MoveIt!

Strengths, Weaknesses, and Developer Insights

ROSCon Hamburg 2015

Dave Coleman
dave@dav.ee

Overview

Establish Credibility 1min

Background of MoveIt! 5min

What it's done well 5min

Typical use patterns 5min

Demystifying complexity 5min

Amazon Picking Challenge 5min

Where MoveIt! needs improvement 4min

Future Roadmap 10min

Q&A 5min

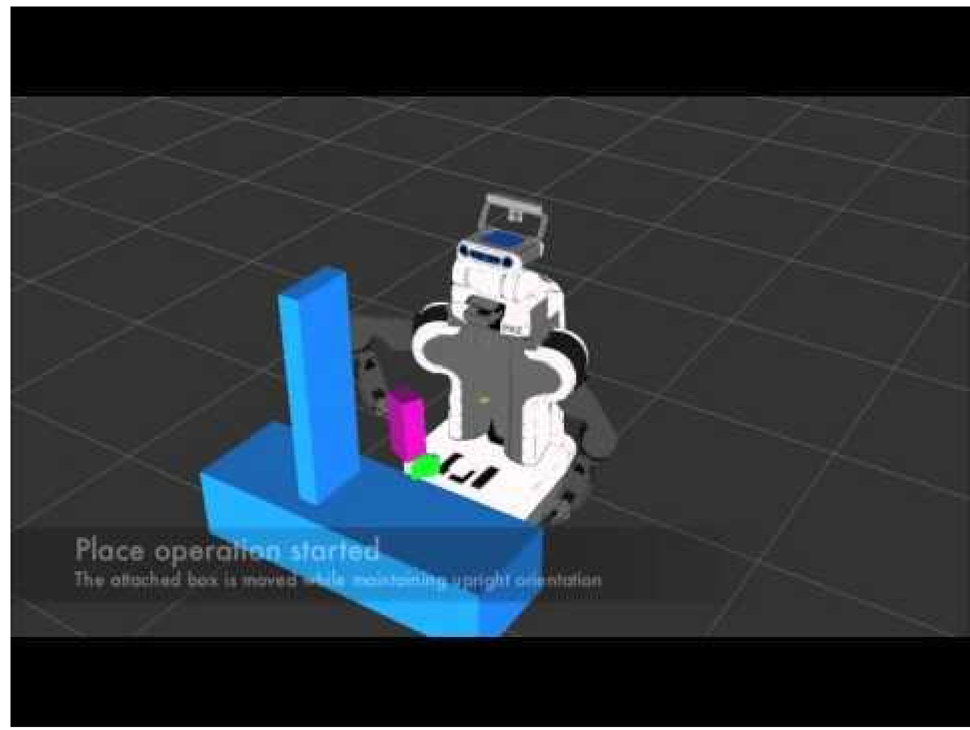
Establish Credibility 1min

- PhD Student At CU Boulder with Nikolaus Correll
- Interned with E. Gil Jones & Ioan Sucan at Willow Garage
  - Created Setup Assistant
- Have used and contributed to MoveIt! since before it was released
- Am a MoveIt! maintainer
- Have contributed to OMPL and many other ROS packages

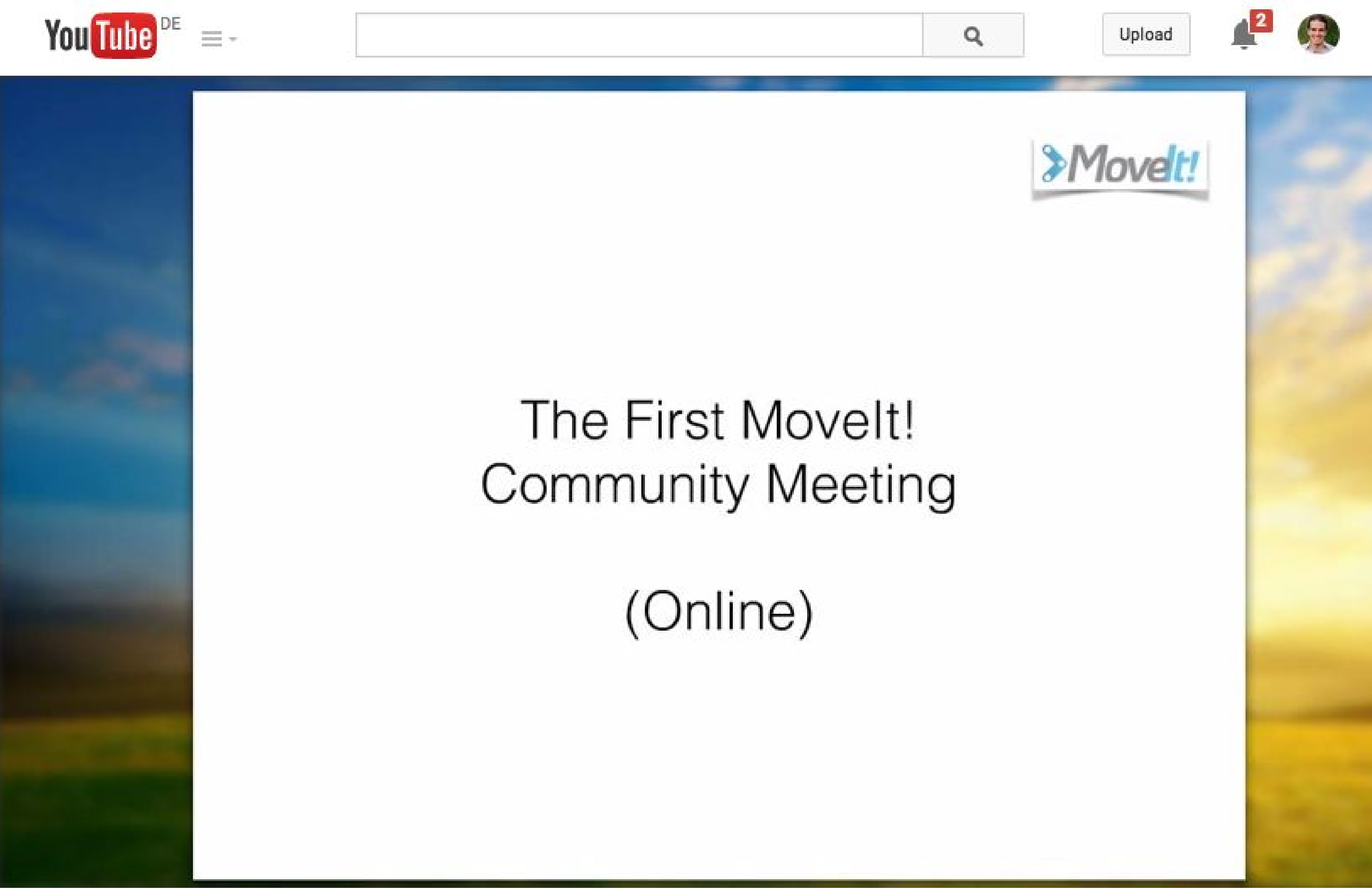
Background of MoveIt! 5min

Easy to use framework for motion planning, manipulation, 3D perception, kinematics, control and navigation

- Created at Willow Garage by Ioan Sucan, Sachin Chitta, many others
- Collaboration between many organizations
- Predecessor: arm_navigation announced in March 2010
- 31 contributors to moveit_core
- Written in C++ with Python bindings
- https://github.com/ros-planning/



1:00

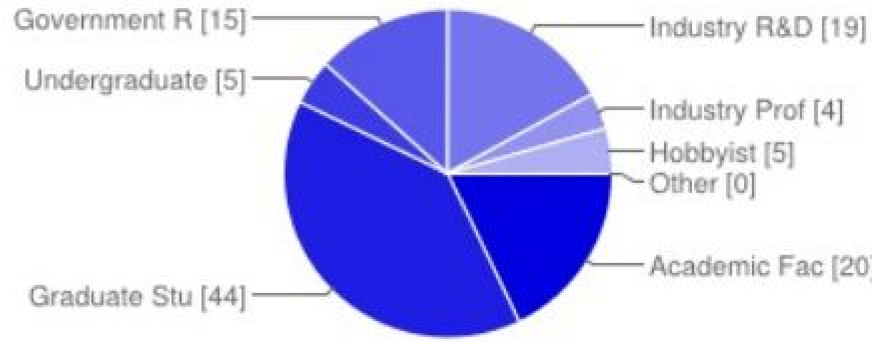


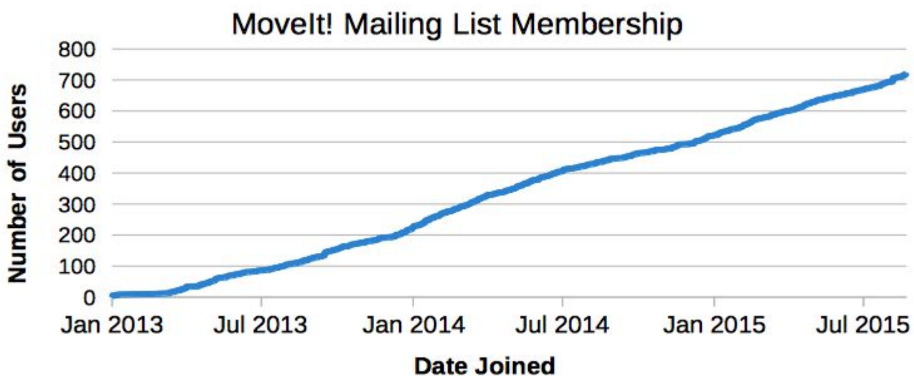
240 people registered, 150 who attended

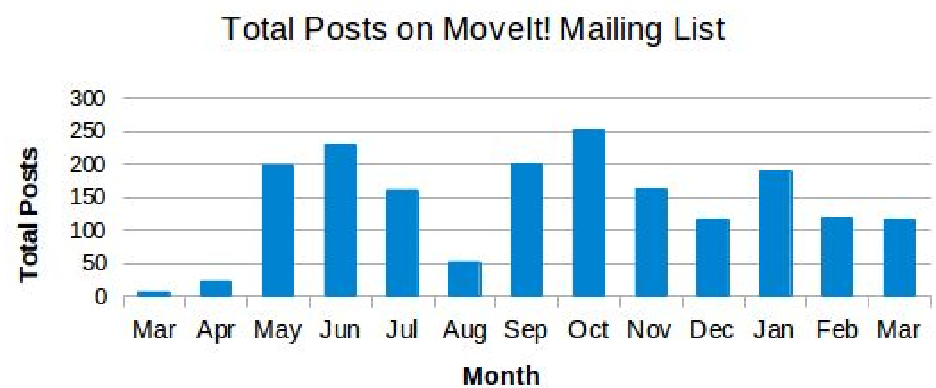
It's popular.

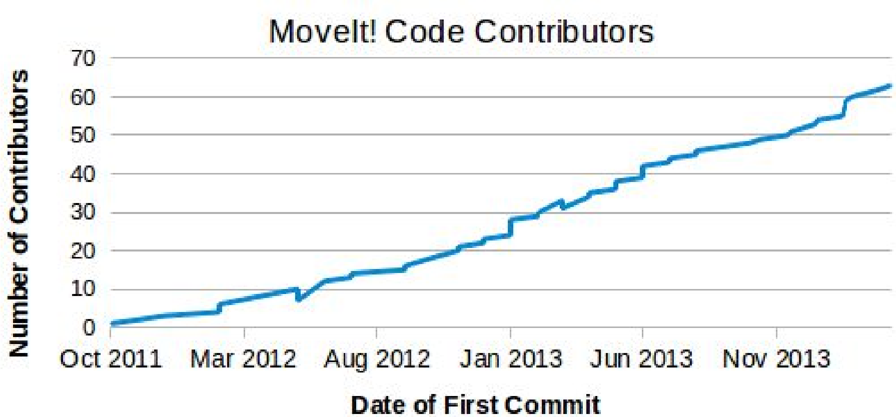
- #3 Package in ROS (ROS survey)
- 700 Members on Mailing list
- Number of installations 2015: 10,089
- ICRA 2015
  - 11 Papers cited/used MoveIt!
- IROS 2015
  - 5 Papers cited/used MoveIt!
- Has been run on over 65 robots worldwide

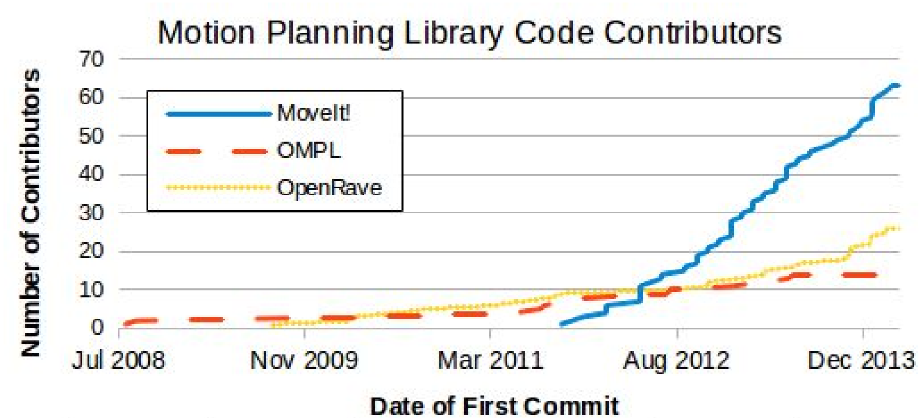
Community


Government R [15]
Undergraduate [5]
Graduate Stu [44]
Industry R&D [19]
Industry Prof [4]
Hobbyist [5]
Other [0]
Academic Fac [20]


MoveIt! Mailing List Membership
Number of Users
0
100
200
300
400
500
600
700
800
Jan 2013
Jul 2013
Jan 2014
Jul 2014
Jan 2015
Jul 2015
Date Joined


