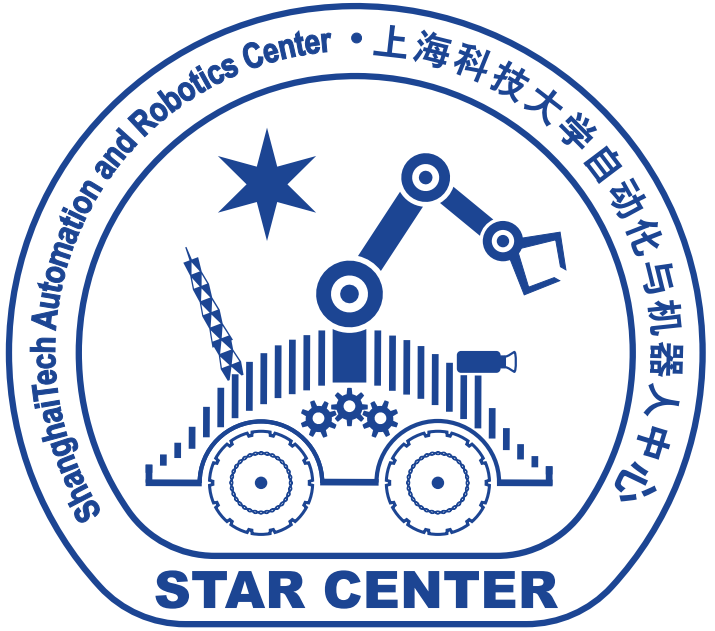
上海科技大学
ShanghaiTech University

CS289: **Mobile Manipulation Fall 2025**

Sören Schwertfeger

ShanghaiTech University



Outline

- Admin
- Planning
- OMPL
- MoveIt

ADMIN

Project

- 1 credit point!
- Work in groups, min 2 students, max 3 students – maybe 4 students.

- ~~Next lecture: Topics will be proposed…~~
  - ~~You can also do your own topic, but only after approval of Prof. Schwertfeger~~
    - ~~Prepare a short, written proposal till next Tuesday!~~
  - ~~Choose/ suggest a topic that is in line with your graduate research!~~
  - ~~We are flexible w.r.t. the topics – as long as they are involved with manipulation or complex mobile robotics.~~

- One graduate student from my group will co-supervise your project
- Weekly project meetings!

- Oral ”exams” to evaluate the contributions of each member
- No work on project => bad grade of fail

Grading

- Grading scheme is not 100% fixed
- Approximately:
  - Lecture: 60%
    - Homework: 30%
    - Final: 30%
  - Project: 40%
    - Project Proposal: 5%
    - Weekly project meetings: 10%
    - Final Report: 10%
    - Final Demo: 10%
    - Final Webpage: 5%

Paper Presentation

- Choose one paper from ICRA or IROS which is relevant to your project!
  - ICRA: https://ieeexplore.ieee.org/xpl/conhome/1000639/all-proceedings (any year!)
  - IROS: https://ieeexplore.ieee.org/xpl/conhome/1000393/all-proceedings (any year!)
  - Only full papers (6 or more pages) are allowed; no workshop papers

- **Send TA an email with your paper – no double papers are allowed – first come first serve!**

- Present the paper as if it were your own work!
- Front page: Name of the Paper; Full citation of the paper; **Your name in Pinyin; Your email address**
- Last slide: **ONE** slide about how this paper is relevant to your project.
- Your presentation has to be professional – not cute…
- **No videos/ no animations allowed!**
- **Put slide numbers in your presentation!**
- Submit pdf or ppt to the paper repository till Monday, Nov 03 22:00 to repo! Late submissions (or if you come with the ppt/ pdf to the presentation time) will receive a **flat 33% loss of points**!
- Presentations in 2 (or 3) slots – during lecture slots Nov 04 & 06.
- 10 minute presentation plus 1 minute project relevance plus 2 minutes questions
  - Do not rush your presentation! Better present less items more slowly!
  - 10 minute presentation => 8 – max. 12 slides
  - Maybe have a slide towards the end that you can skip if you run out of time.
  - Give a test presentation to your friends beforehand!
- Finish early for practicing – don’t learn by heart.

