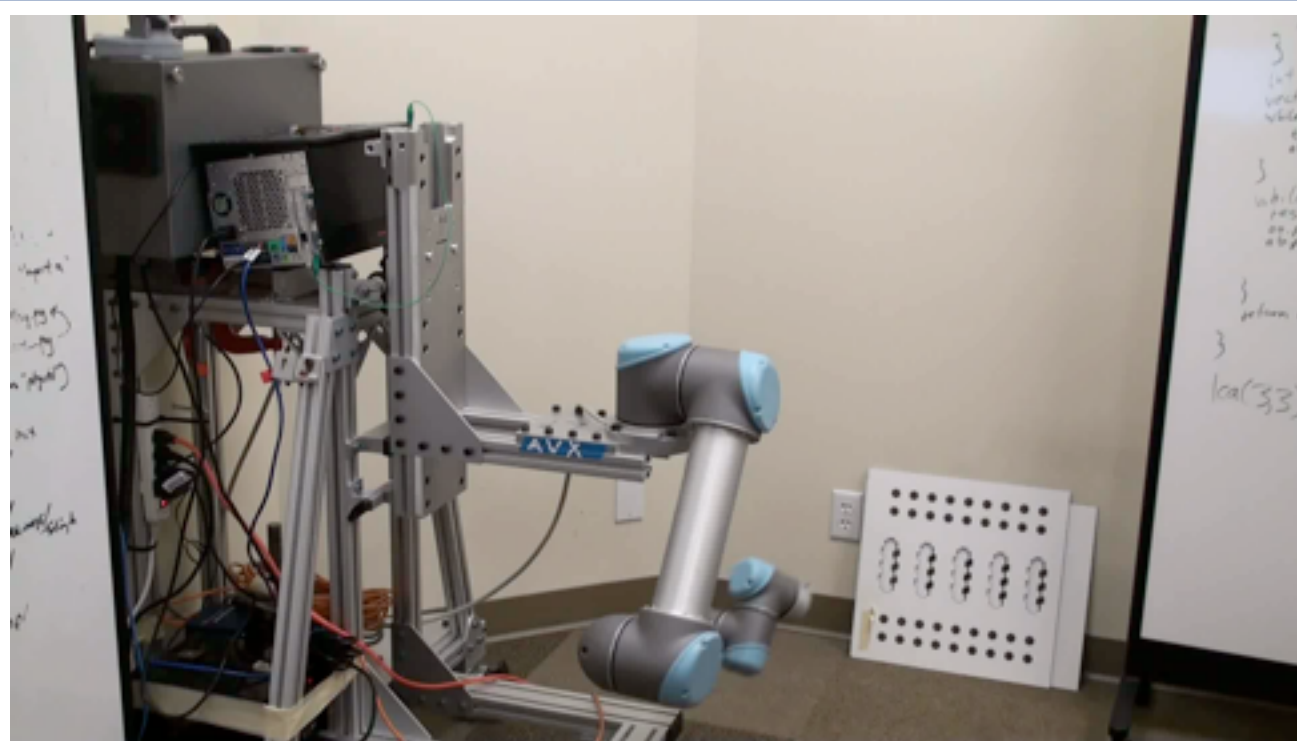
Robots Using Our Software



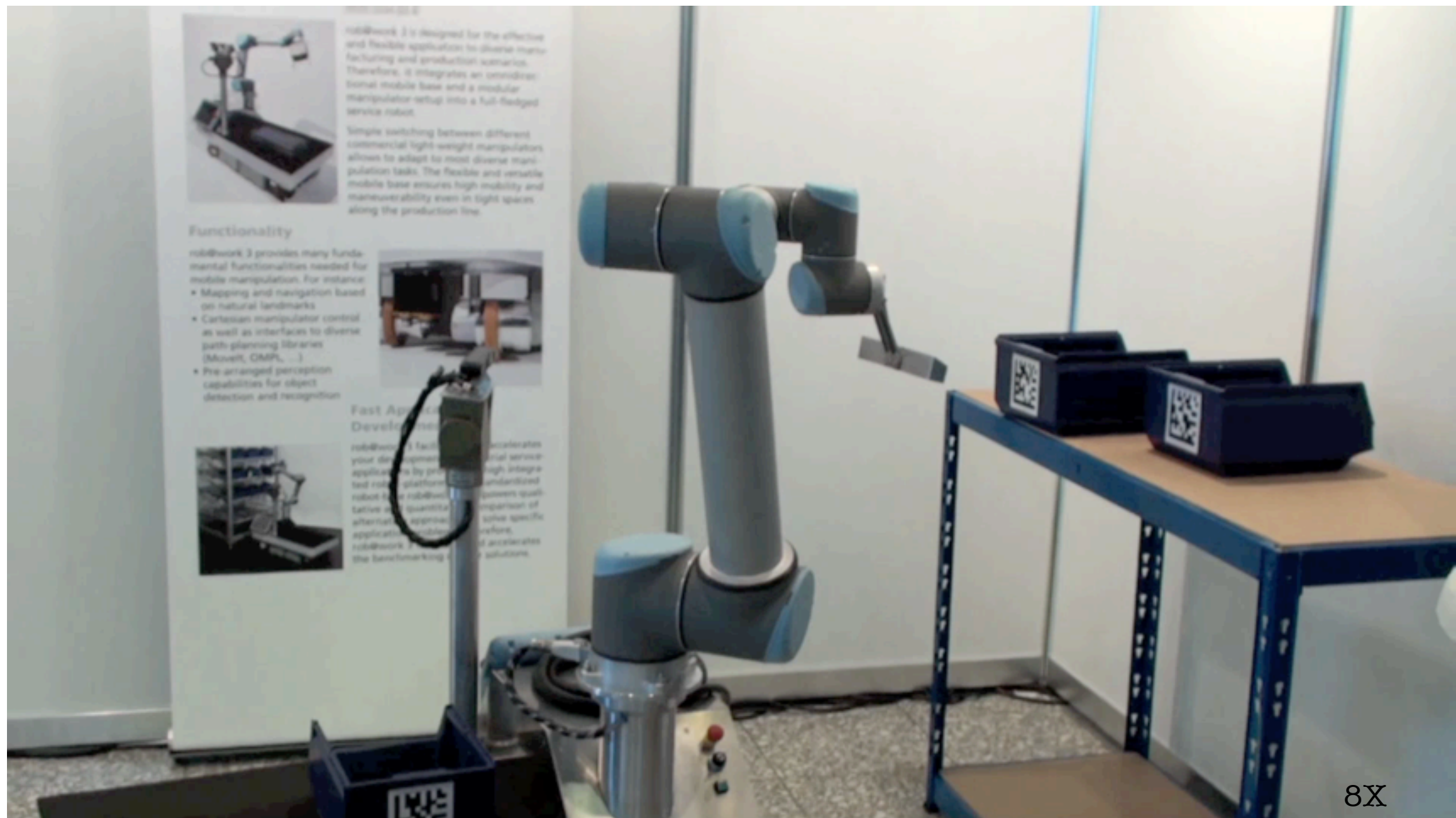
Application - ROS-Industrial

Same Software Running on Industrial Hardware

MoveIt!



Application - ROS-Industrial



MoveIt!

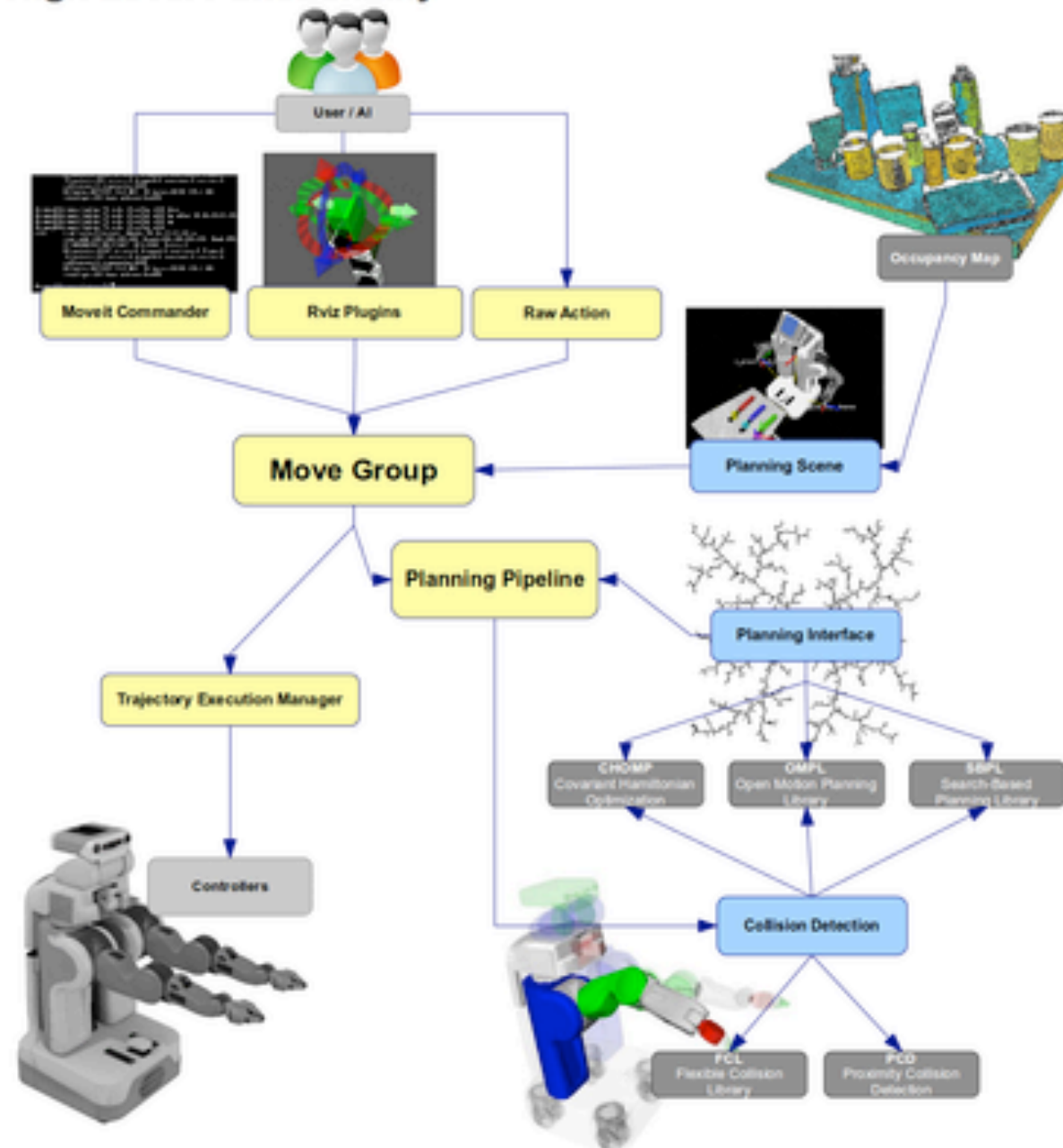
- Technical Capabilities

- ❖ Collision Checking: fast and flexible
- ❖ Integrated Kinematics
- ❖ Motion Planning
 - fast, good quality paths
 - kinematic constraints
- ❖ Integrated Perception for Environment Representation
- ❖ Standardized Interfaces to Controllers
- ❖ Execution and Monitoring
- ❖ Kinematic Analysis

MoveIt!