Total Posts on MoveIt! Mailing List
Total Posts
0
50
100
150
200
250
300
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan
Feb
Mar
Month


MoveIt! Code Contributors
Number of Contributors
0
10
20
30
40
50
60
70
Oct 2011
Mar 2012
Aug 2012
Jan 2013
Jun 2013
Nov 2013
Date of First Commit

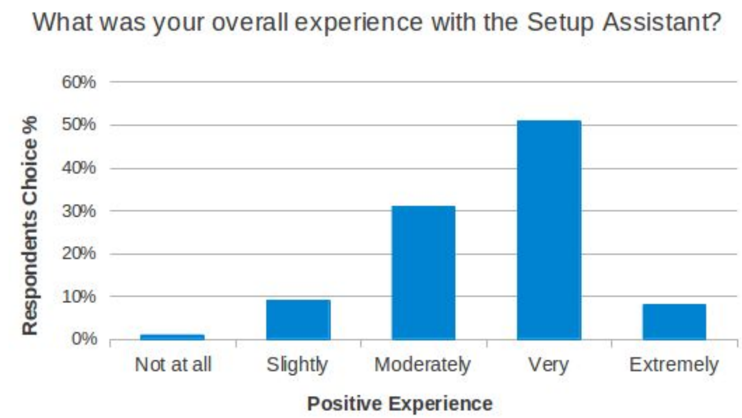

Motion Planning Library Code Contributors
Number of Contributors
0
10
20
30
40
50
60
70
MoveIt!
OMPL
OpenRave
Jul 2008
Nov 2009
Mar 2011
Aug 2012
Dec 2013
Date of First Commit

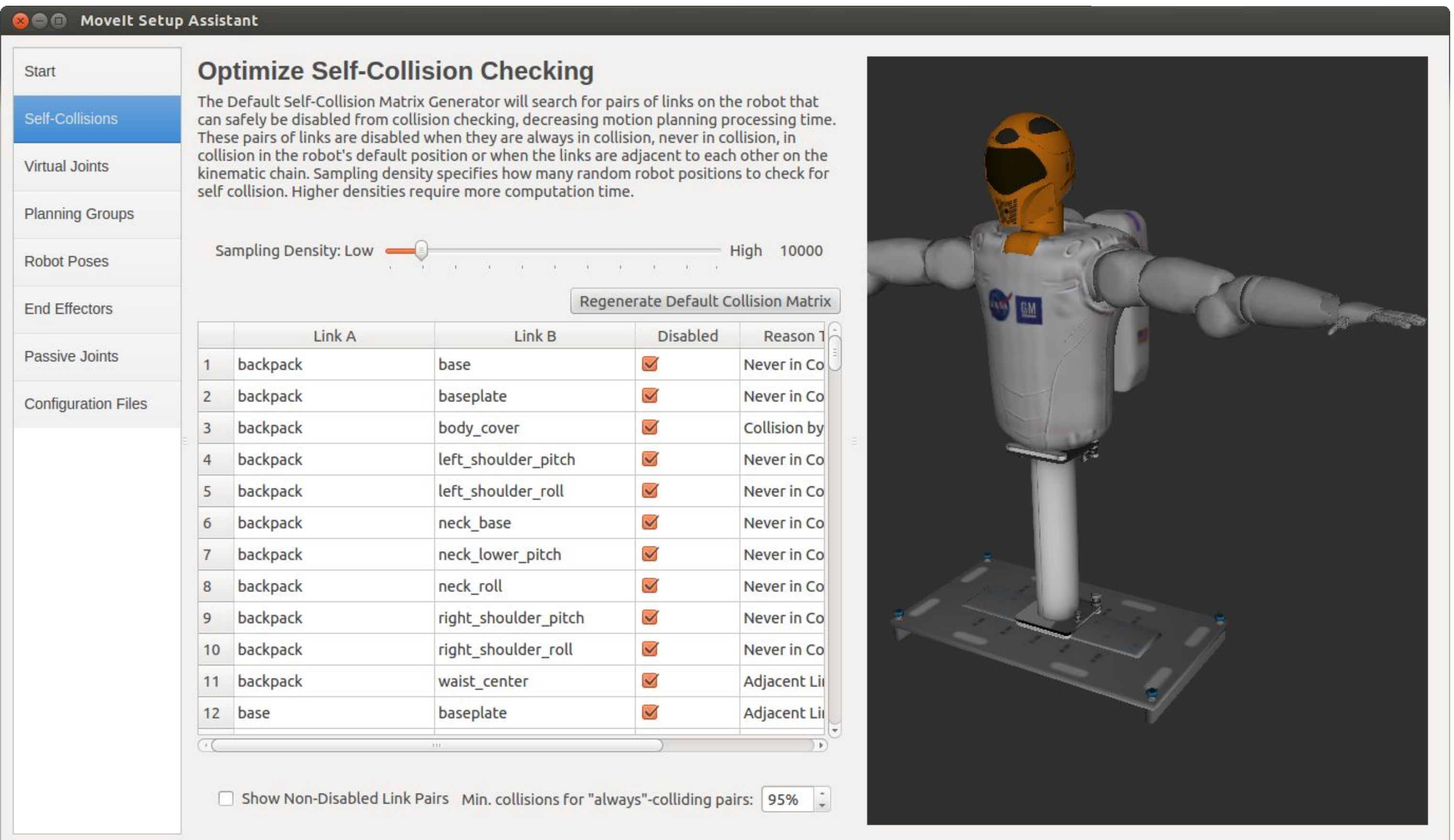
Exciting Developments

- Benchmarking getting rewrite
  - Mark Moll / Kavraki Lab
- STOMP being revived
  - Jorge Nicho / SwRI
- Descartes Cartesian Planner
  - Shaun Edwards / SwRI
- Collision detection plugin
  - Michael Ferguson / Fetch
- New release maintenance manager
  - Michael Ferguson / Fetch

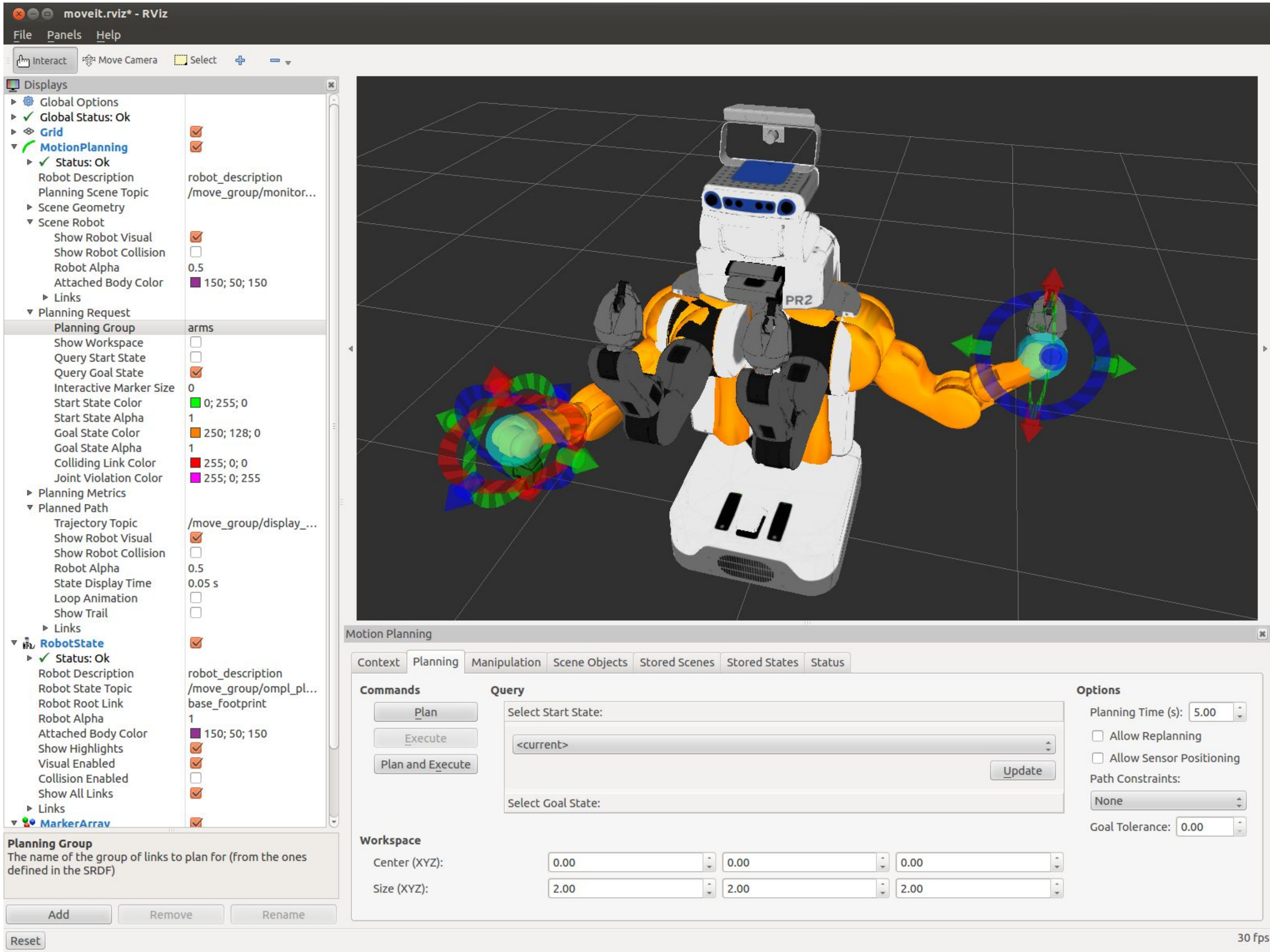
What it's done well 5min

Setup Assistant

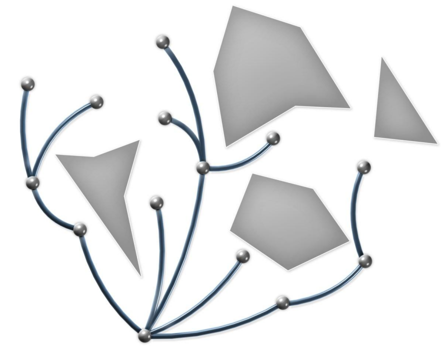




Rviz Motion Planning Plugin



Comparing research libraries



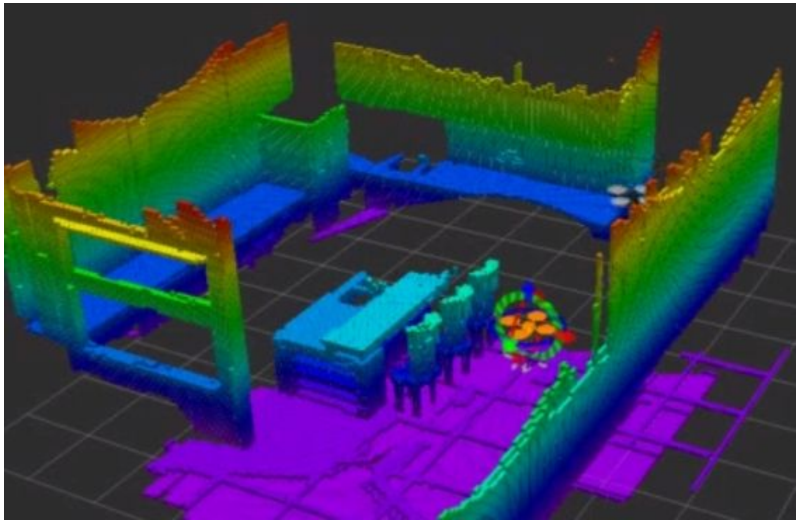
- **OMPL**
  Open Motion Planning Library
- **SBPL**
  Search Based Planning Library
- **CHOMP**
  Covariant Hamiltonian Optimization
- **STOMP**
  Stochastic Trajectory Optimization
- **FCL**
  Fast Collision Checking Library
- **PCD**
  Proximity Collision Detection
- **IKFast**
  Analytical Inverse Kinematics Solver
- **KDL**
  Kinematics Dynamics Library - Inverse Kinematics
- **Octomap**
  3D occupancy grid mapping

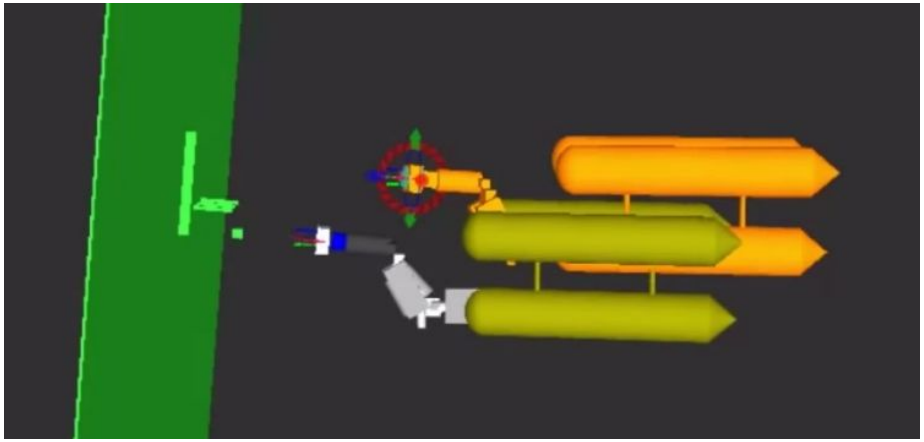
Cohen, B.; Sucan, I.A.; Chitta, S., ***"A generic infrastructure for benchmarking motion planners,"*** *in Intelligent Robots and Systems (IROS), 2012*

Robot Agnostic

Flexibility (but also complexity)