Grading of the Presentation

- 10 %: Your basic understanding/ knowledge about the paper you present
- 20 %: Presentation timing (plus or minus one minute is ok) – no rushing – good speed!
- 10 %: Correct written English in presentation:
  - No complete sentences, no grammatical or spelling mistakes
- 10 %: Good structure of presentation:
  - Depends on the type of paper, how much time you have, how long you need to present the main achievement.
  - For example: outline, introduction/ motivation, **problem statement**, state of the art, **approach**, experiments, **results**, **conclusion**, outlook
- 20 %: Clarity of written presentation
- 10 %: Good presentation style:
  - Interact with audience: look at the whole room (not just your slides, notes, or the back of the room)
  - Present the paper – do not read (or repeat the learned) speech from a prepared text
  - Use the presentation as visual aid – not as your tele-prompter to read from
  - Move your body – do not stand frozen at one place
- 10 %: Answering the questions
  - Questions have to be asked and answered in English – Chinese can be used for clarification
- 10 %: Asking questions to other students!
- **Not** scored: Your English skill

Project Proposal (1)

- **Title:** Find a nice, catchy title for your project
- **Abstract**: A short abstract/ summary what the project is about
- **Introduction**: general description & Motivation
- **State of the Art**: Literature & open-source-ROS packages
- Per team member:
  - present and cite three papers with just three or four sentences
  - present in more detail one further paper relevant to your project. Describe it with at least 1/3rd of a page.
  - present in detail one open source ROS package relevant to your project. At least 1/3rd of a page
  - => about one page per team member – => 3 pages for 3 person team

Project Proposal (2)

- **System Description**
- **System Evaluation**: Describe how you want to test your system.
  - Experiments & how to measure their success
- **Work Plan**: Define some mile stones.
  - Possible phases: Algorithm design, implementation, testing, evaluation, documentation – some of those things can also happen in a loop (iteration).
  - Deliverables of Project:
    - Proposal (this document)
    - Final demo
    - Final Report
    - Website
- **Conclusions**: Short summary and conclusions

Project Proposal (3)

- Important dates:
- **Oct 10, 22:00: due date for the proposal**
- **Jan 13, 22:00: due date for the final report**
  Jan 13: due date for the final demo & website

- Parts of proposal go into the final project report.
- Please don’t forget to take **pictures and videos** when testing your system!

- In English! Using LaTeX!
- Put sources and PDF in git.

- **Additional task:** In glit/ gitlab: “Readme.txt” with:
  - Team Name and Members; email addresses
  - Documentation and how to’s regarding your project.

Project Suggestion: Fetch

- We have a kitchen now in the lab!
- Use fetch to open cabinet or open drawer and pick some object