MoveIt – A Planning Framework High Level Functionality

24 Mar 2013

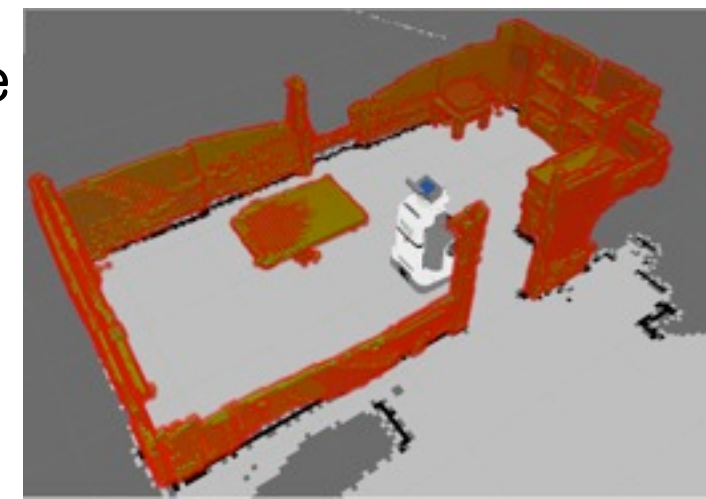


Collision Checking

- FCL - Flexible Collision Library*
 - ❖ parallelizable collision checking
 - ❖ Maximum about 2-3,000 full body collision checks for the PR2 per second
 - ✓ with realtime sensor data
 - ❖ + high fidelity mesh model
- Proximity Collision Detection



- ❖ Uses 3D distance transform to determine distance to nearest obstacle and gradient
- ❖ + very fast - 40 to 80,000 collision checks per second for the full body of the PR2
- ❖ - not as accurate

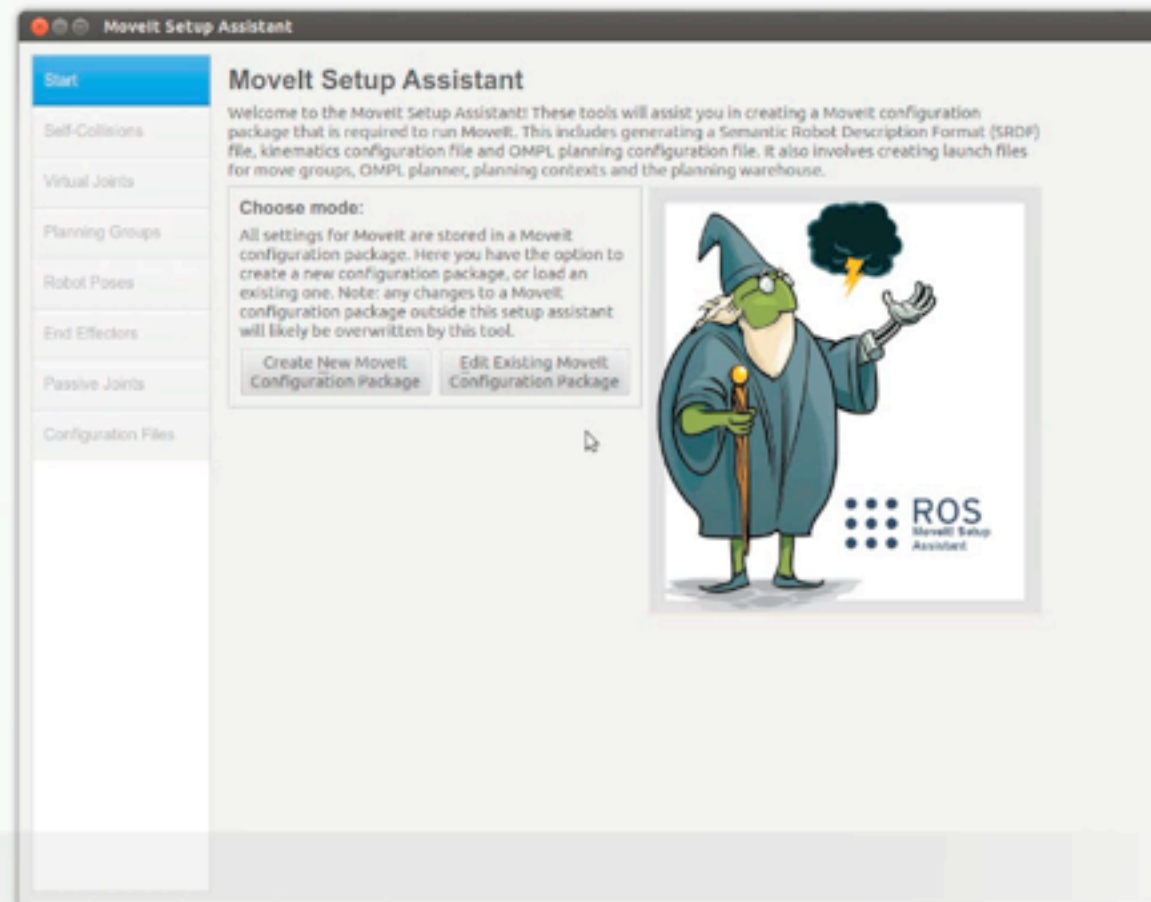


Motion Planning



- Plugin interface for planners
- Integration with robots through MoveIt!
- Automatically configured using the MoveIt! Setup Assistant
 - ❖ Sampling based planners (OMPL) [*http://ompl.kavrakilab.org](http://ompl.kavrakilab.org)
 - ❖ Search Based Planning Library (SBPL) [*http://www.ros.org/wiki/sbpl](http://www.ros.org/wiki/sbpl)

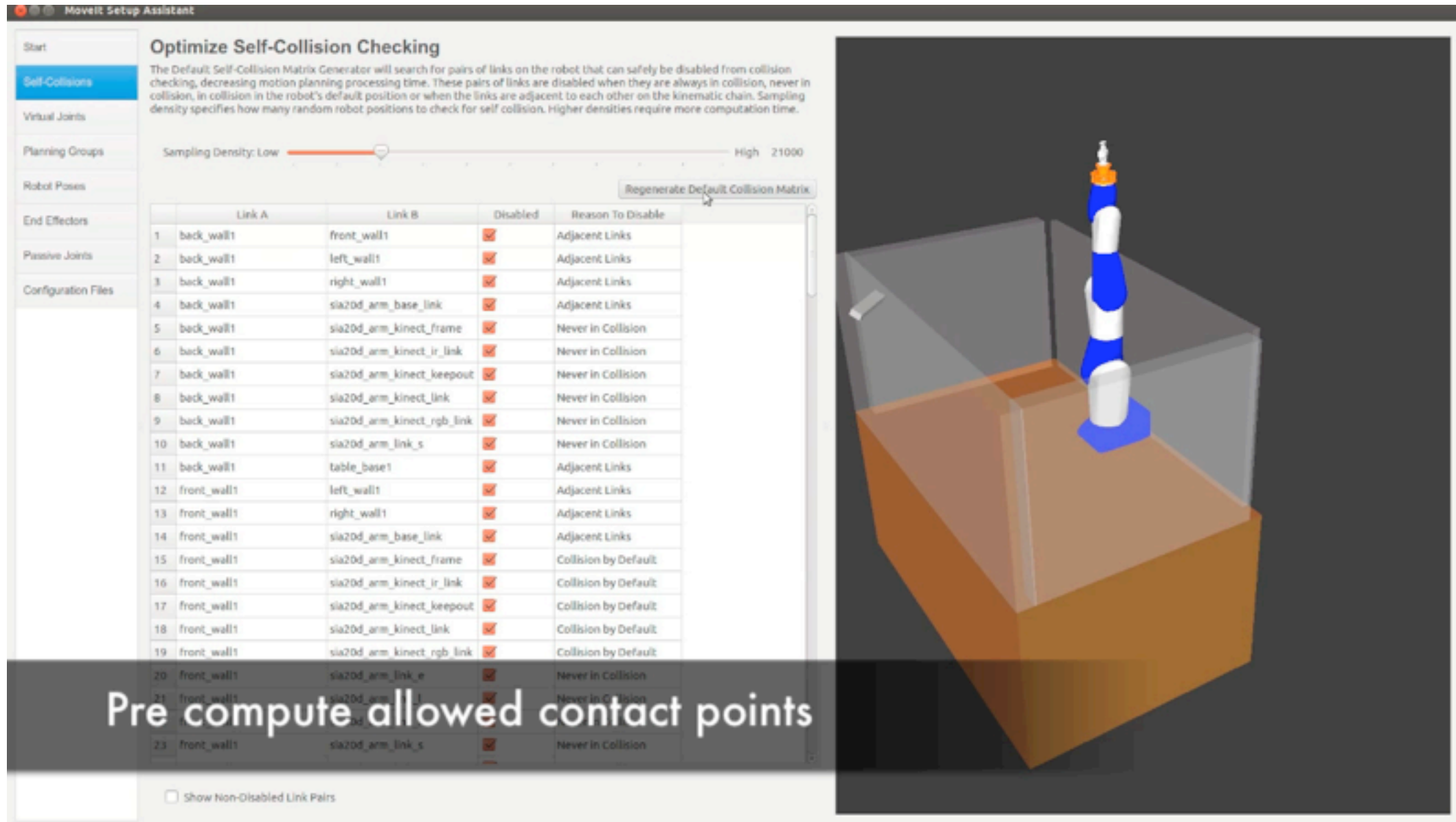
Easy Setup and Configuration



The MoveIt

We can create a new configuration package or edit an existing

Easy Setup and Configuration



Optimize Self-Collision Checking

The Default Self-Collision Matrix Generator will search for pairs of links on the robot that can safely be disabled from collision checking, decreasing motion planning processing time. These pairs of links 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. Higher densities require more computation time.