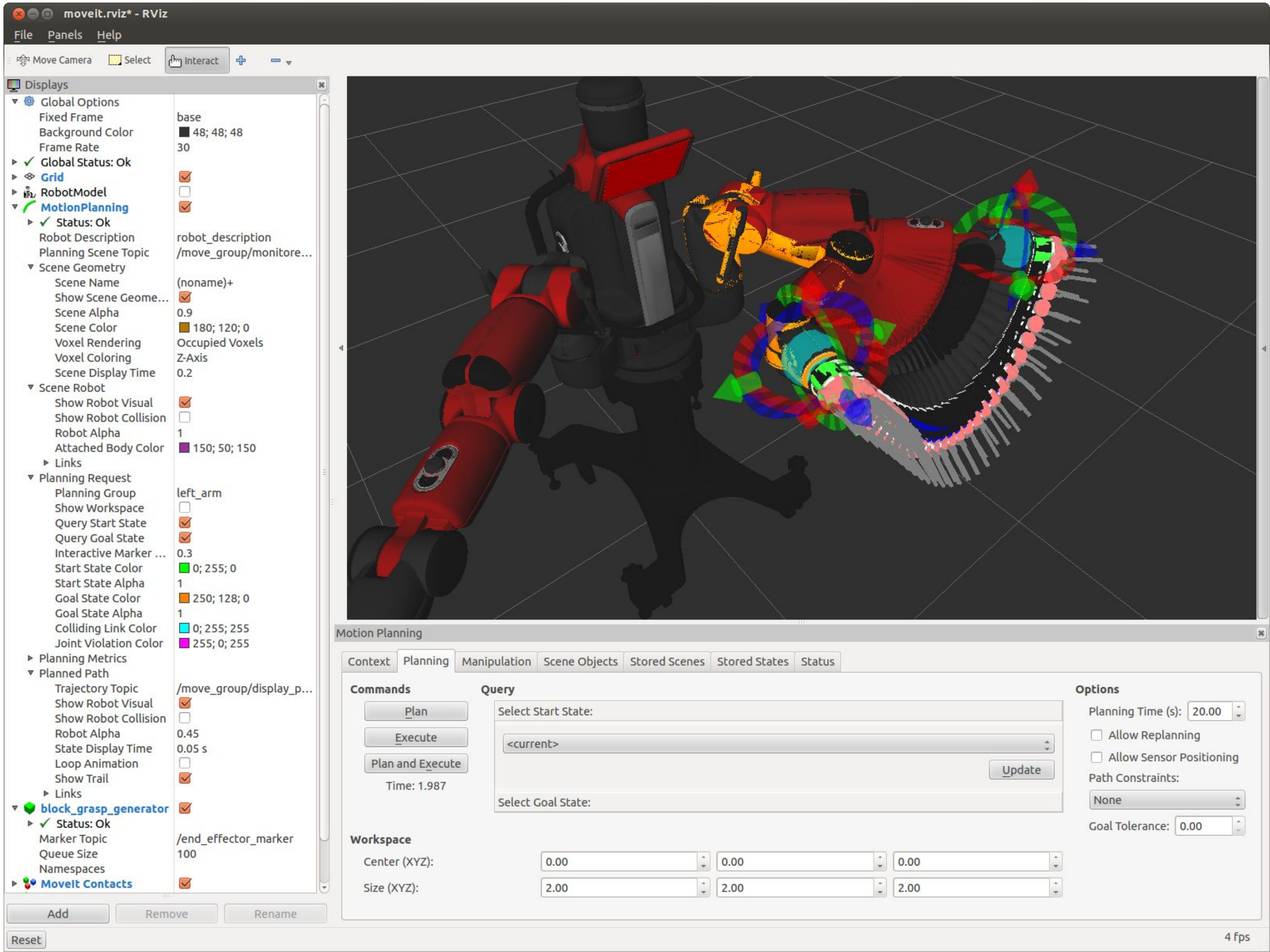
- Can Handle:
  - Groups of joints
  - Multivariable joints
  - Mimic joints
- Notions of:
  - Cartesian-Space Planning
  - Joint-Space Planning
  - Orientation Constraints
  - Visibility Constraints





Typical Use Patterns 5min

Rviz Motion Planning Plugin



MoveIt!

Commander

```
> a = current
> go rand
> wait 5
> plan a
```

```
Known commands:

  help                  show this screen
  id|which              display the name of the group that is operated on
  load [<file>]         load a set of interpreted commands from a file
  save [<file>]         save the currently known variables as a set of commands
  use <name>            switch to using the group named <name> (and load it if necessary)
  use|groups            show the group names that are already loaded
  vars                  display the names of the known states
  show                  display the names and values of the known states
  show <name>           display the value of a state
  record <name>         record the current joint values under the name <name>
  delete <name>         forget the joint values under the name <name>
  current               show the current state of the active group
  constrain <name>      use the constraint <name> as a path constraint
  constrain             clear path constraints
  eef                   print the name of the end effector attached to the current group
  go <name>             plan and execute a motion to the state <name>
  go <dir> <dx>|        plan and execute a motion in direction up|down|left|right|forward|backwar
  go rand               plan and execute a motion to a random state
  plan <name>           plan a motion to the state <name>
  execute               execute a previously computed motion plan
  rotate <x> <y> <z>    plan and execute a motion to a specified orientation (about the X,Y,Z axe
  tolerance             show the tolerance for reaching the goal region
  tolerance <val>       set the tolerance for reaching the goal region
  wait <dt>             sleep for <dt> seconds
  x = y                 assign the value of y to x
  x[idx] = val          assign a value to dimension idx of x
  x = [v1 v2...]        assign a vector of values to x
  trace <on|off>        enable/disable replanning or looking around
  ground                add a ground plane to the planning scene
  allow replanning <true|false>    enable/disable replanning
  allow looking <true|false>       enable/disable looking around
```

"move_group" Python Interface

```
group = moveit_commander.MoveGroupCommander("left_arm")

pose_target = geometry_msgs.msg.Pose()
pose_target.orientation.w = 1.0
pose_target.position.x = 0.7
pose_target.position.y = -0.05
pose_target.position.z = 1.1
group.set_pose_target(pose_target)

plan1 = group.plan()
```

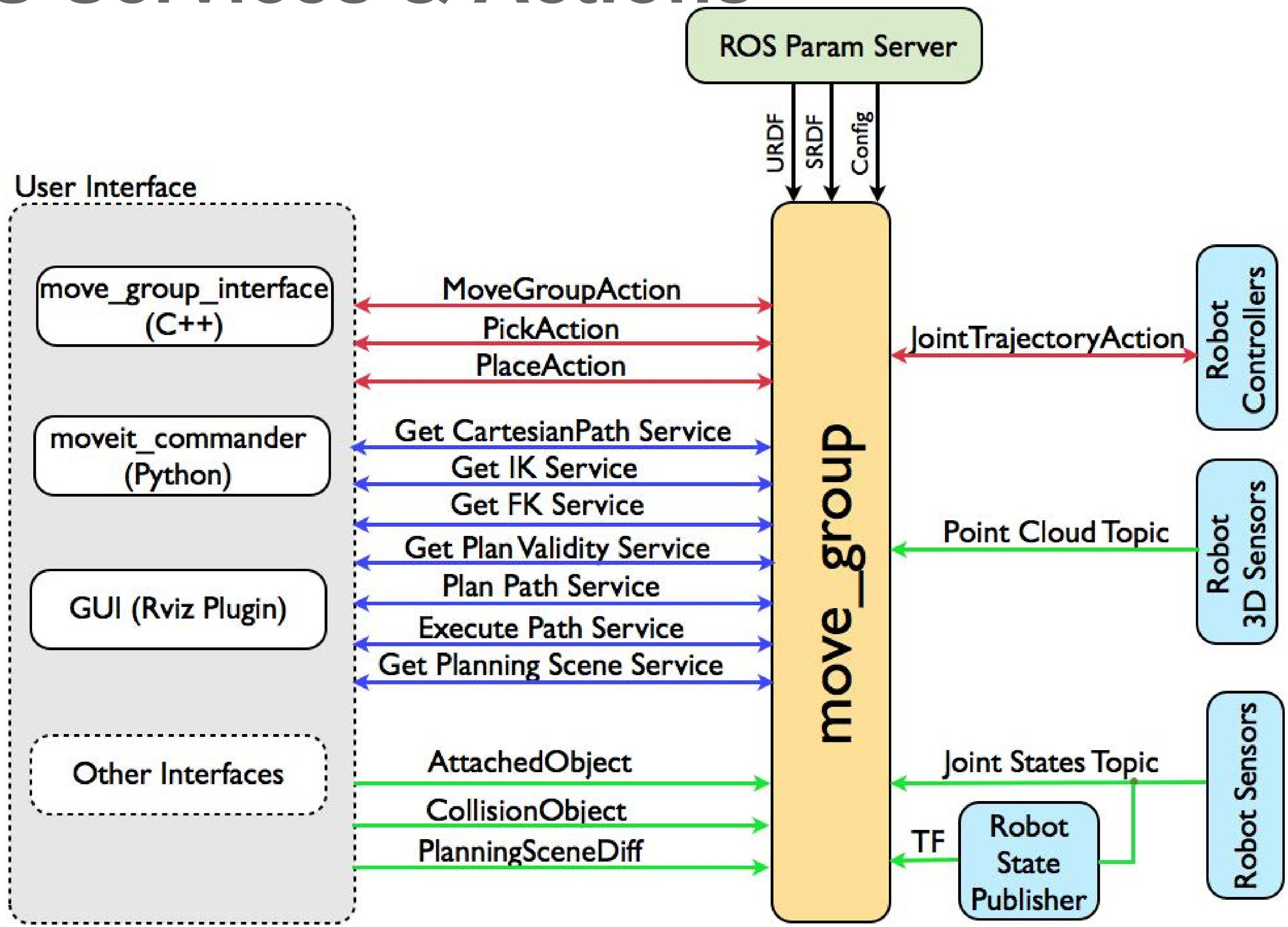
"move_group" C++ Interface

```
moveit::planning_interface::MoveGroup group("right_arm");

geometry_msgs::Pose target_pose;
target_pose.orientation.w = 1.0;
target_pose.position.x = 0.28;
target_pose.position.y = -0.7;
target_pose.position.z = 1.0;
group.setPoseTarget(target_pose);

moveit::planning_interface::MoveGroup::Plan my_plan;
bool success = group.plan(my_plan);
```

ROS Services & Actions

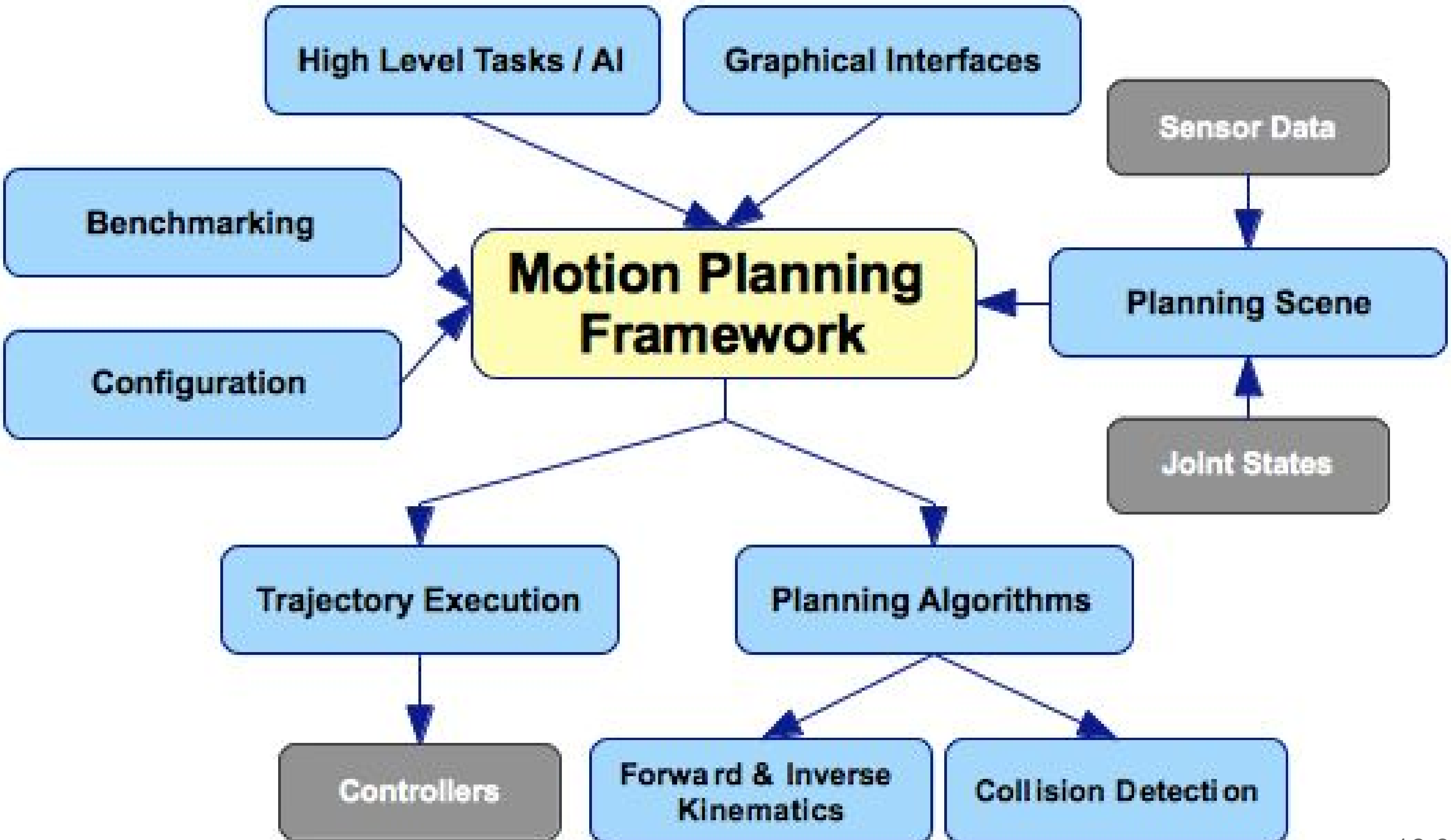


Pro-Tip: Use C++ classes individually

```
robot_model_loader_.reset(new robot_model_loader::RobotModelLoader("robot_descrition"));
robot_model_ = robot_model_loader_->getModel();
planning_scene_.reset(new planning_scene::PlanningScene(robot_model_));
tf_.reset(new tf::TransformListener(nh_));
psm_.reset(new planning_scene_monitor::PlanningSceneMonitor(
            planning_scene_, robot_model_loader_,  tf_,"my_scene"));
psm_->startStateMonitor("/joint_states", "");
psm_->startPublishingPlanningScene(planning_scene_monitor::PlanningSceneMonitor::
                                   UPDATE_SCENE, "my_planning_scene");
visuals_tools_.reset(new MoveItVisualTools(robot_model_, planning_scene_monitor_));
planning_pipeline_.reset(new planning_pipeline::PlanningPipeline(robot_model_ nh_,
                        "planning_plugin", "request_adapters"));
trajectory_execution_manager_.reset(new trajectory_execution_manager::
                                   TrajectoryExecutionManager(robot_model_));
```