Take object with 3-finger Kinova hand from Parallel Gripper

Application: Hand-over of objects from one robot to another…

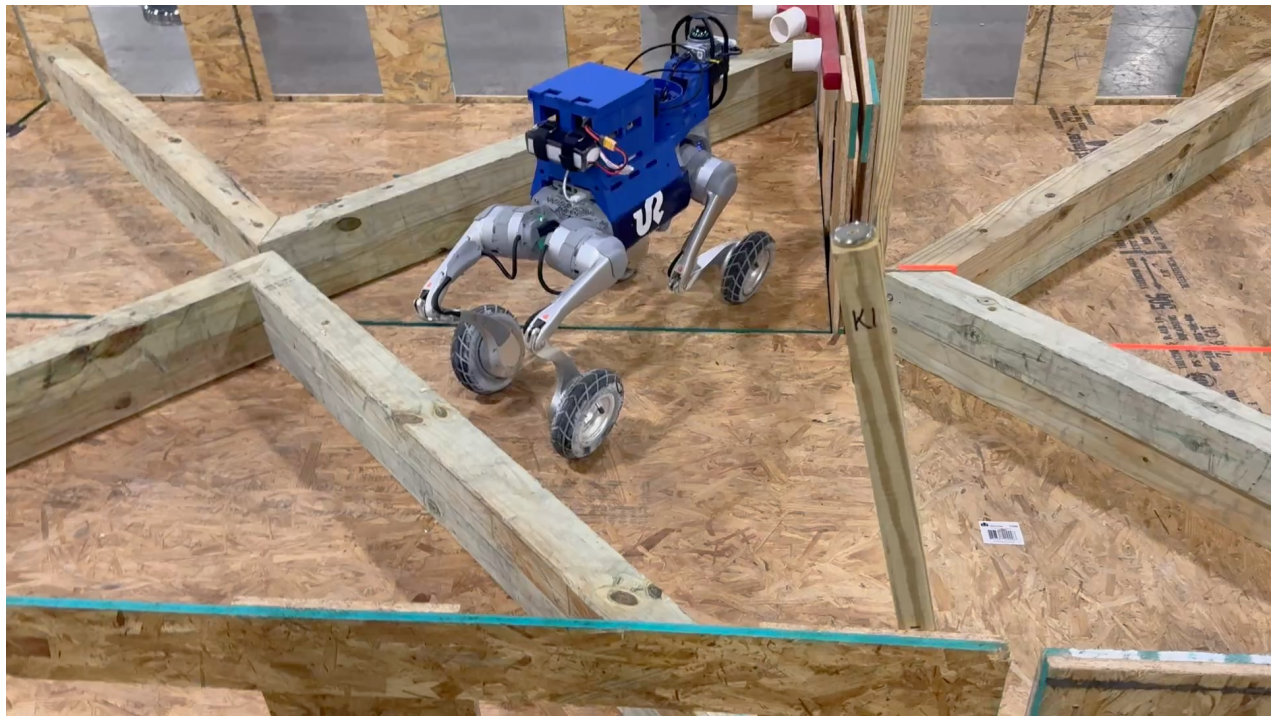
Jackal Mobile Platform with Kinova Robot Arm

Do something cool with this robot…

Husky Mobile Platform with
AUBO Robot Arm

Quadruped Robot Challenges & RoboCup

- Do some work regarding those competitions!



QRC Ideas:

- QRC P1: D1 robot arm on go2 or go2-w: Hardware integration and remote control for Inspection and maybe touch and/ or insert. MoveIt! cartesian control. Maybe also MoveIt! planning.
- QRC P2: object detection & localization for QRC/ RoboCup manipulation targets (linear & omni star) & automatic MoveIt planning (datasets/ gazebo). RGB-D data. Very end test on hardware. (needs some group to do QRC P1)
- QRC P3: no arm! Use whole body control (RL) on flat ground to position camera (front and/ or omni) at the right pose in front of target (can use QRC P2 detection). Including first cartesian manual control: camera up/ down/ left/ right.
- QRC P4: Use whole body control with a 3DoF arm, similar to P3. In simulation only?
- QRC P5: Model some/ several QRC/ RoboCup lanes in Isaac Lab (so others can do RL). Use it as a simulator: can run a robot model with a joystick through it. Including: have lots of parts individually, such that we can later use DR so vary the shape of the arena.
- QRC P6: Try autonomy navigation on the easiest lane (low K-rails flat): LiDAR localization, waypoint goal, planning, control.
- QRC P7: For RoboCup Maze: Use RGB/ RGB-D data and/ or map of Maze exploration with VLM to give a report/ list of detected things/ what is out of place/ what is interesting w.r.t. a rescue scenario? (Kind of real time (within 2 minutes))

Hospital Perception Projects

- STAR Center may want to use a mobile robot in the new hospital

- Use sensors to remotely observe human status:
  - IR camera for temperature
  - RGB camera for pulse and breathing frequency
  - Sensitive, mutli-channel microphones to locate sounde source (breathing!?)
  - ~~Hyperspectral camera for ???~~
  - ~~Polarized camera for ???~~
  - Human robot interaction to ask how the patient is feeling