Sampling Density: Low High 21000

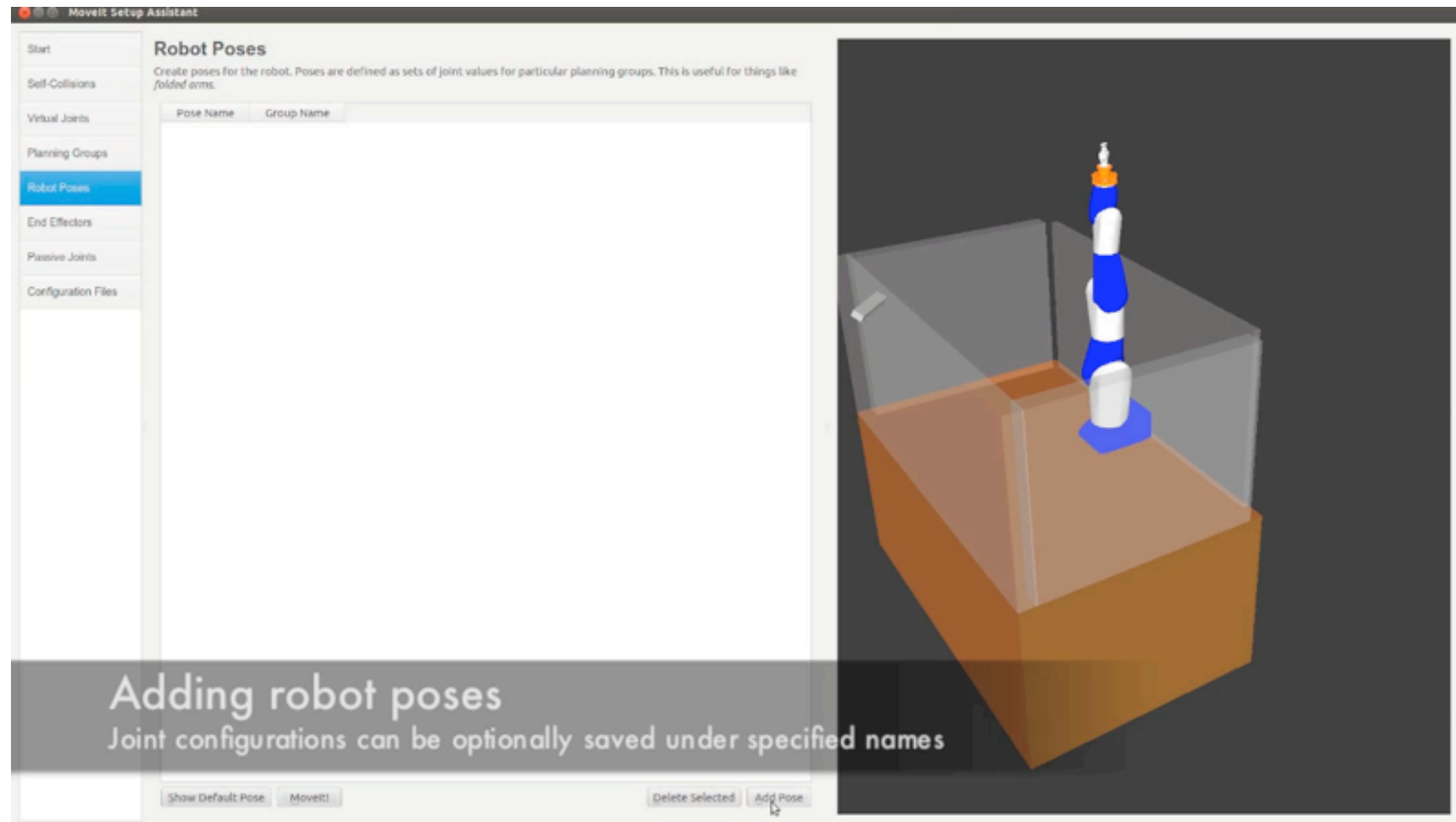
[Regenerate Default Collision Matrix](#)

	Link A	Link B	Disabled	Reason To Disable
1	back_wall1	front_wall1	<input checked="" type="checkbox"/>	Adjacent Links
2	back_wall1	left_wall1	<input checked="" type="checkbox"/>	Adjacent Links
3	back_wall1	right_wall1	<input checked="" type="checkbox"/>	Adjacent Links
4	back_wall1	sia20d_arm_base_link	<input checked="" type="checkbox"/>	Adjacent Links
5	back_wall1	sia20d_arm_kinect_frame	<input checked="" type="checkbox"/>	Never in Collision
6	back_wall1	sia20d_arm_kinect_ir_link	<input checked="" type="checkbox"/>	Never in Collision
7	back_wall1	sia20d_arm_kinect_keepout	<input checked="" type="checkbox"/>	Never in Collision
8	back_wall1	sia20d_arm_kinect_link	<input checked="" type="checkbox"/>	Never in Collision
9	back_wall1	sia20d_arm_kinect_rgb_link	<input checked="" type="checkbox"/>	Never in Collision
10	back_wall1	sia20d_arm_link_s	<input checked="" type="checkbox"/>	Never in Collision
11	back_wall1	table_base1	<input checked="" type="checkbox"/>	Adjacent Links
12	front_wall1	left_wall1	<input checked="" type="checkbox"/>	Adjacent Links
13	front_wall1	right_wall1	<input checked="" type="checkbox"/>	Adjacent Links
14	front_wall1	sia20d_arm_base_link	<input checked="" type="checkbox"/>	Adjacent Links
15	front_wall1	sia20d_arm_kinect_frame	<input checked="" type="checkbox"/>	Collision by Default
16	front_wall1	sia20d_arm_kinect_ir_link	<input checked="" type="checkbox"/>	Collision by Default
17	front_wall1	sia20d_arm_kinect_keepout	<input checked="" type="checkbox"/>	Collision by Default
18	front_wall1	sia20d_arm_kinect_link	<input checked="" type="checkbox"/>	Collision by Default
19	front_wall1	sia20d_arm_kinect_rgb_link	<input checked="" type="checkbox"/>	Collision by Default
20	front_wall1	sia20d_arm_link_e	<input checked="" type="checkbox"/>	Never in Collision
21	front_wall1	sia20d_arm_link_i	<input checked="" type="checkbox"/>	Never in Collision
22	front_wall1	sia20d_arm_link_j	<input checked="" type="checkbox"/>	Never in Collision
23	front_wall1	sia20d_arm_link_s	<input checked="" type="checkbox"/>	Never in Collision