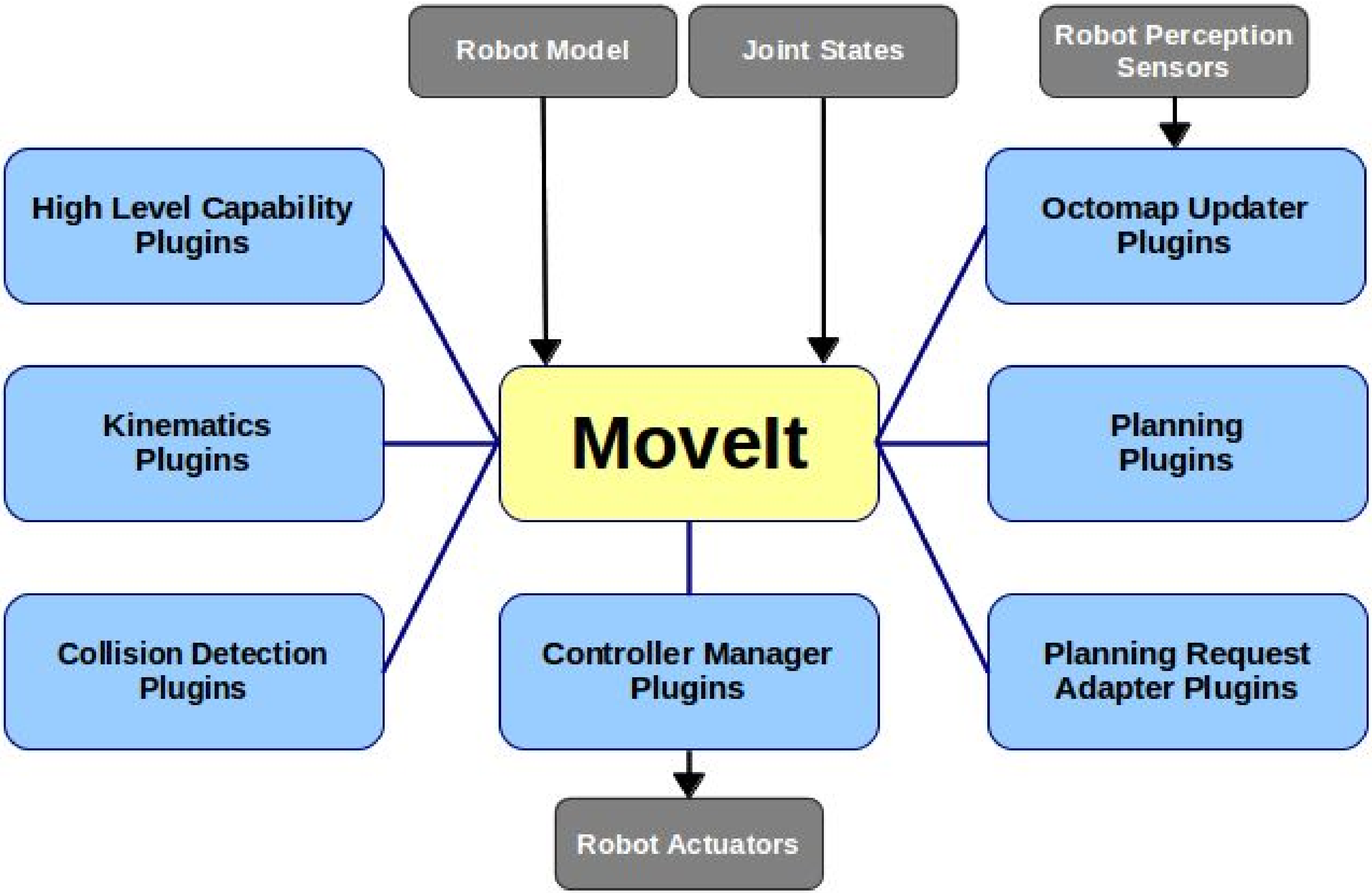
Best for researchers who want to modify code

Demystifying Complexity 5min

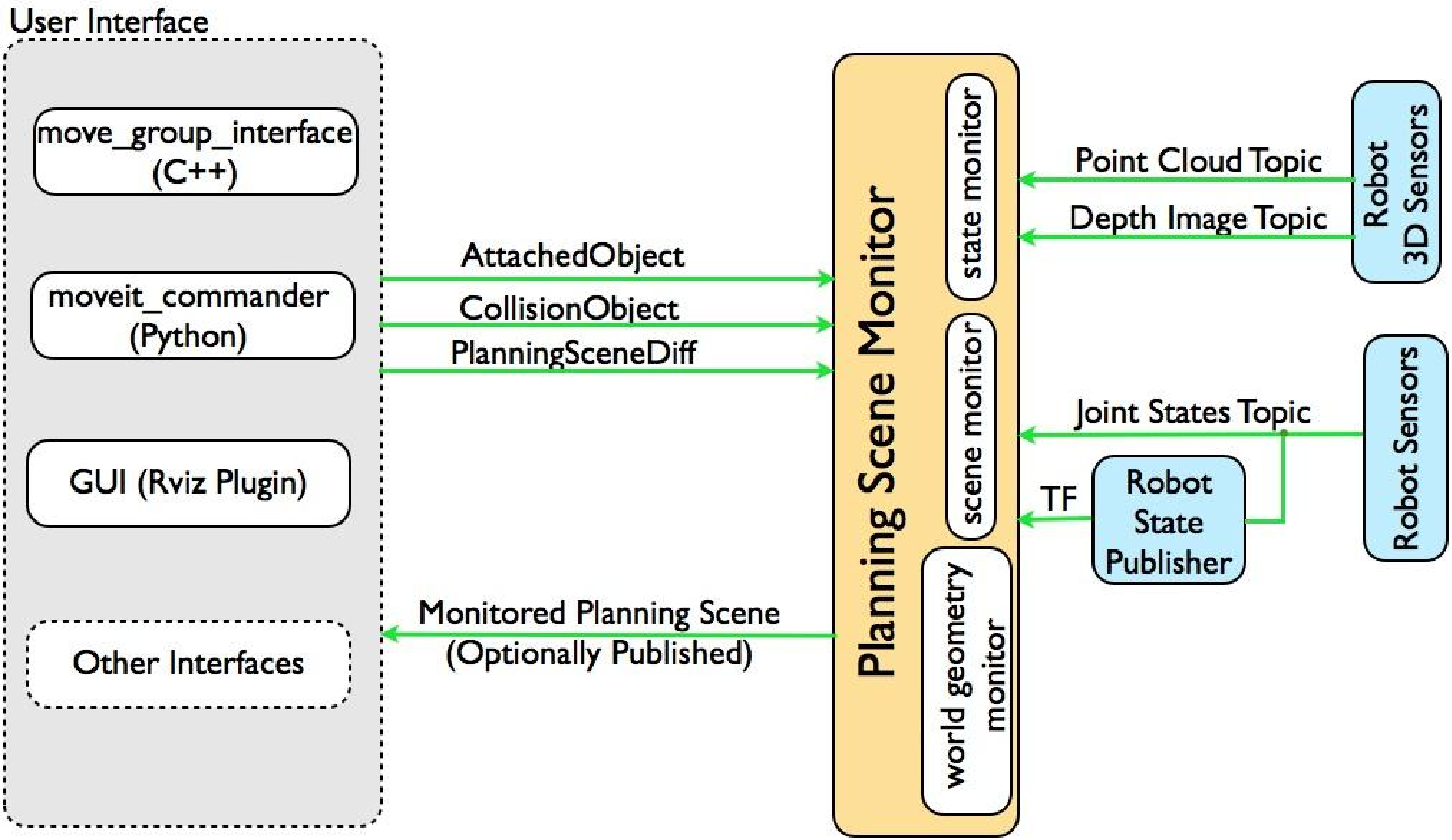


16:00

Many Plugins



Planning Scene Monitor



Planners

OMPL

- Lydia Kavraki's lab
- Sampling-based planners
- Stochastic
- Probabilistically complete
- Typically no optimality guarantees
- Computationally fast
- More reliable runtime for real-world applications
- Many variants of algorithms available