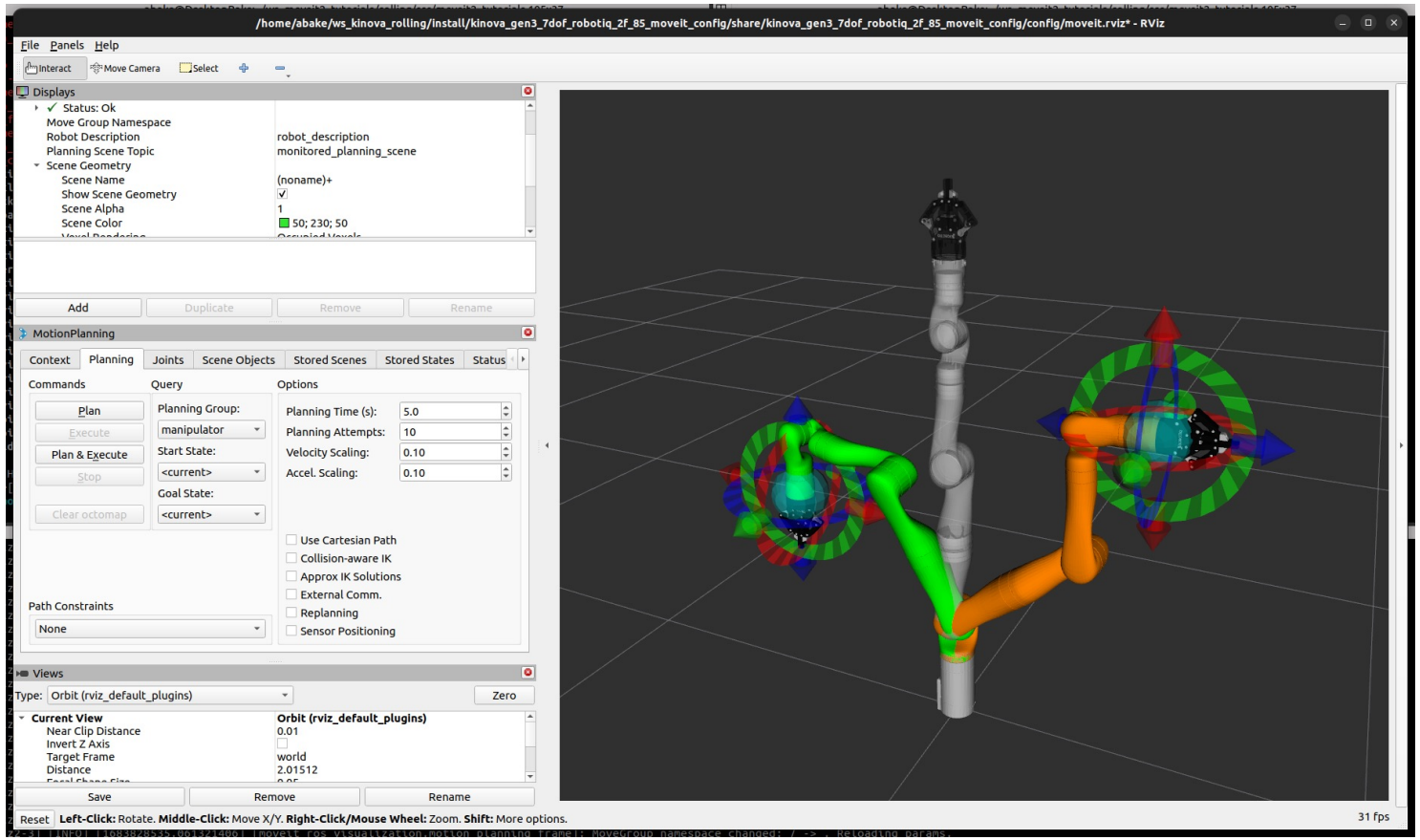
- Several projects with selection of those topics possible

Project Suggestion: Draw with Sand or Water

- Build and program such a robot …
- Quite difficult …
- But cool …
- Bigger group (with sub-tasks) allowed