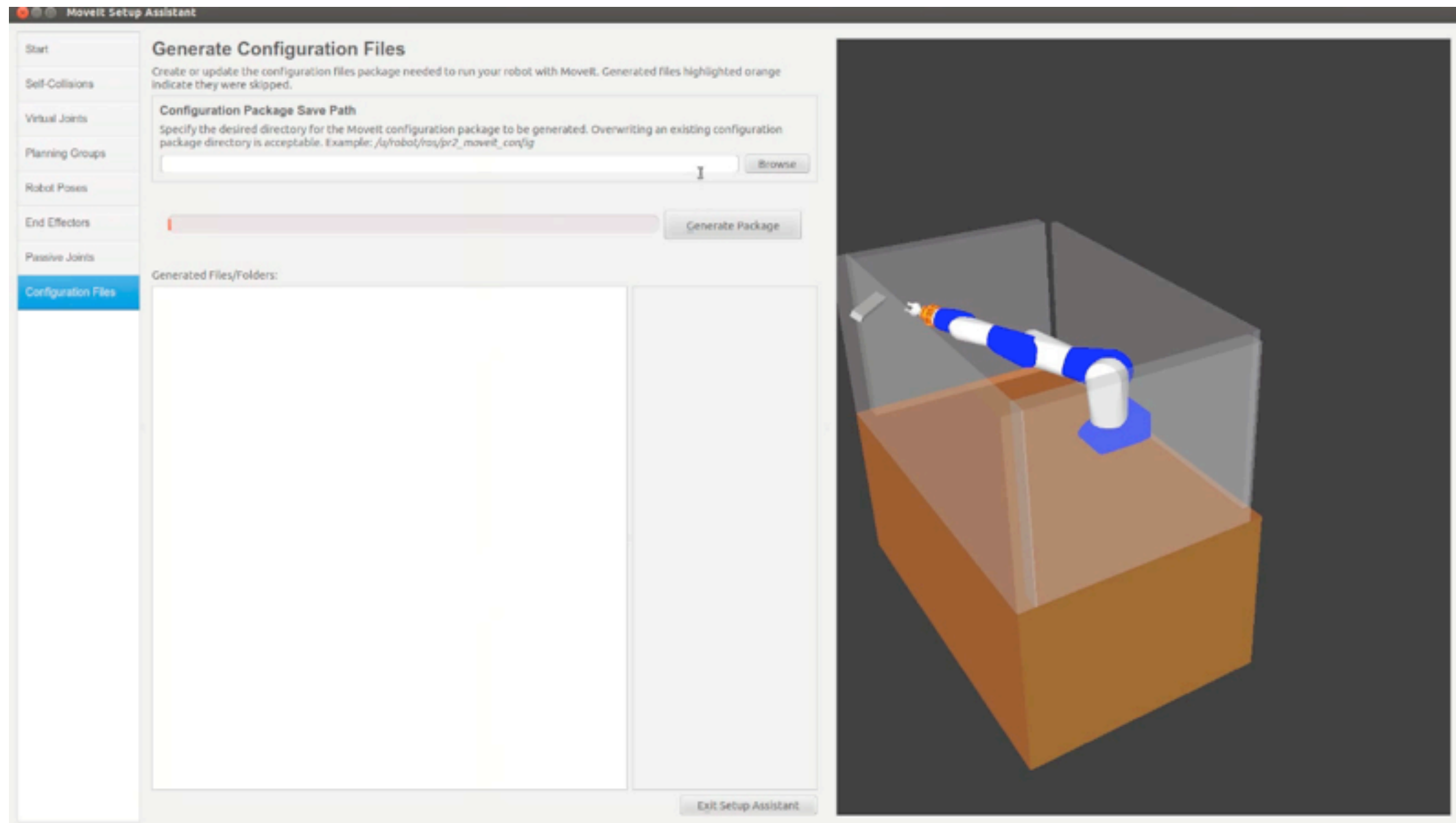
☐ Show Non-Disabled Link Pairs

Pre compute allowed contact points

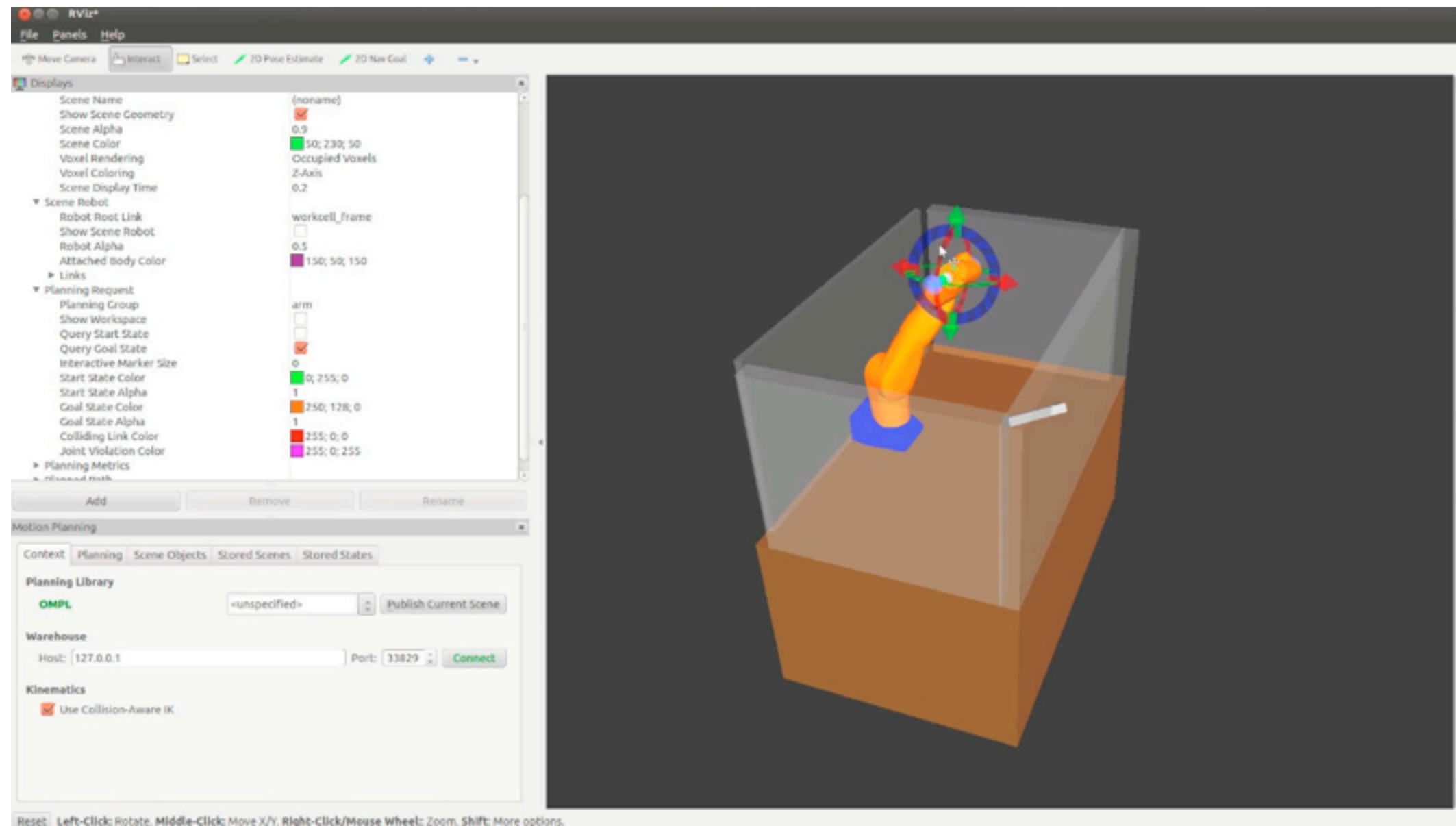
Easy Setup and Configuration



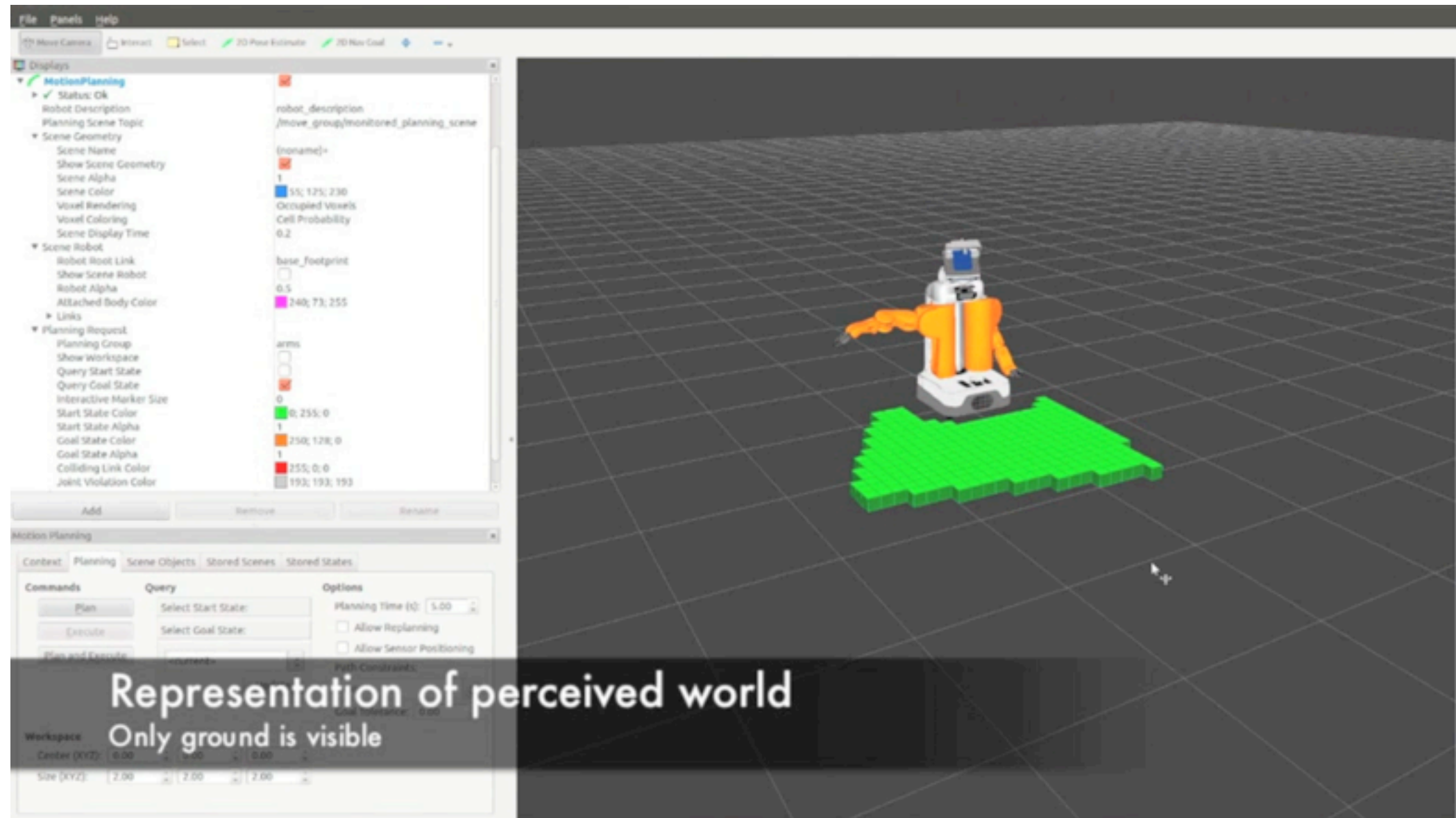