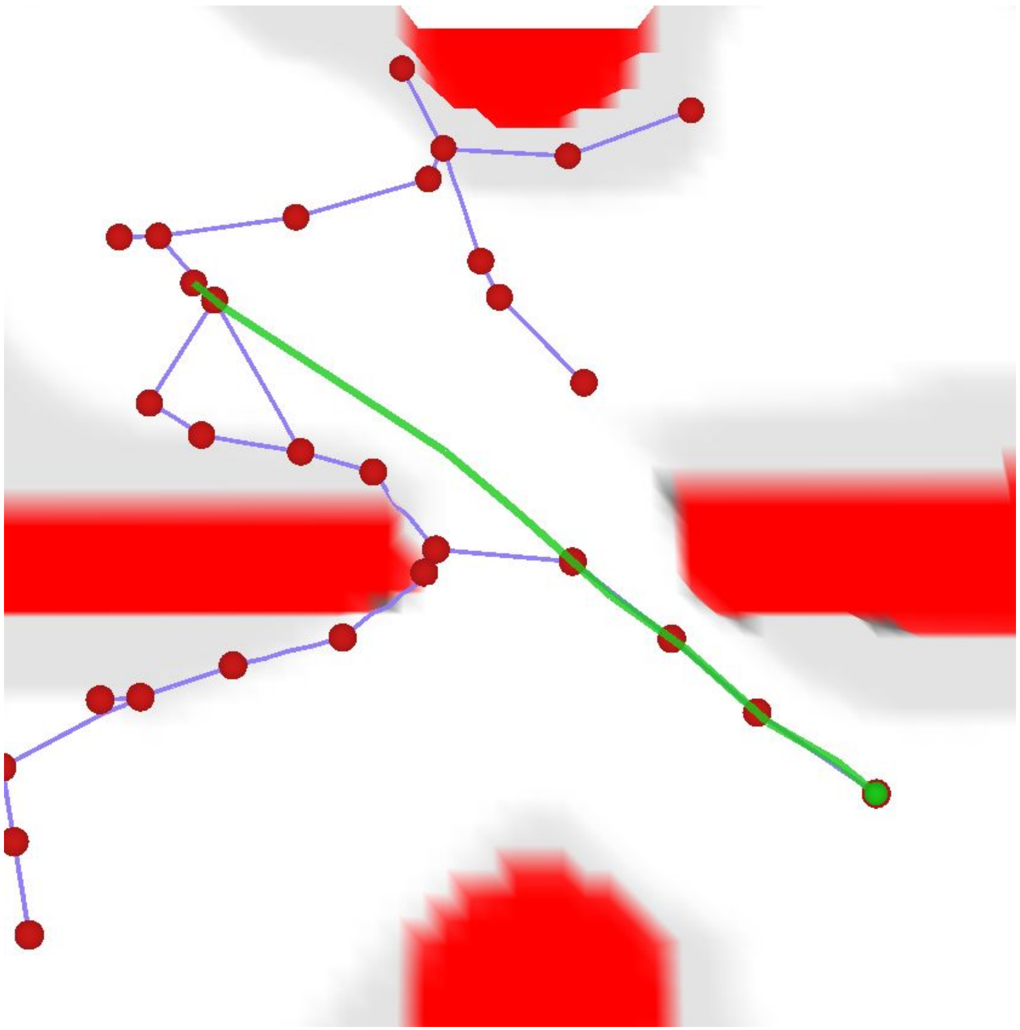
SBPL

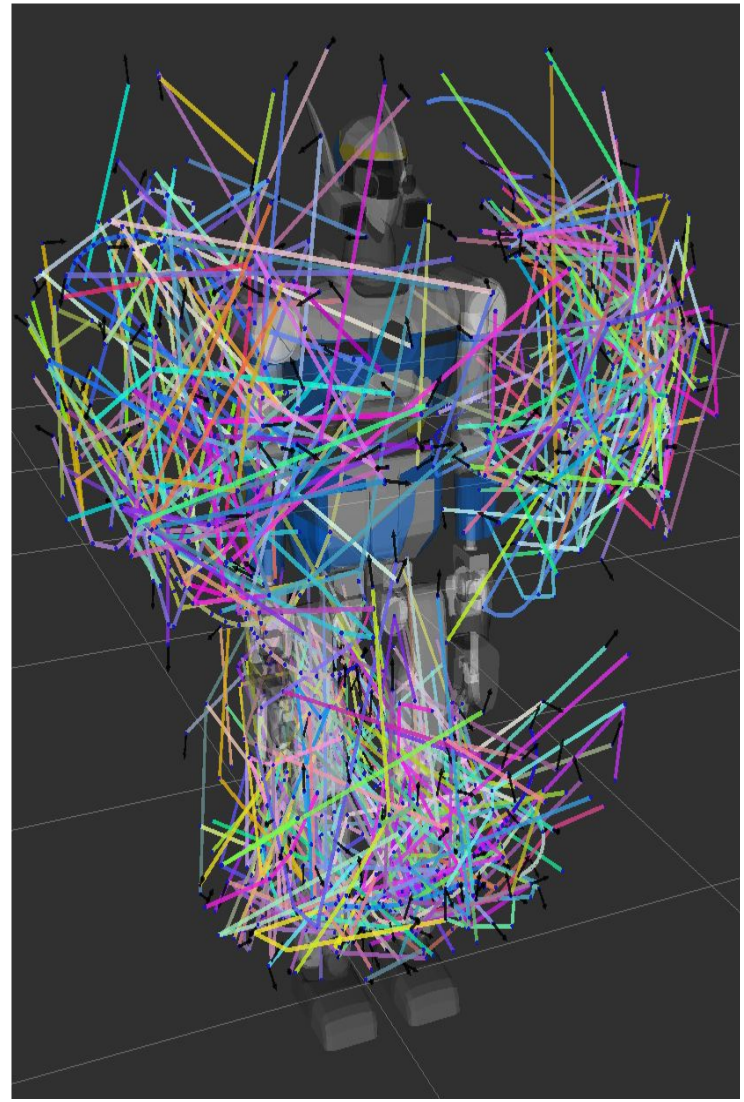
- Maxim Likhachev's lab
- Graph-based planners
- Deterministic
- Resolution complete
- Optimality guarantees
- Requires pre-processing phase
- Computationally expensive
- More reliable solutions for real-world applications
- Renewed work from Michael Ferguson

CHOMP/STOMP

- Kalakrishnan et al
- Optimization-based planner that generates smooth well behaved collision free motion paths in reasonable time
- Can incorporate additional objective functions - collision avoidance and smoothness
- CHOMP being resurrected by ROS Industrial group

Experience Planners





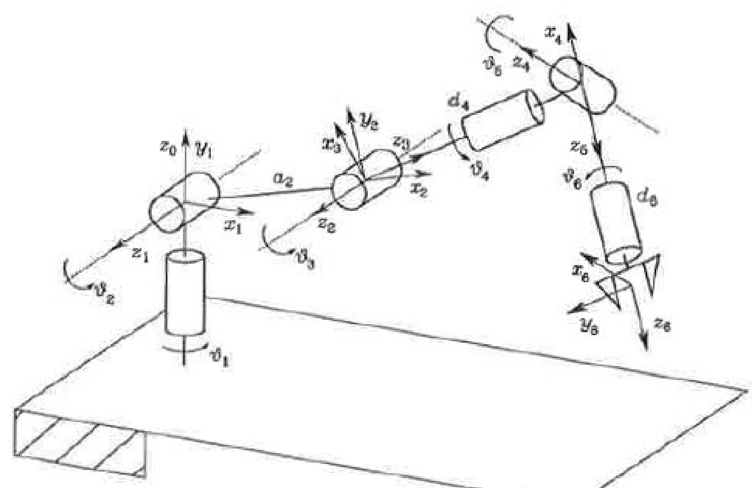
Planner Request Adapters

- **AddTimeParameterization**
  - Modifies the timestamps of a kinematic (position-based) trajectory to respect velocity and acceleration constraints
  - Uses iterative parabolic time parameterization
- **FixWorkspaceBounds**
  - If no minimum workspace bounds is specified, sets to a default
- **FixStartStateBounds**
  - Tweaks joints to not be outside joint limits
  - Accounts for floating point and encoder noise
- **FixStartStateCollision**
  - Tweaks start state to not be in collision with environment
  - Creates a new planning request with modified start state
- **FixStartStatePathConstraints**
  - Plans separate path from invalid start state to valid start state

Adapts research theory to real world hardware

IK Solvers

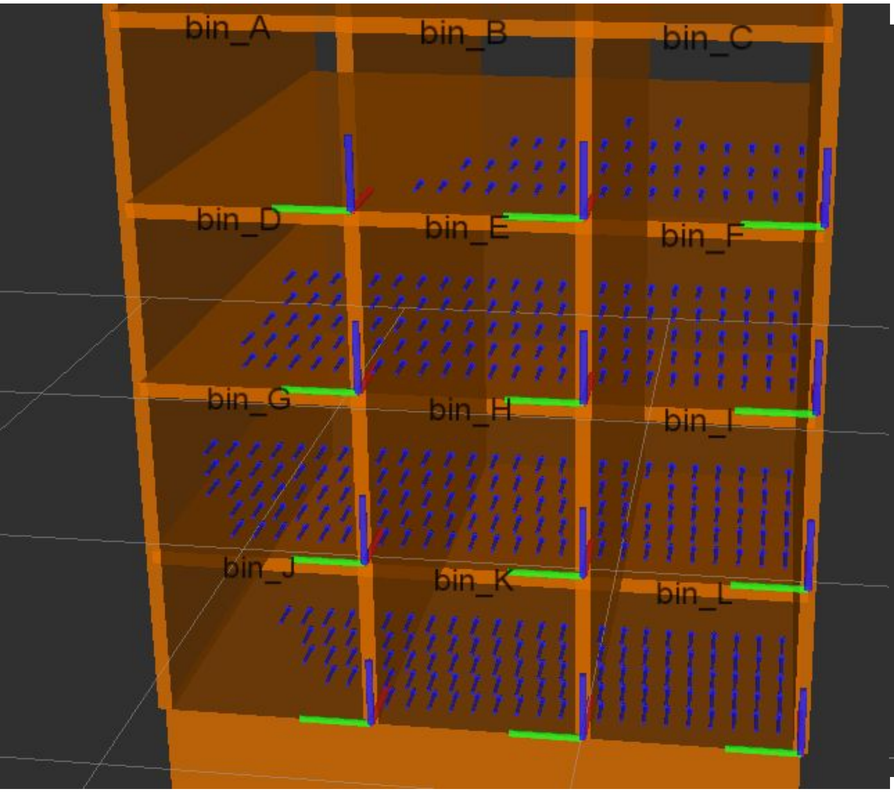
- KDL
  - Kinematics Dynamics Library, OROCOS
- IKFast
  - OpenRave Analytical
- Robot-specific custom solvers
  - PR2