Easy Setup and Configuration



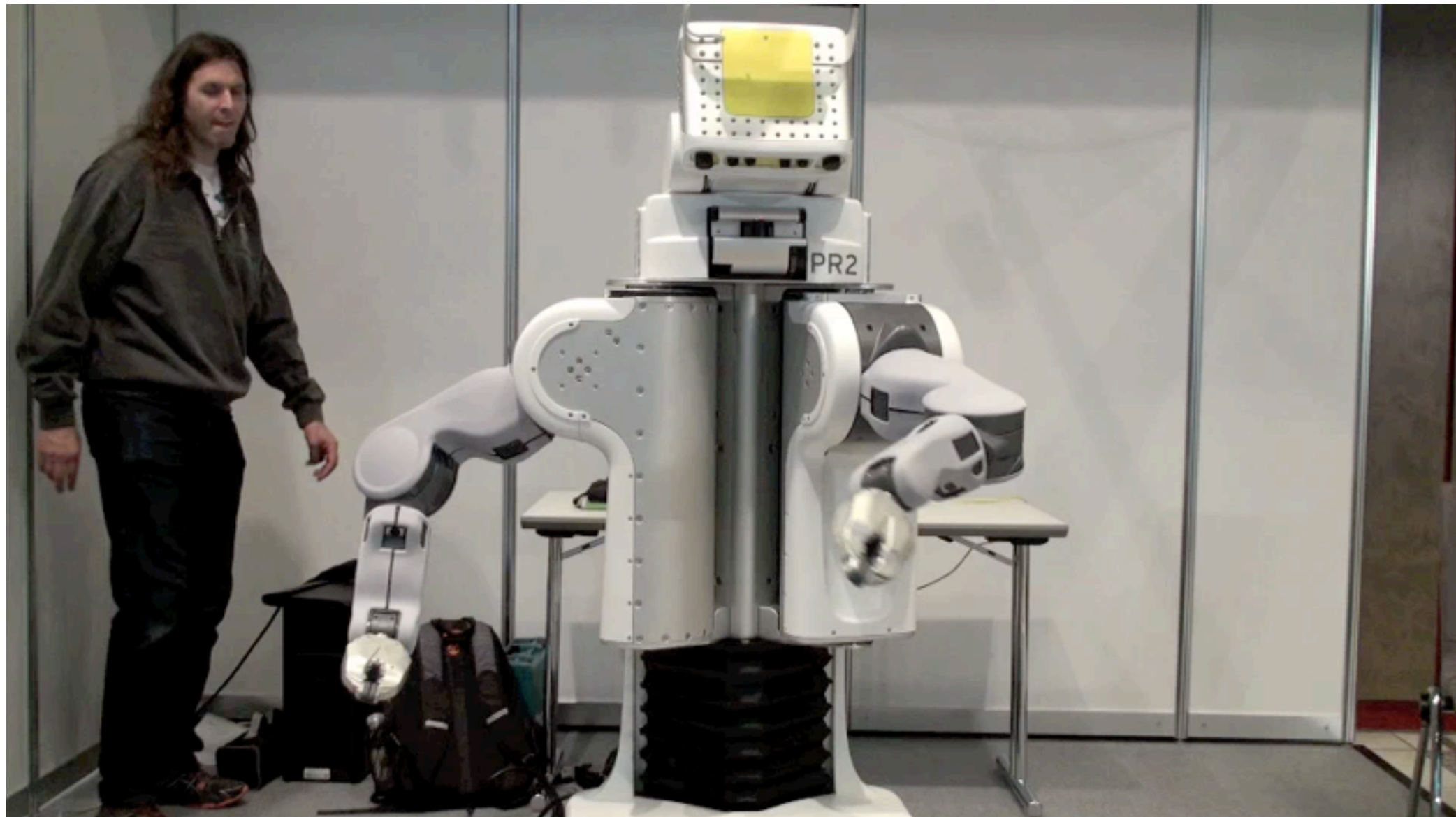
Easy Setup and Configuration



Integrating Perception

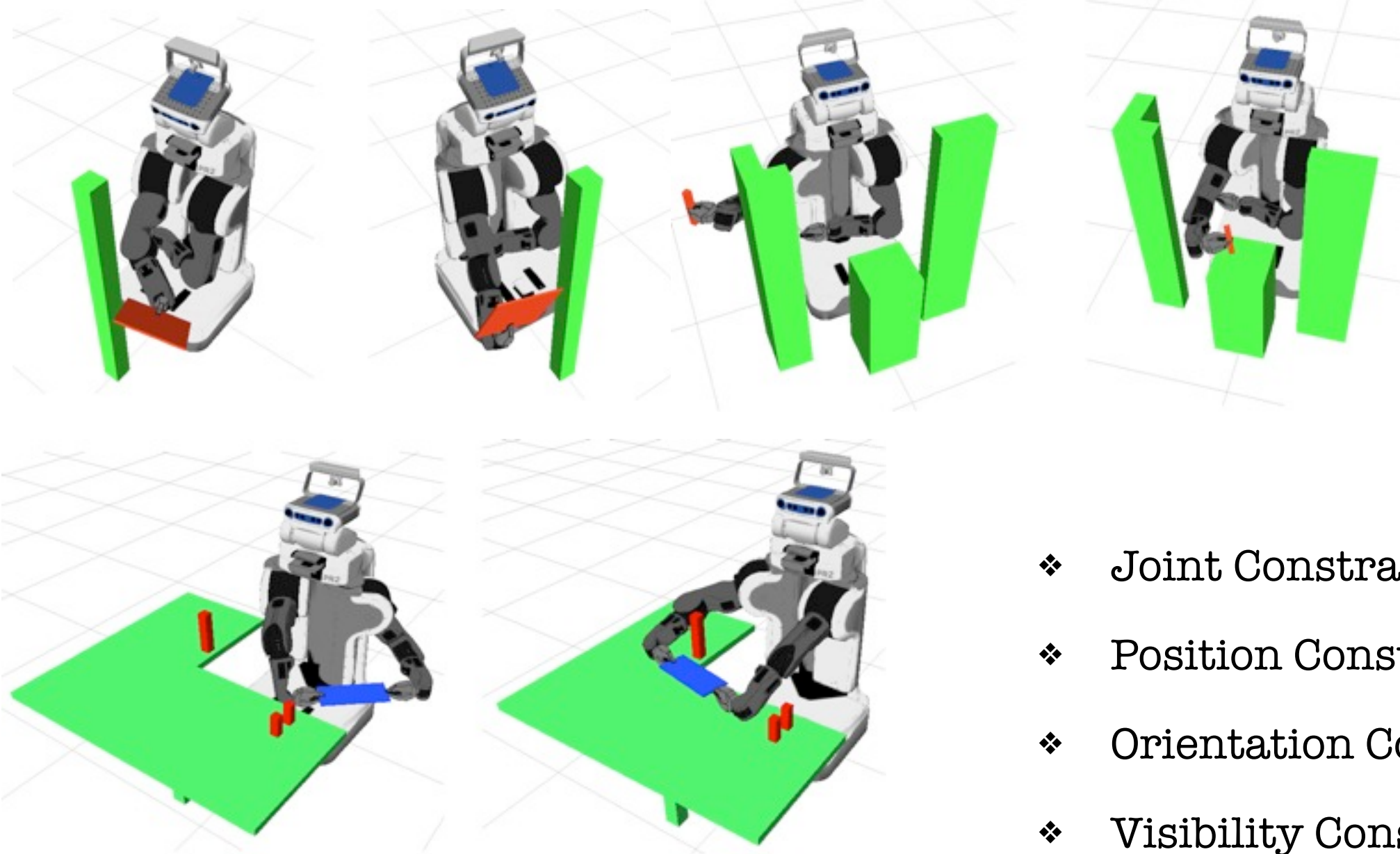


Reactive Motion



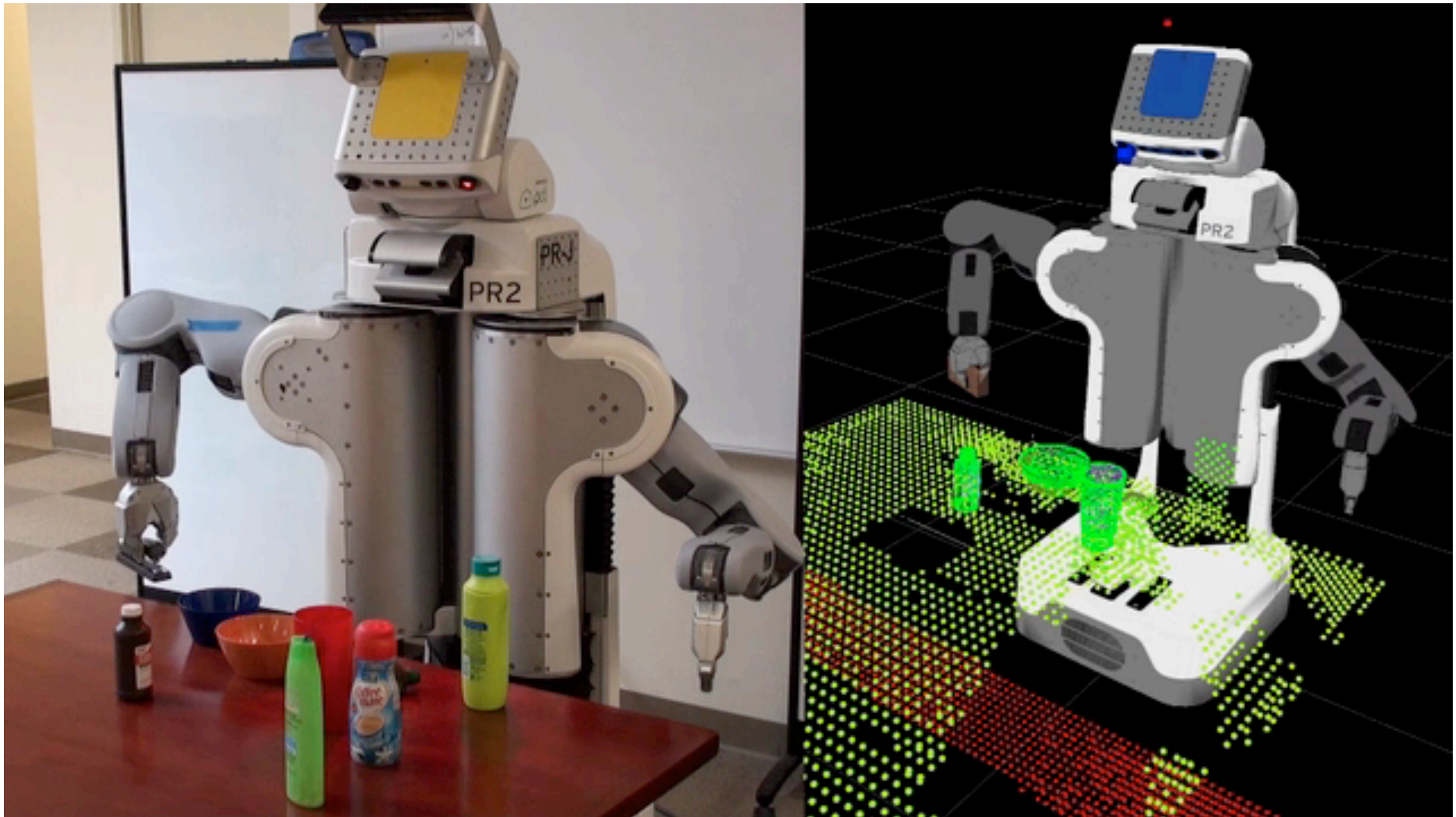
* Live Demo later!!