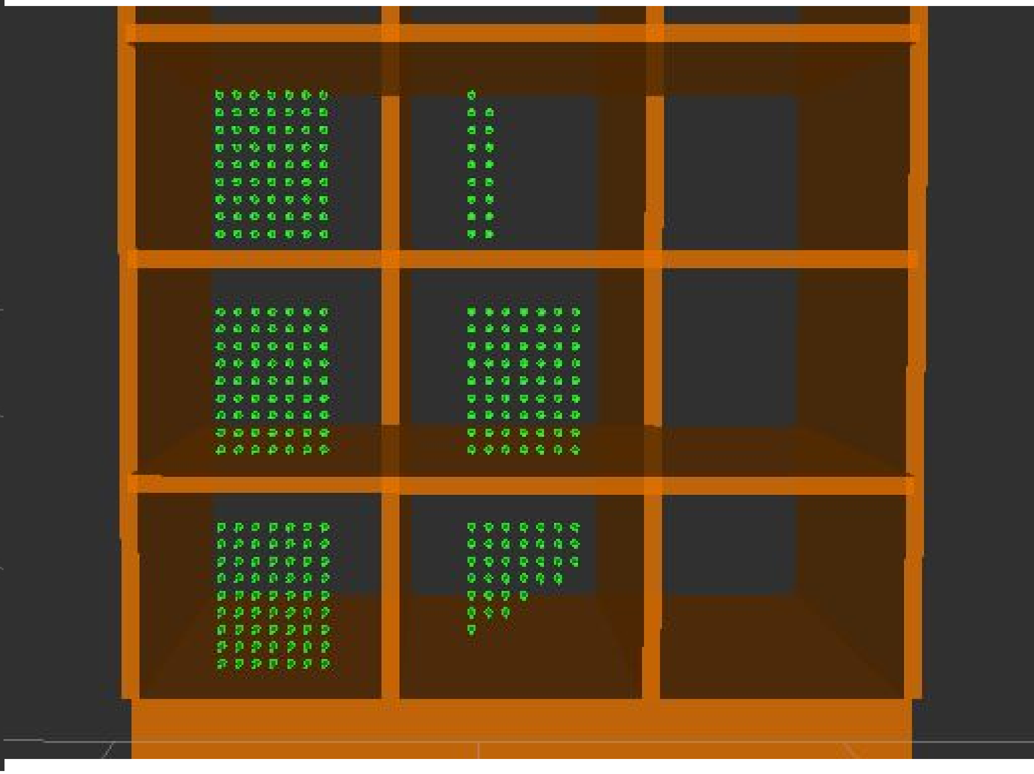


Amazon Picking Challenge 5min

21:00

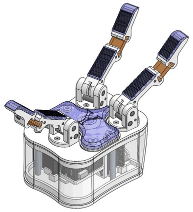
Workspace Analysis





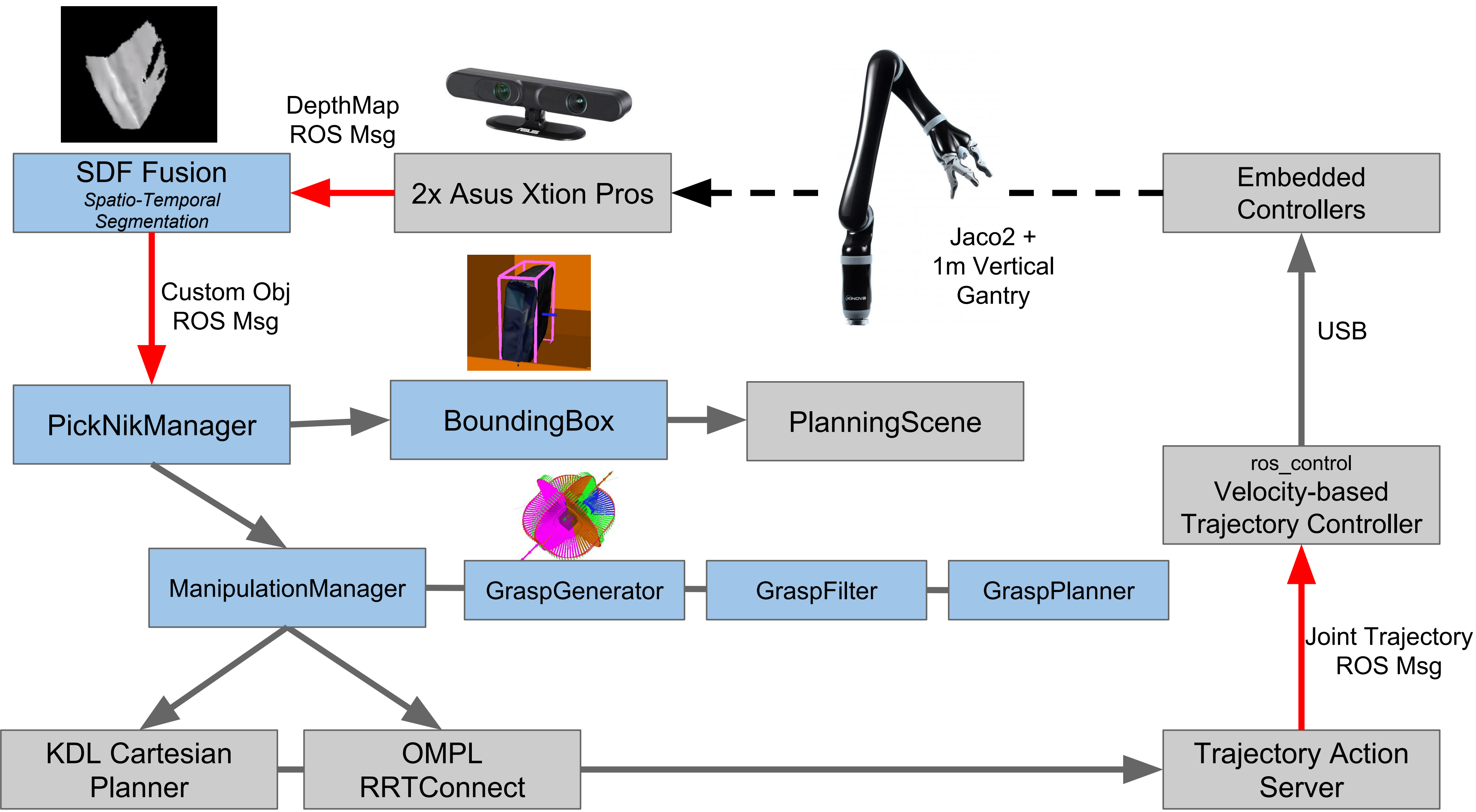
Baxter Parallel Electric Gripper

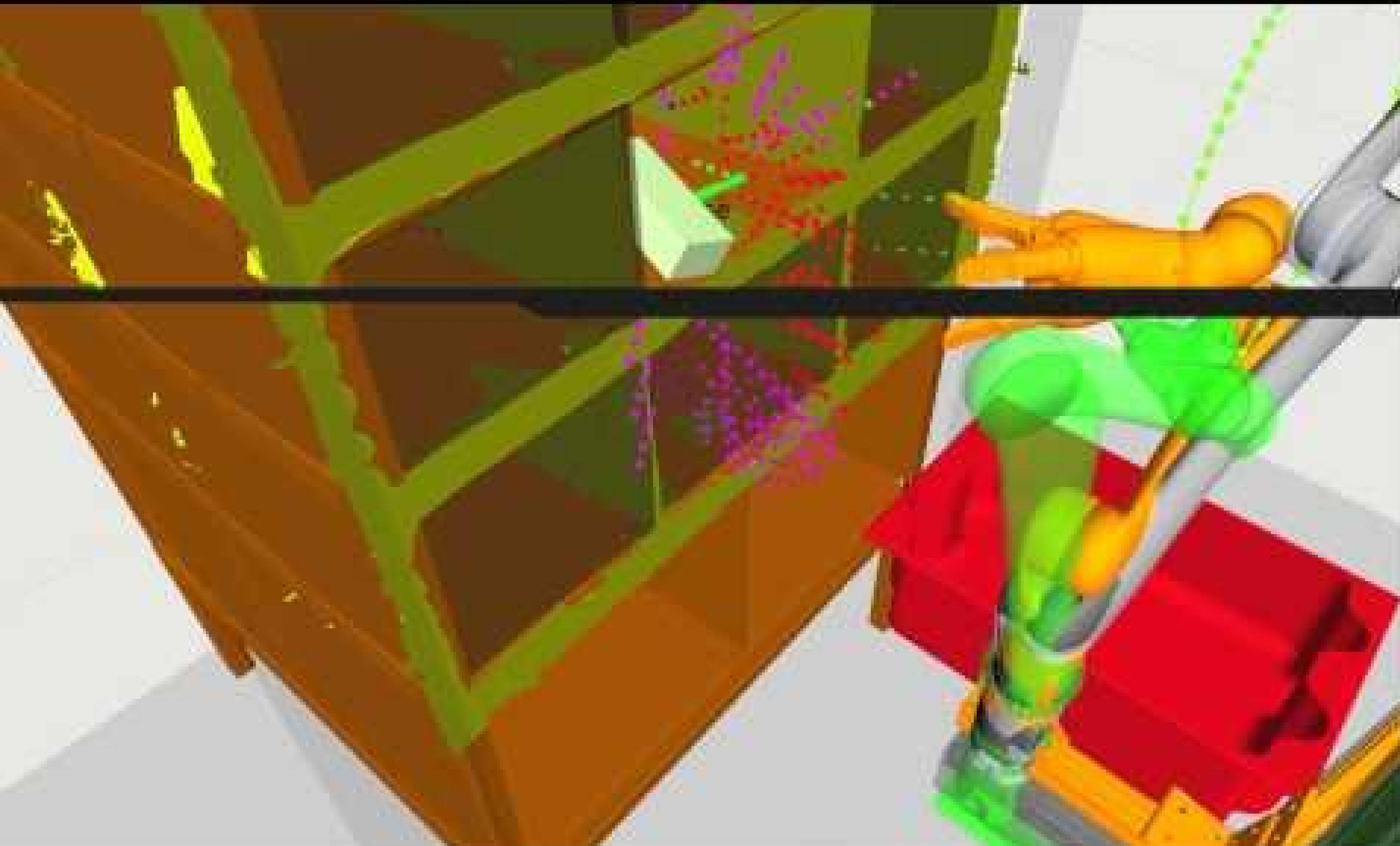
Yale OpenHand 3-Finger Gripper



Kinova Jaco2 + 1m Vertical Gantry

PickNik Architecture (Pretty Standard)





Challenge Takeaways

- Simplest possible grasping → suction
- Low cost hardware → visual servoing
- Reduce calibration needs
- 2 mobile bases won → larger workspace
- Slim arms → better reachability
- Good visualizations → introspection and development
- Perception and manipulation teams must work closely
- Test whole system working together often

MoveIt! used by at least 10 teams:

  - PickNik, Z.U.N., University of Washington, Team IntBot, NUS_SMART_HAND, Team Applied Robotics, Team WPI, University of Alberta Team, Plocka Packa, Team CVAP

None of the winning teams used MoveIt!

Team RBO - 1st Place

TU Berlin

148 points

Barrett WAM arm

- Backdrivability key to skillful interactions with the environment

Nomadic XR4000 mobile base

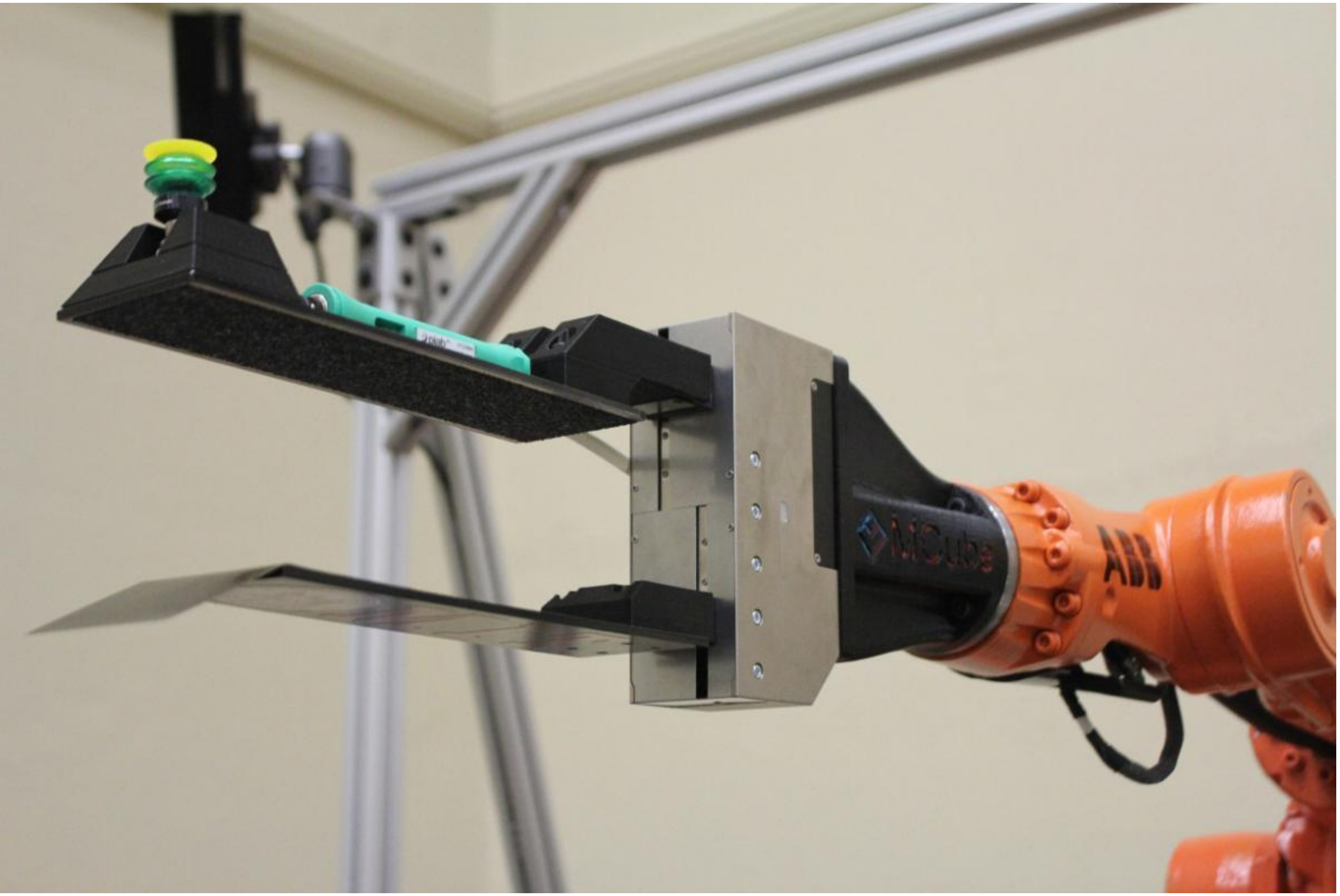
- Omnidirectional / holonomic
- Very large workspace

Did not rely on motion planning

Hybrid automaton composed of sequences of controllers with sensor-based transitions.

Vacuum attachment tool with suction cup drilled into side of fender

“Simple but robust” RGB object recognition algorithm



Team MIT - 2nd Place
MIT
88 points

ABB 1600ID
- Sub-millimeter precision
- Internal canals for cables

Custom dual-purpose end effector
- Aviation-grade aluminum
- Spatula-like finger nail
- Suction also

Used MIT Drake (Locomotion Group at MIT) for motion planning

Automatically chooses which motion primitive to use based on dynamics simulator
- **grasp, suck, scoop, toppling, push-rotate**

Kinect2 cameras mounted on frame, Realsense on arm

Outsourced perception to a robotics startup - Caspen Robotics

Team Grizzly - 3rd Place

Oakland University
w/Dataspeed Inc.
35 points

- Rethink Baxter
- Custom Mobile Base
- Yale OpenHand
- Suction gripper
- Kinect2 on Head

Custom Cartesian motion planning algorithm accepted position and orientation commands from the perception system

Where MoveIt! needs improvement 4min

(and where you can help!)

Motion Planner Reliability

- Sometimes fails with difficult to understand explanations
- Sometimes generates very suboptimal paths

Solutions:

- Hybridize several planning attempts (threads)
- Plan with cost functions, e.g. RRT*, PRM*
- Increase the time MoveIt! spends on smoothing paths
- For some applications, planners other than OMPL's defaults are better
- Improve user feedback to diagnose setup issues
  - Check your joint space is parameterized correctly (<2pi)
  - Introspection tools

Obstacle Clearance

Can generate plans that come very close to obstacles
Solutions:

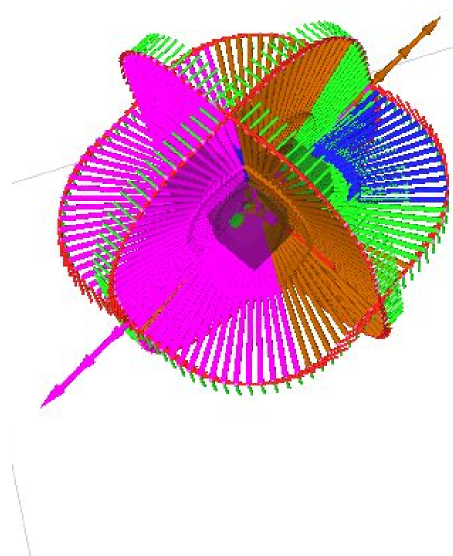
- Add out of box support for biasing trajectories away from obstacles
- Cost-based OMPL, STOMP, CHOMP

Grasping Support

Difficult to generate grasps in MoveIt!

Solution:

- Provide default URDF + SRDF compatible grasp generator
- Clearer documentation on how to integrate third party grasping pipeline



Documentation

- Need more exhaustive documentation from community support (*you!*)
- Tutorials for how to use MoveIt! beyond quick start demo
- Make it easier for our many users to contribute back

Future Roadmap 10min

Community meeting's end of year goals

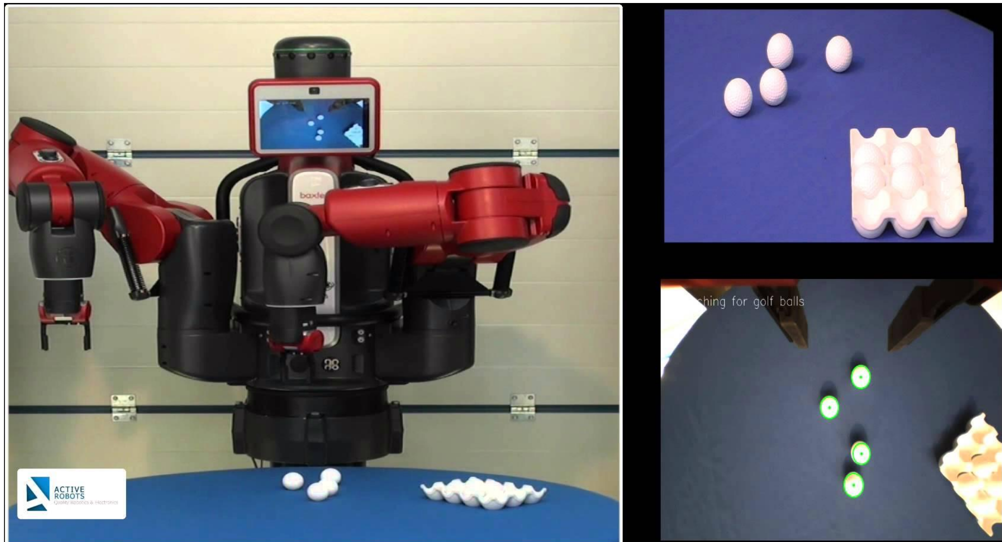
- Integrate better support for humanoid kinematics
- Integrate benchmarks updates
- Resurrect support for other types of planners (STOMP)

Visual Servoing Support

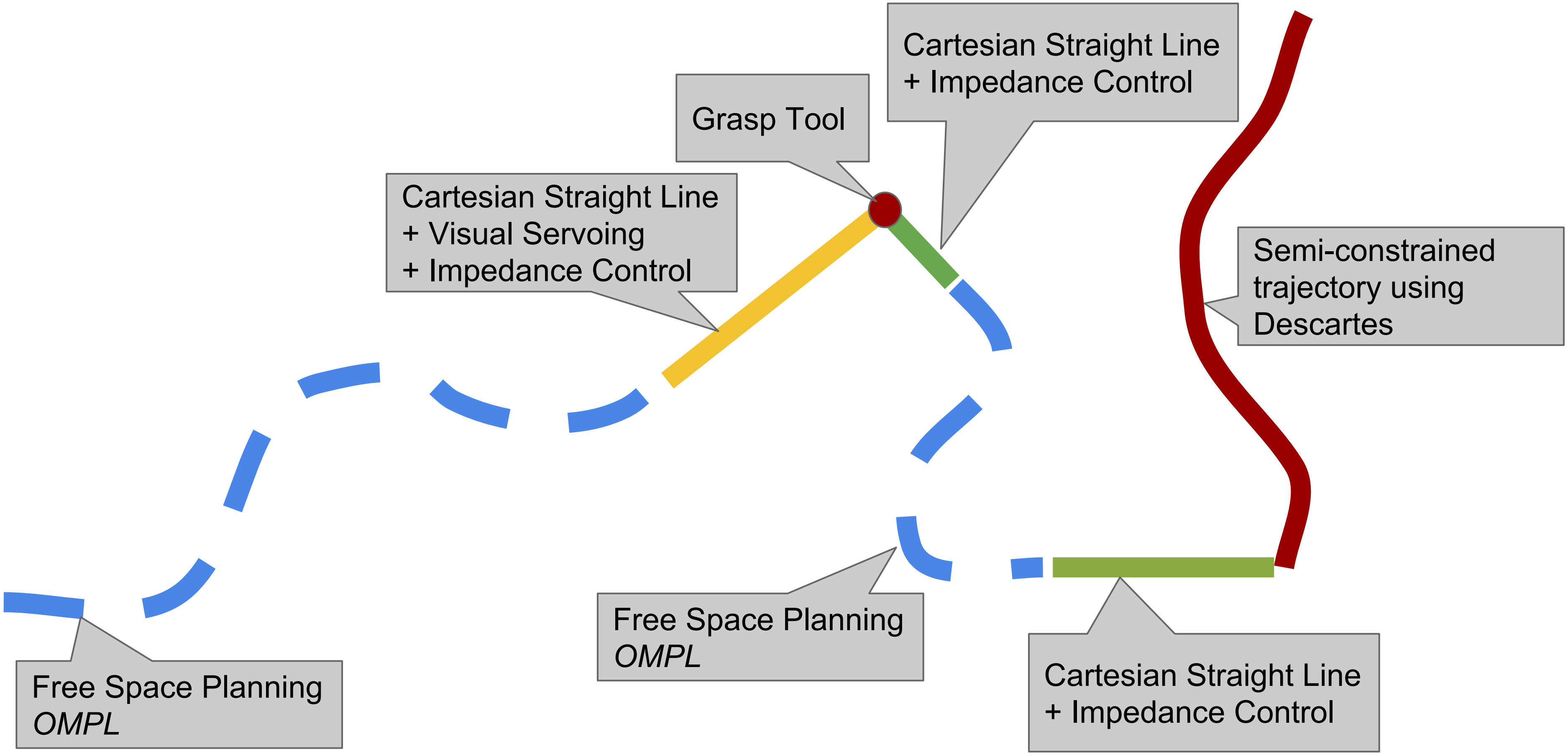
- Once a trajectory is planned, no easy way to integrate visual or tactile feedback

Solution:

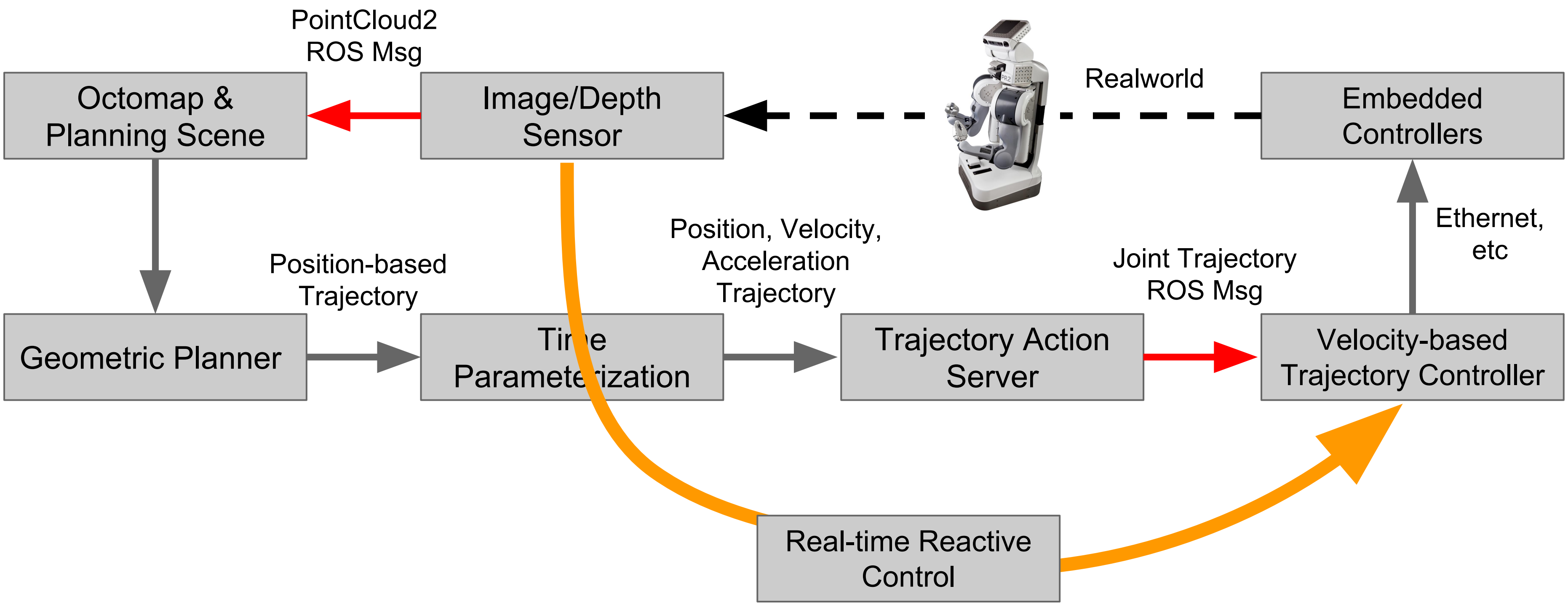
- Position/pose-based visual servoing (PBVS)
- Hooks to modify plan based on alignment of target object
- Ability to add meta-data to trajectories indicating when to use VS, what objects to track
- Requires much tighter coupling with controllers, planners, and perception system



Planning with Behaviors



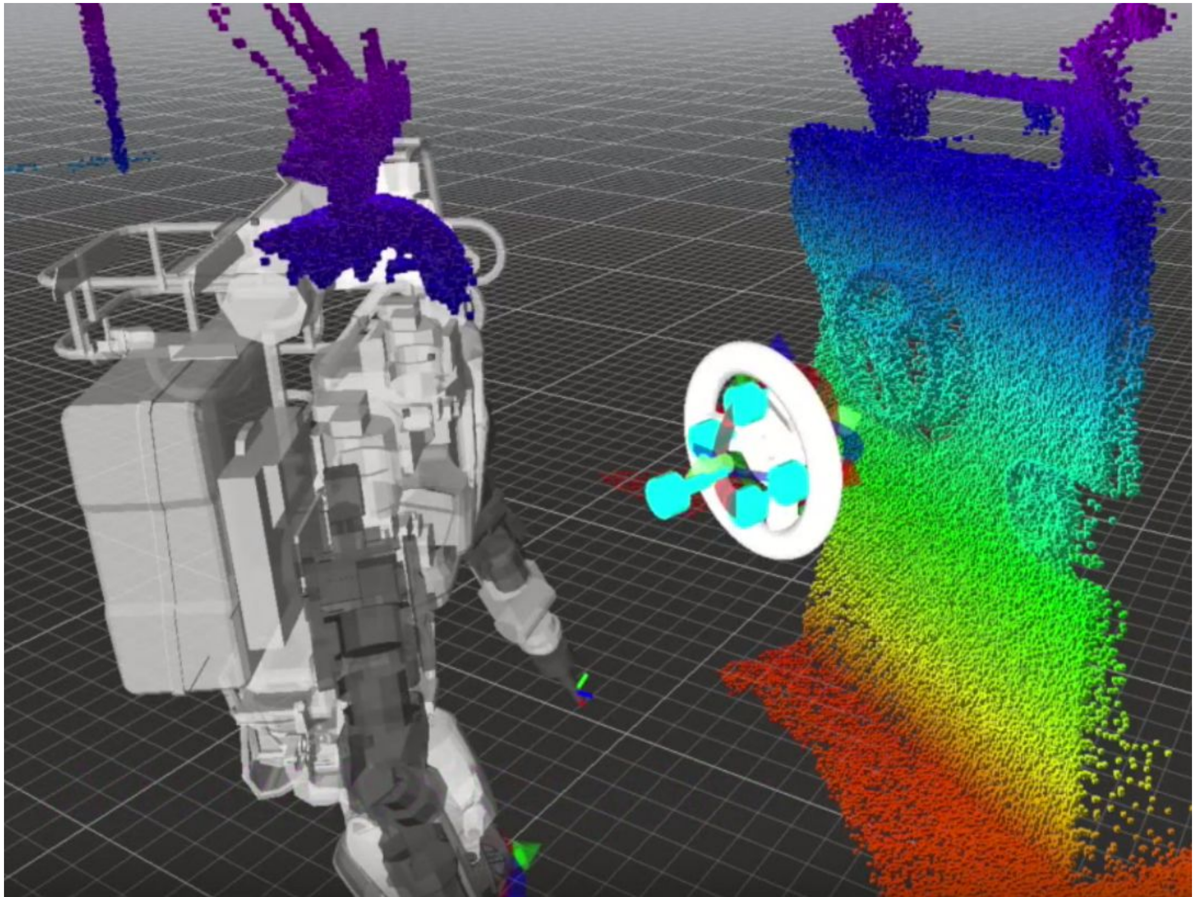
Sense-Plan-Act & ROS Control



- Faster connection for streaming commands
- Integrate ros_control with Setup Assistant
- Rename MoveIt ControllerManager to Controller*Interface*
- More advanced plugin than SimpleControllerManager
- Switching controllers

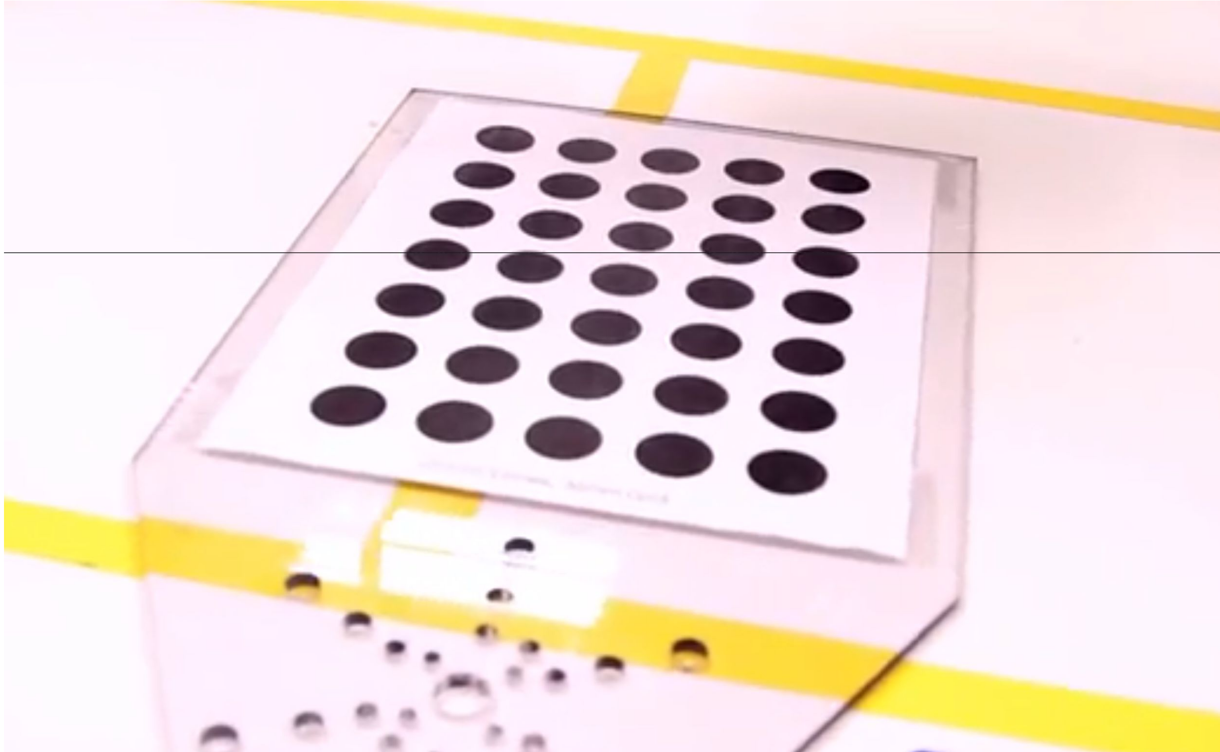
Affordance Templates

Human in the loop tools for high level commands such as more sophisticated interactive markers