Constraint Representation

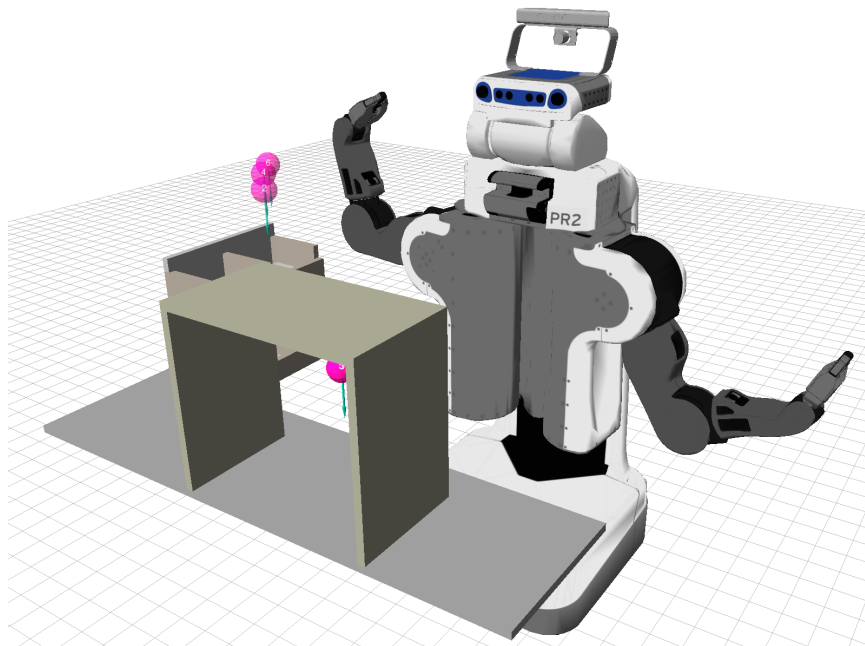


- ❖ Joint Constraints
- ❖ Position Constraints
- ❖ Orientation Constraints
- ❖ Visibility Constraints

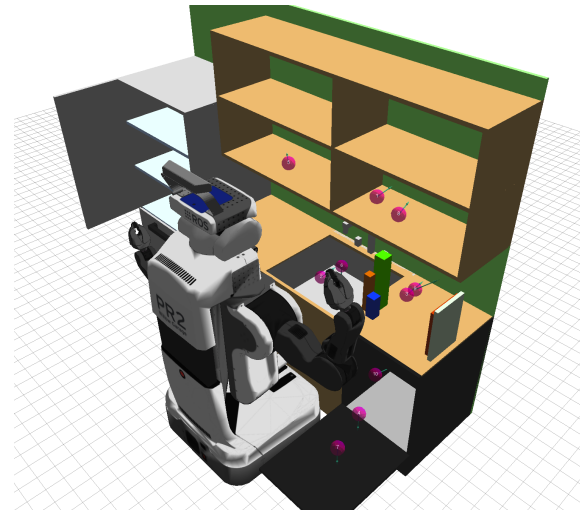
Orientation Constraints



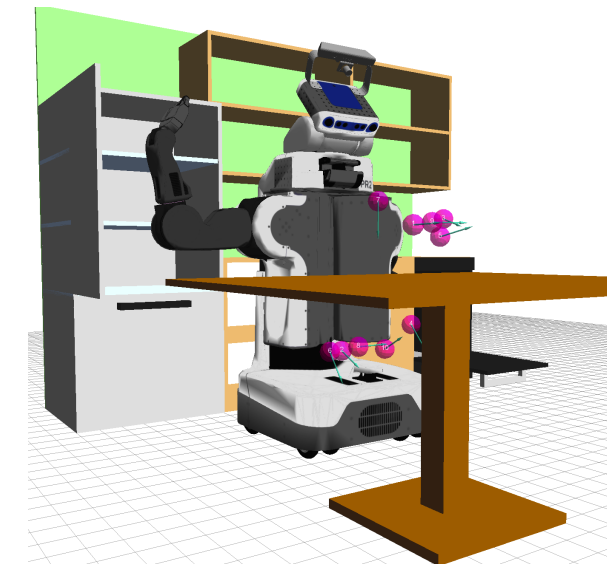
Applications: Benchmarking



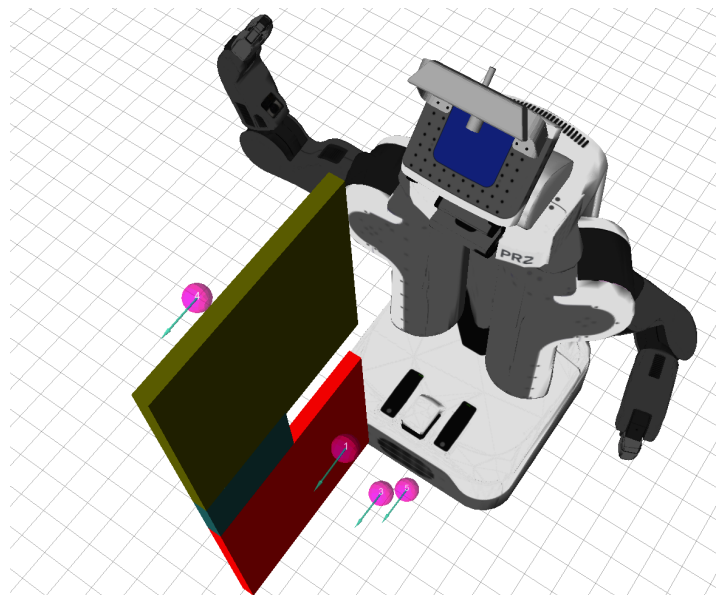
- Industrial Environment



- Kitchen Environment

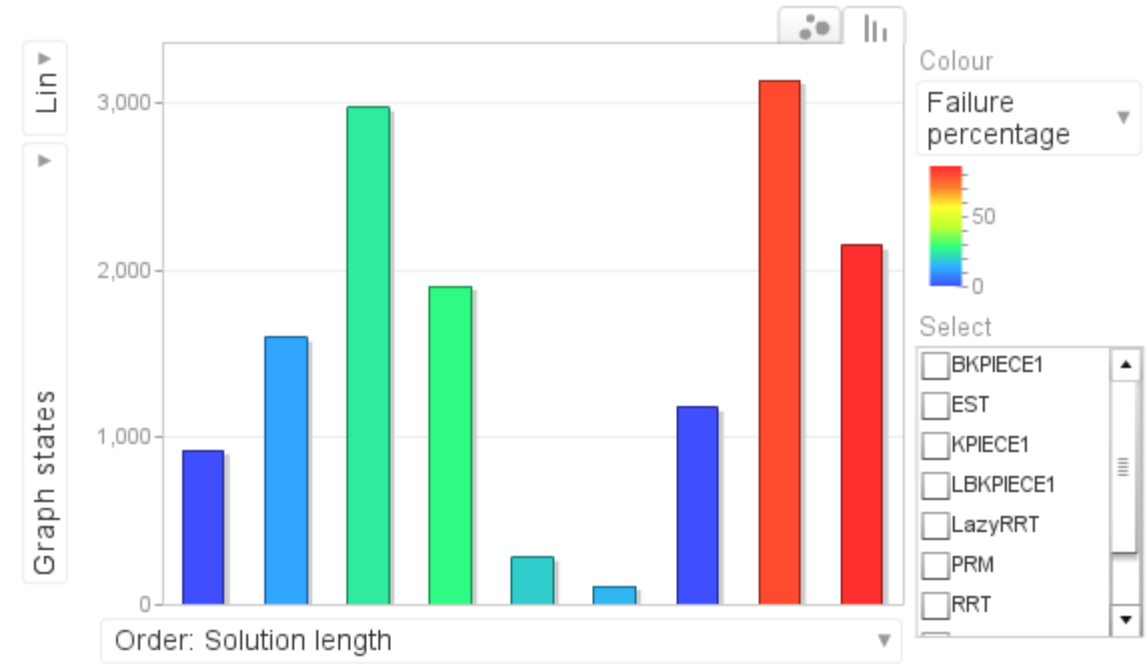
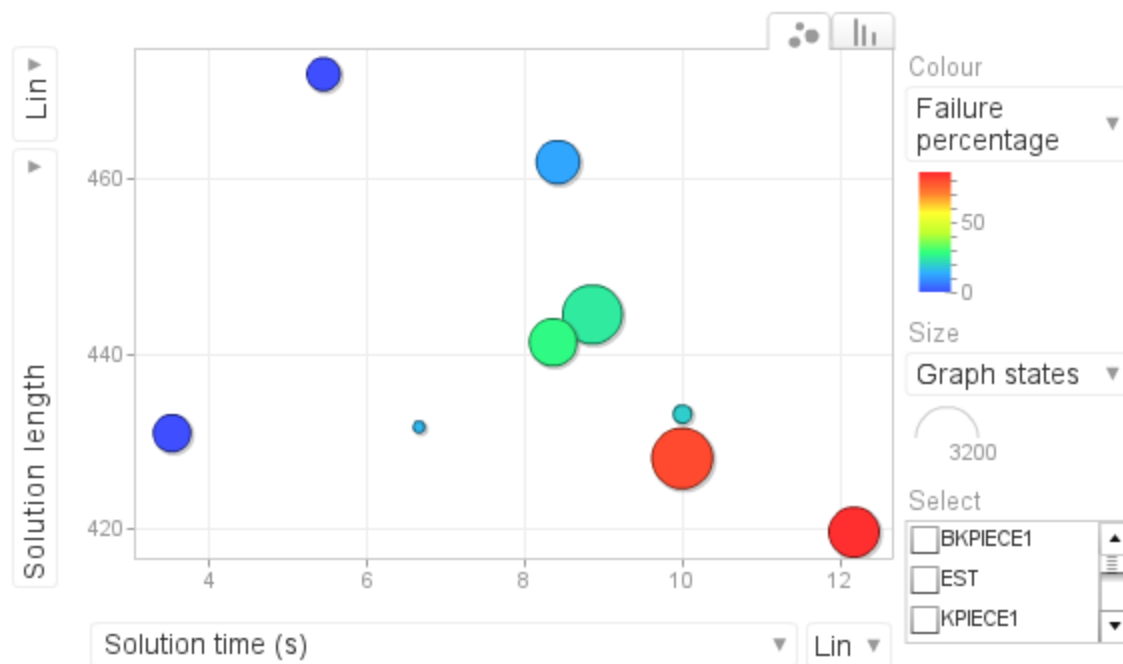