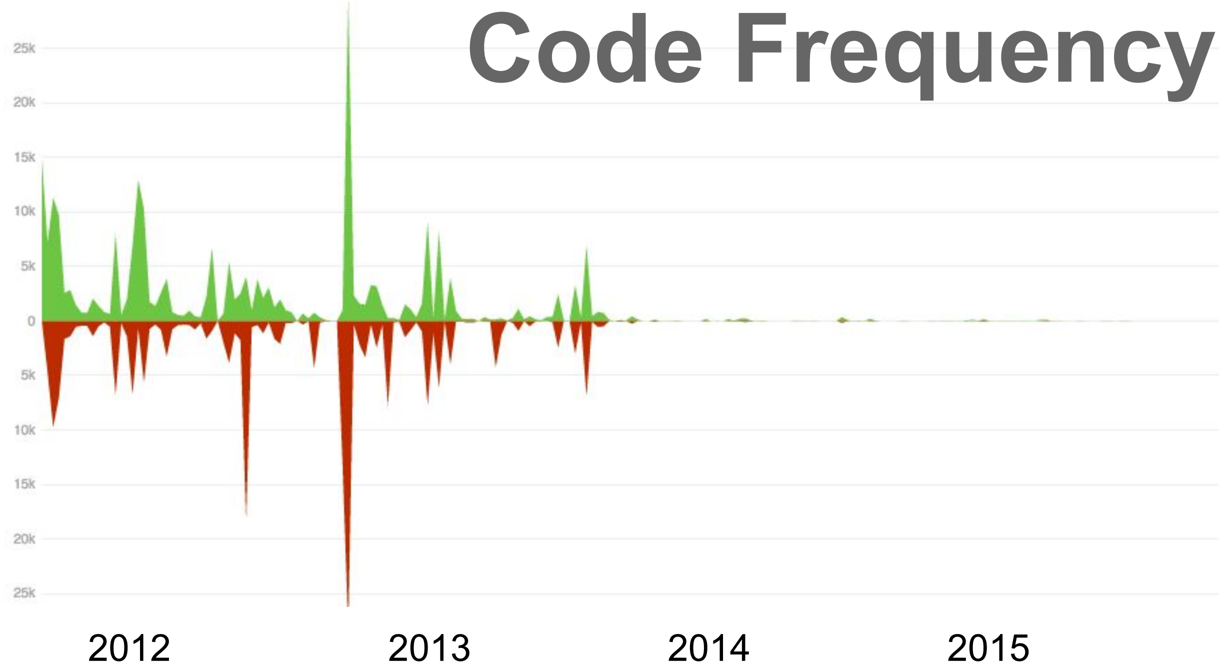
Calibration

- Integrate better calibration packages
  - ROS Industrial - industrial_calibration
  - Fetch Robotics - robot_calibration
- More clearly document how this should be integrated



Code Frequency

moveit_core



moveit_ros

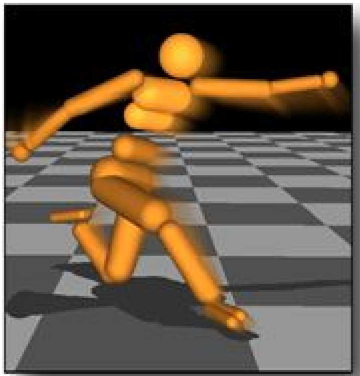
Stability vs. Progress

- MoveIt! needs to stay current
- Other motion planning frameworks are very capable
  - OpenRave, MIT Drake, MuJuCo + Whole Body Planning, etc

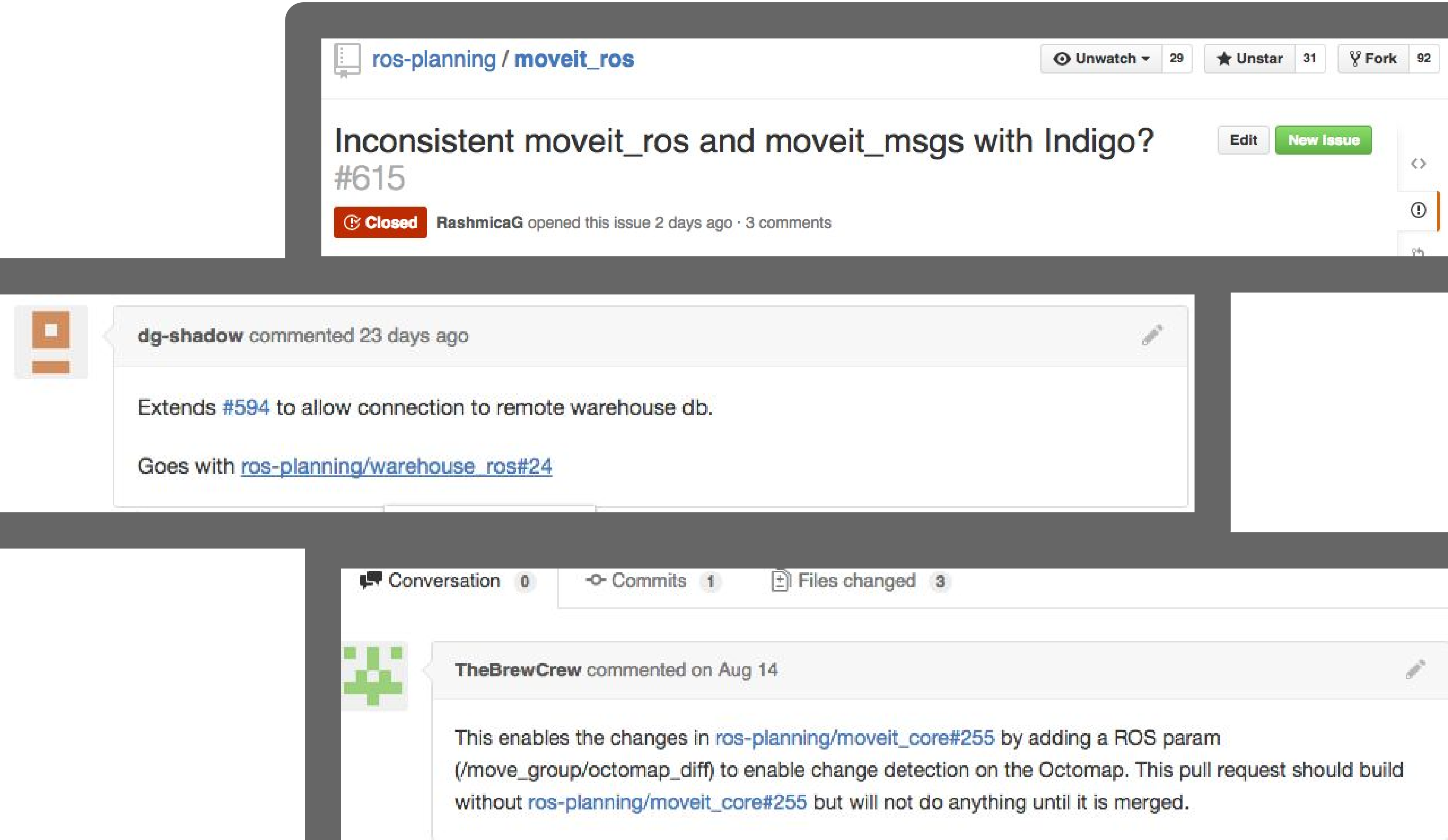
Distributed Software Collaboration Is Hard

- Currently we have 50 open pull requests
- Need continuous integration badly
- Need more simulation tests

If there are good features worth upgrading to, breaking changes are tolerable.



Proposal: Consolidate to One Repo



Too many repos to keep synced

moveit_core
moveit_ros
moveit_planners
moveit_docs
moveit_msgs
moveit_robots
moveit_ikfast
moveit_commander
moveit_kinematic_tests
moveit_advanced
moveit_setup_assistant
moveit_metapackages
moveit_plugins
moveit_resources
moveit_pr2

<u>packages not yet in **ros-planning** group:</u>

moveit_benchmarks
moveit_visual_tools
moveit_simple_grasps
moveit_python
moveit_web
moveit_whole_body_ik
industrial_moveit

Plus many ros-planning packages not prefixed with moveit_*

PointCloudLibrary / pcl — Watch 209 — Star

Point Cloud Library (PCL) http://www.pointclouds.org

9,683 commits | 1 branch | 25 releases | 218 contributors

Branch: master — pcl / +

Merge pull request #1352 from VictorLamoine/add_pointWrange …

taketwo authored a day ago — latest commit 3890dc7cac

2d	Fix include in 'keypoint.hpp' (2d module)	a year ago
3rdparty	moved 3rdparty android make files to the mobile repository	3 years ago
apps	Use lazyProduct to fix compilation on ppc64el	a month ago
cmake	CUDA >= 7.5 supports clang with libc++	7 days ago
common	Fix wrong member copy when using = operator in PCA	21 days ago
cuda	Check WITH_* variables instead of *_FOUND and BUILD_* everywhere in …	a month ago
doc	Replace dead links with Wayback machine snapshots	19 days ago
examples	Fixed compile error related to example projects	a month ago
features	Fix wrong erasing order on feature_map_ in PFHEstimation	19 days ago
filters	passthrough: Fix user_filter_value_ not being used at all	2 months ago
geometry		8 months ago
gpu	fix run-time exception bugs when '--viz' cmd-arg used	8 days ago
io	Merge pull request #1334 from VictorLamoine/fix_dssk_warnings	20 days ago
kdtree	Various doxygen fixes	a year ago
keypoints	Merge pull request #1102 from soyersoyer/preinc_iterator	9 months ago
ml	preincrement iterators to avoid the temporary	9 months ago
octree	Merge pull request #1297 from rhuitl/master	2 months ago
outofcore		8 months ago
people	Check WITH_* variables instead of *_FOUND and BUILD_* everywhere in …	a month ago
recognition	fixed linemod func memory leak issue	6 months ago
registration	Transform point cloud in GICP6D align function	a month ago
sample_consensus	Normalizing optimized cone direction	a month ago
search	updated estimateProjectionMatrix() to make the down-sampled image hav…	6 months ago
segmentation	Bug fix	16 days ago
simulation	Warning fixes	6 months ago
stereo	Fix warnings in stereo/digital_elevation_map.h	9 months ago
surface	Fixed compile error related to example projects	a month ago
test	seed rand to make tests reproducible	a month ago
tools	Check WITH_* variables instead of *_FOUND and BUILD_* everywhere in …	a month ago
tracking	Fixed error with initialized Vector and openMP	5 months ago
visualization	Add pcl::PointWithRange to the list of core point types	a day ago
.travis.sh	Improve Travis script	a month ago
.travis.yml	Add dvipng package to generate math formulas in tutorials	a month ago
AUTHORS.txt	renamed license and authors	4 years ago
CHANGES.md	Add a changelist for 1.7.2	a year ago
CMakeLists.txt	Add CMake module for DepthSense SDK	20 days ago
CONTRIBUTING.md	Add "License" section to CONTRIBUTING.md	2 years ago
LICENSE.txt	changed base license to point to OP	3 years ago

Itseez / opencv — Watch 917 — Star 6,792 — Fork

Branch: master — opencv / modules / +

Merge pull request #5436 from jet47:fix-cuda-normalize

alalek authored 4 hours ago — latest commit 76afd9a1b5

..		
calib3d	IPPInitSingelton was added to contain IPP related global variables;	a day ago
core	IPPInitSingelton initialization guards;	a day ago
cudaarithm	fix cuda::normalize (dtype < 0) case	9 hours ago
cudabgsegm	move obsolete algorithms from cudabgsegm to cudalegacy:	9 months ago
cudacodec	Adding support for WinRT(WinPhone 8/8.1 and Win Store) via CMake 3.1	7 months ago
cudafeatures2d	Adding support for WinRT(WinPhone 8/8.1 and Win Store) via CMake 3.1	7 months ago
cudafilters	Adding support for WinRT(WinPhone 8/8.1 and Win Store) via CMake 3.1	7 months ago
cudaimgproc	Use stream argument when launching bilateral filter kernel	14 days ago
cudalegacy	Some changes to support mingw-w64	a month ago
cudaobjdetect	changed hog to work with variable parameters and changed the hog samp…	2 months ago
cudaoptflow	Fixes namespace error on cudaoptflow	2 months ago
cudastereo	Adding support for WinRT(WinPhone 8/8.1 and Win Store) via CMake 3.1	7 months ago
cudawarping	Cast some image coordinates and sizes to double.	5 months ago
cudev	add opencv_test_cudev to installation package	3 months ago
features2d	IPPInitSingelton was added to contain IPP related global variables;	a day ago
flann	fixed uninitialized memory writing/reading in flann	19 days ago
hal	Warning fix	18 days ago
highgui	Update window_w32.cpp	12 days ago
imgcodecs	adding new flags to imread to load image reduced	5 days ago
imgproc	IPPInitSingelton was added to contain IPP related global variables;	a day ago
java	fix Android camera datarace (mCameraFrameReady)	10 days ago
ml	Merge pull request #5346 from art-programmer:art-programmer-patch-1	11 days ago
objdetect	IPPInitSingelton was added to contain IPP related global variables;	a day ago
photo	typos in comments	4 months ago
python	Fixing typo in variable name.	2 months ago
shape	Python support	7 months ago
stitching	fix cyclic deps error (world,shared)	3 months ago
superres	superres: Fix return value VideoFrameSource_GPU	4 months ago
ts	fix perf tests	10 days ago
video	ocl: workaround for getUMat()	23 days ago
videoio	Merge pull request #5371 from Dikay900:ports_to_master	11 days ago
videostab	Added configuration changes enabling videoio WinRT support.	5 months ago
viz	Added new functionalities to viz module	2 months ago
world	fix tests build (win,shared,world)	4 months ago
CMakeLists.txt	world fix	a year ago

Overall

- MoveIt! is awesome
- Successful because it is easy for beginners
- Needs many more features and improvements
- Stability (stagnation) should not be the #1 focus
- Please contribute!

Q & A 5min

Do you like to MoveIt MoveIt?

Thanks to Mike Ferguson, Sachin Chitta, Ioan Sucan, Shaun Edwards, Jon Bohren, Conor Brew, Acorn Pooley, Dave Hershberger, Chris Lewis, Jorge Nicho, Ben Chretien, Adolfo Rodriguez. Kei Okada, Stefan Kohlbrecher, and many more...