- Tabletop Environment



- Narrow Passage Environment

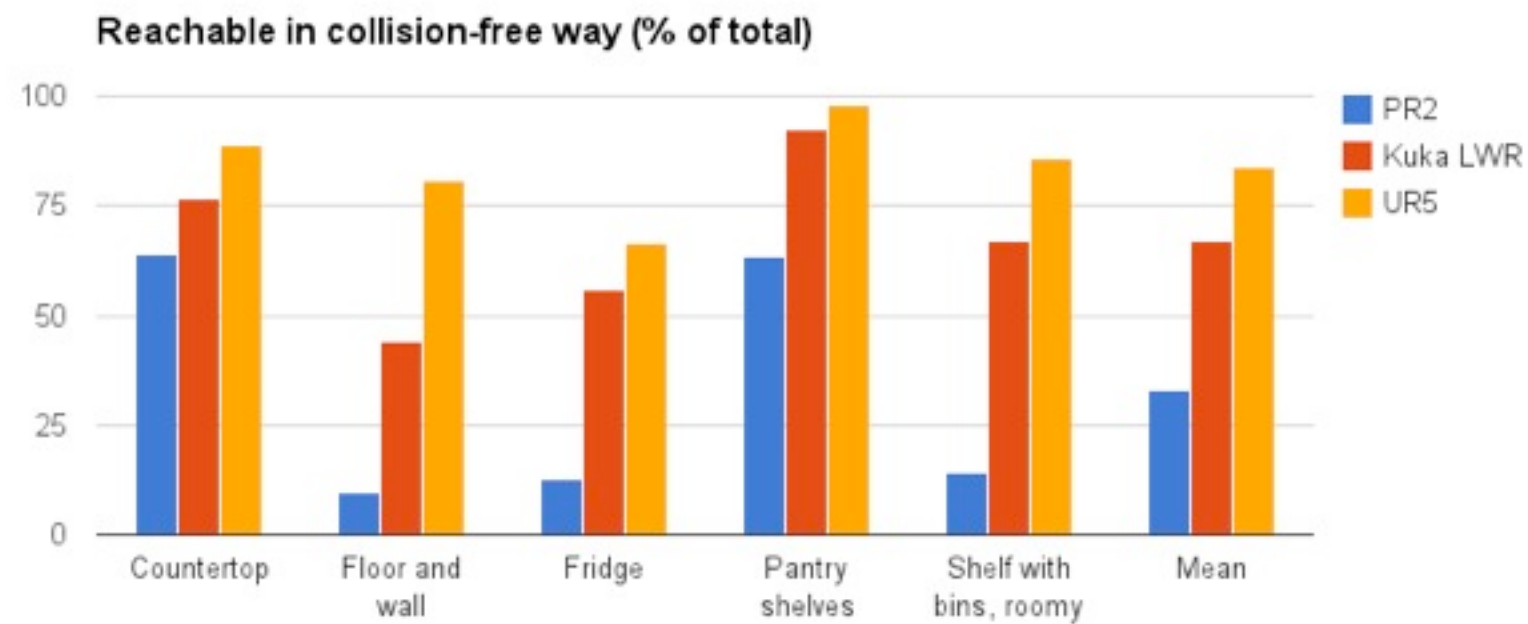
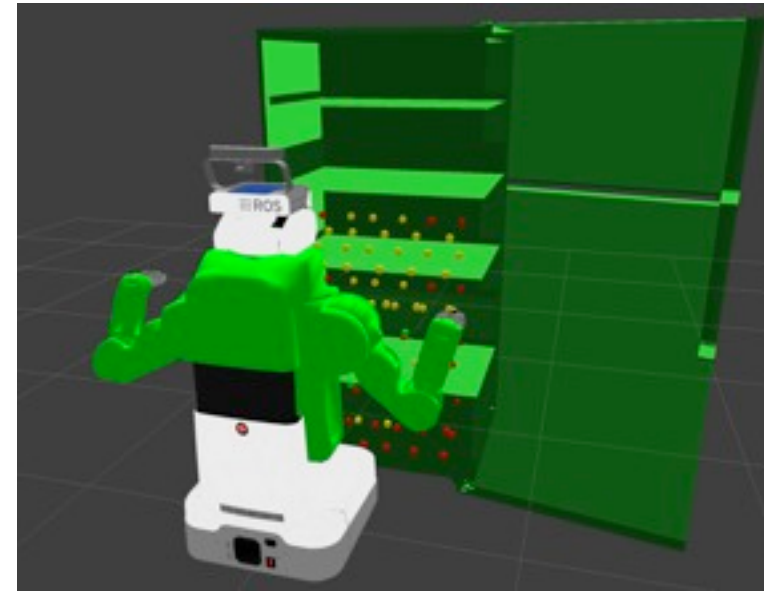
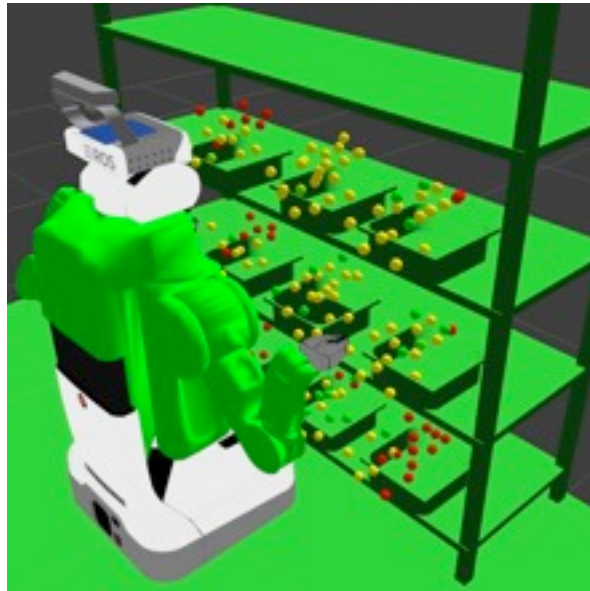
Benchmarking



Applications: Kinematic Workspace Analysis

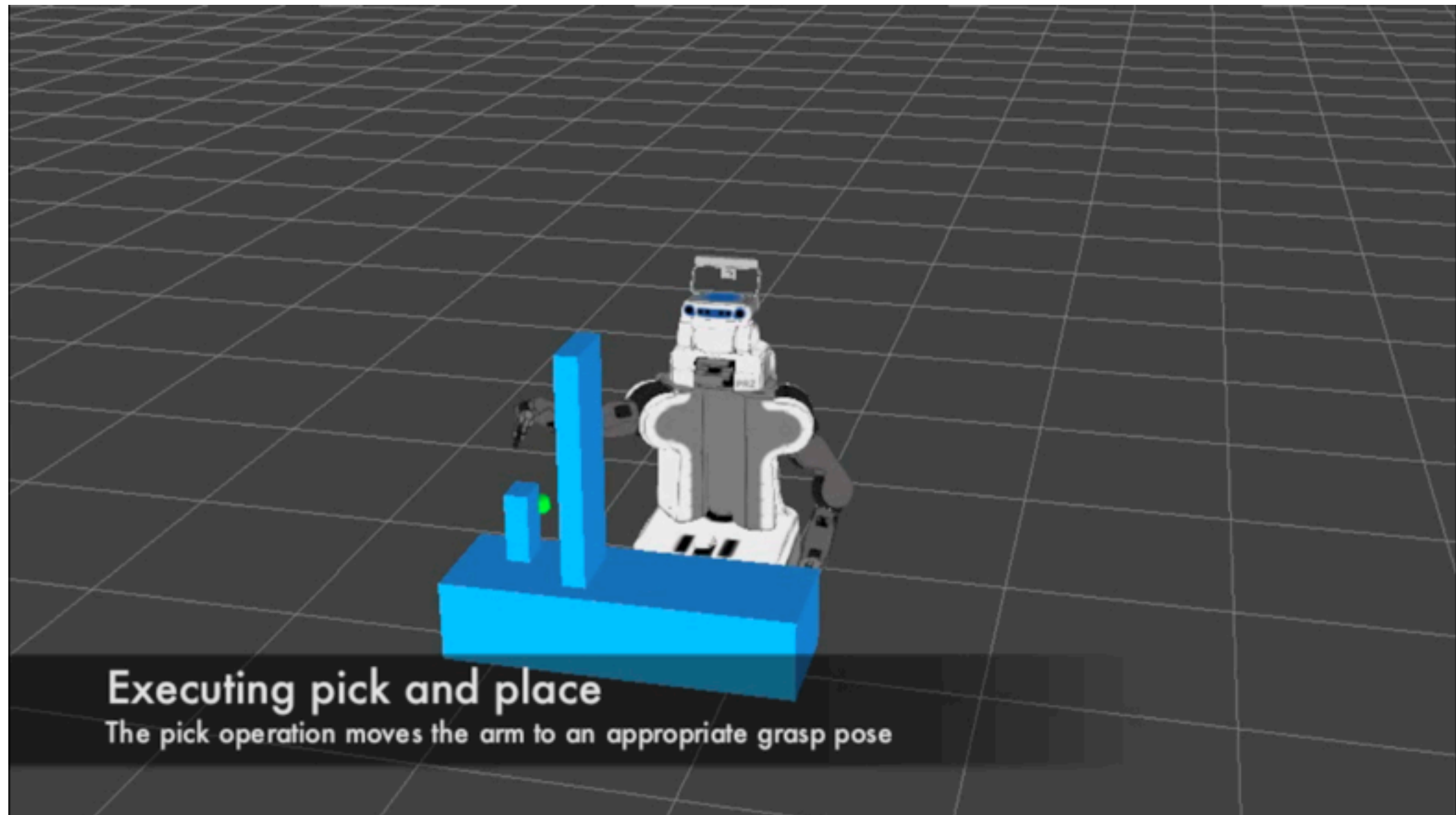
- Robot Design Evaluation
- Robot Workspace Placement

Kinematic Workspace Analysis



MoveIt!

- Applications - Pick and Place
 - ❖ Integrated Grasping, Planning, Perception and Execution



User API

- Really simple API (e.g. moving an arm):

```
move_group_interface::MoveGroup group( "arm" );  
group.setRandomTarget( );  
group.move( );
```

MoveIt!

- New Architecture (different from arm navigation)
 - ✓ Minimize transport and messaging overhead - Single process for planning and perception, shares environment representation (planning scene) vs. multiple ROS nodes each performing individual functions
 - ❖ Computation - Core capabilities (e.g. motion planning, kinematics, etc.) are setup in C++ libraries
 - ❖ Communication and Configuration through ROS
 - ❖ Emphasis on speed and efficiency – parallelize collision checking, kinematics, etc.

MoveIt!

- Capabilities (differences to arm navigation)
 - ❖ Collision Checking
 - ✓ Parallelizable
 - ✓ can switch between different types of collision checkers
 - ✓ cleaner C++ interface
 - Motion Planning
 - ✓ plugin based C++ interface (in addition to ROS interface)
 - ✓ Parallelizable
 - ✓ planning pipeline includes trajectory smoothing

Highlights

- Technical
 - Performance
 - ❖ Single process sharing environment representation
 - ❖ Parallelizable collision checking and kinematics
 - ❖ Parallelizable pick and place (upcoming capability)
 - Integrated Perception for Environment Representation
 - ❖ Can incorporate any source of point clouds
 - ❖ Fast self-filtering and environment representation
 - Reactive Motion Planning
 - ❖ Safer operation in collaborative environments

Highlights

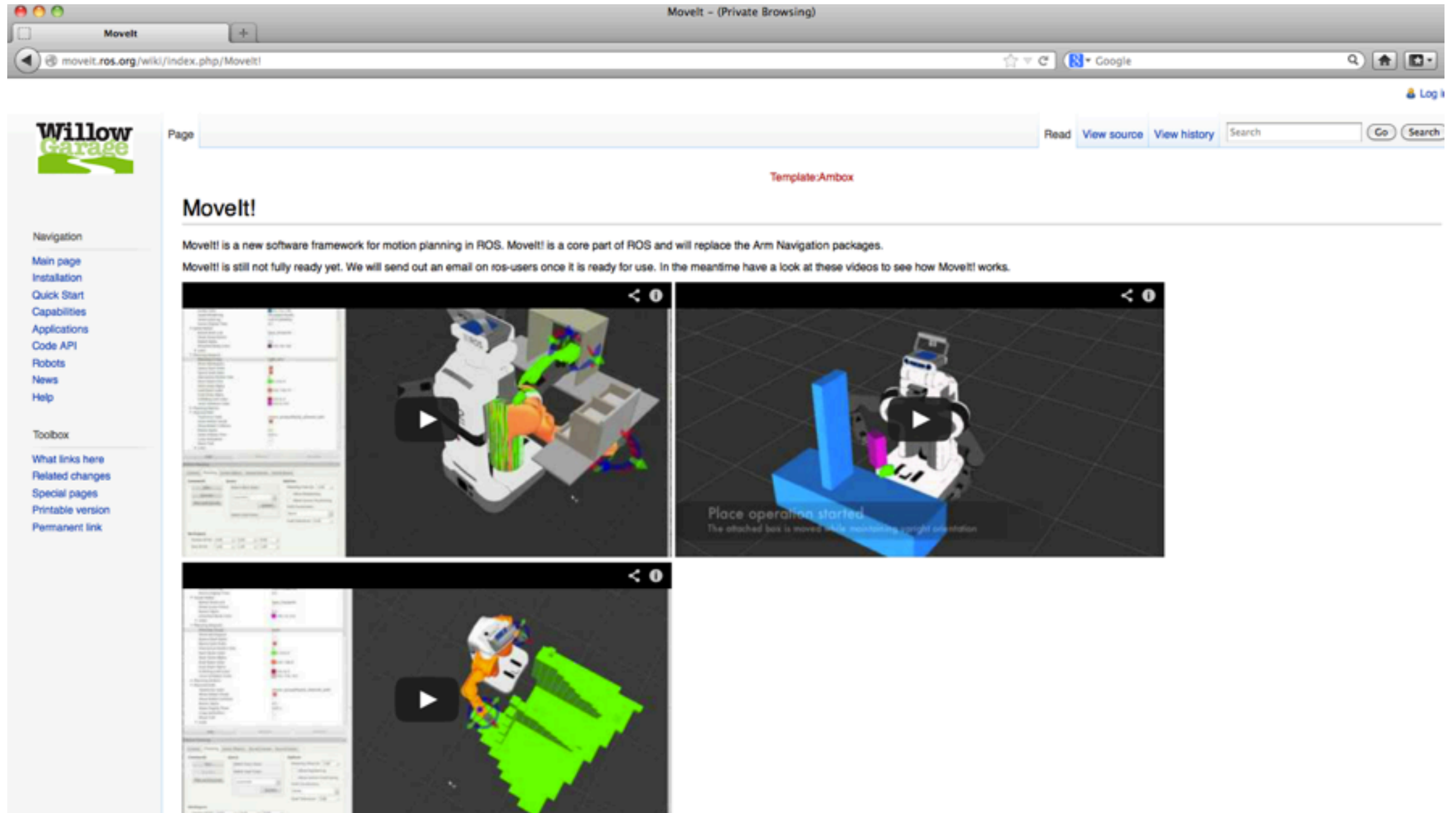
- User Friendly

- ❖ Easy configuration for new robots
- ❖ Graphical User Interfaces
- ❖ Better Visualization and Introspection
- ❖ Easy to use C++ API
- ❖ Python bindings

Highlights

- Integrated Applications
 - ❖ Collision-free Motion Planning and Execution
 - ❖ Kinematic Analysis/ Reachability Analysis
 - ❖ Benchmarking
- More applications in development ...
 - ❖ Pick and Place - more about this in afternoon session

Documentation - Wiki



The screenshot shows a web browser window titled "MoveIt! - (Private Browsing)". The address bar displays "moveit.ros.org/wiki/index.php/MoveIt!". The page features the Willow Garage logo in the top left corner. A navigation sidebar on the left includes links for "Main page", "Installation", "Quick Start", "Capabilities", "Applications", "Code API", "Robots", "News", and "Help". Below this is a "Toolbox" section with links for "What links here", "Related changes", "Special pages", "Printable version", and "Permanent link". The main content area is titled "MoveIt!" and contains the following text:

MoveIt! is a new software framework for motion planning in ROS. MoveIt! is a core part of ROS and will replace the Arm Navigation packages.

MoveIt! is still not fully ready yet. We will send out an email on ros-users once it is ready for use. In the meantime have a look at these videos to see how MoveIt! works.

Below the text are three video thumbnails. The top-left video shows a 3D simulation of a robotic arm with a green trajectory. The top-right video shows a 3D simulation of a robotic arm with a blue trajectory and a text overlay: "Place operation started. The attached box is moved while maintaining upright orientation." The bottom video shows a 3D simulation of a robotic arm with a green trajectory.

Github Repository

The screenshot shows the GitHub web interface for the repository `ros-planning/moveit_ros`. The browser address bar shows the URL `https://github.com/ros-planning/moveit_ros`. The repository is public and has 5 stars and 13 forks. The main navigation tabs include Code, Network, Pull Requests (0), Issues (34), Wiki, Graphs, and Settings. The 'Code' tab is selected, showing the repository's description: 'MoveIt! ROS — Read more'. Below the description are buttons for cloning the repository: Clone in Mac, ZIP, HTTP, SSH, and Git Read-Only. The SSH URL is `git@github.com:ros-planning/moveit_ros.git`. The current branch is `groovy-devel`. The 'Files' tab is selected, showing a list of files and folders. The latest commit is `a882772093` by loan Sucan, authored 22 minutes ago. The commit message is 'fix error reporting for trajectory execution when ensureActiveControl...'. The list of files includes `benchmarks`, `interface`, `manipulation`, `move_group`, `moveit_ros`, `perception`, `planning`, `visualization`, `warehouse`, `.gitignore`, and `README.md`.

File	Time	Commit Message
fix error reporting for trajectory execution when ensureActiveControl...	22 minutes ago	fix error reporting for trajectory execution when ensureActiveControl... [loan Sucan]
visualization	24 minutes ago	disable keeping of attached objects for query states when scene state... [loan Sucan]
planning	22 minutes ago	fix error reporting for trajectory execution when ensureActiveControl... [loan Sucan]
perception	a day ago	bugfixes and optimizations for processing depth images [loan Sucan]
moveit_ros	5 days ago	0.3.22 [loan Sucan]
warehouse	5 days ago	0.3.22 [loan Sucan]
.gitignore	8 days ago	ignore emacs temp files [davecoleman]
README.md	3 months ago	added readme [loan Sucan]

Issue Tracking

The screenshot shows the GitHub interface for the `ros-planning/moveit_ros` repository. The browser address bar shows the URL `https://github.com/ros-planning/moveit_ros/issues?state=open`. The repository is public and has 34 issues, 5 stars, and 13 forks. The 'Issues' tab is selected, showing 34 open issues and 176 closed issues. The left sidebar lists filters for 'Everyone's Issues' (34), 'Assigned to you' (5), 'Created by you' (1), and 'Mentioning you' (1). Below these are labels: 'bug' (1), 'enhancement' (8), 'nice to have' (2), 'CRITICAL' (0), 'MAJOR' (0), 'MINOR' (0), 'duplicate' (0), 'invalid' (0), 'question' (0), and 'won't fix' (0). The main area displays a list of 34 open issues, including:

- #210: modify speed of trajectory execution (a scale) at runtime (by isucan 2 days ago)
- #207: Package naming inconsistency (enhancement) (by davetcoleman 19 days ago)
- #198: rviz dies when disabling plugin / re-enabling (bug) (by davetcoleman a month ago, 2 comments)
- #193: move eef offset computation from benchmark core to benchmark gui (by isucan a month ago)
- #192: split run_benchmark into lightweight node + library (by isucan a month ago)
- #191: Make benchmark service more general (by marioprats a month ago, 5 comments)
- #181: split planning/ into planning/ and common/ (by isucan a month ago)
- #180: default location for loading ros params (by isucan a month ago)
- #171: Add double-buffering to occupancy map updates (enhancement) (by aleeper 2 months ago)
- #168: Add ability to initialize using static octomap to occupancy_map_monitor (enhancement) (by sachinchitta 2 months ago)
- #166: Add ability to specify variable offsets to the control frame for end-effector markers (by aleeper 2 months ago)
- #162: names for background jobs in rviz plugin

Community

- moveit-users@googlegroups.com - questions related to how you can use MoveIt!

Where are we going?

- MoveIt!
 - ✓ what does it take to use MoveIt! in products
 - ✓ interest in enterprise level, supported versions of MoveIt! and associated capabilities in ROS?

More information



Email: moveit@willowgarage.com