MOVEIT

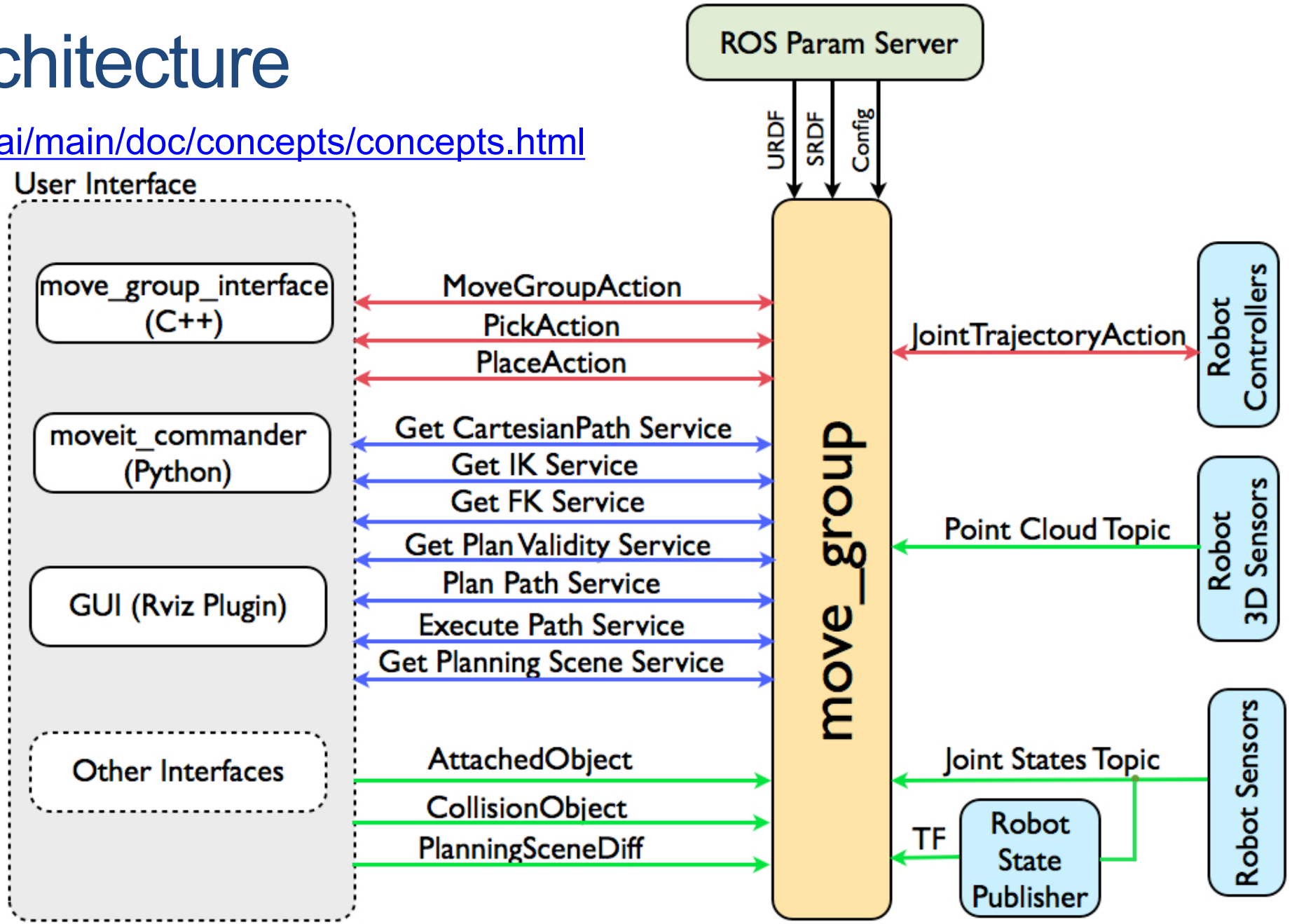
MoveIt!



- Software to Control & Plan with robot arms
- Part of ROS
- https://moveit.picknik.ai/
- First let’s explore how to tell MoveIt! what kind of arm you have…

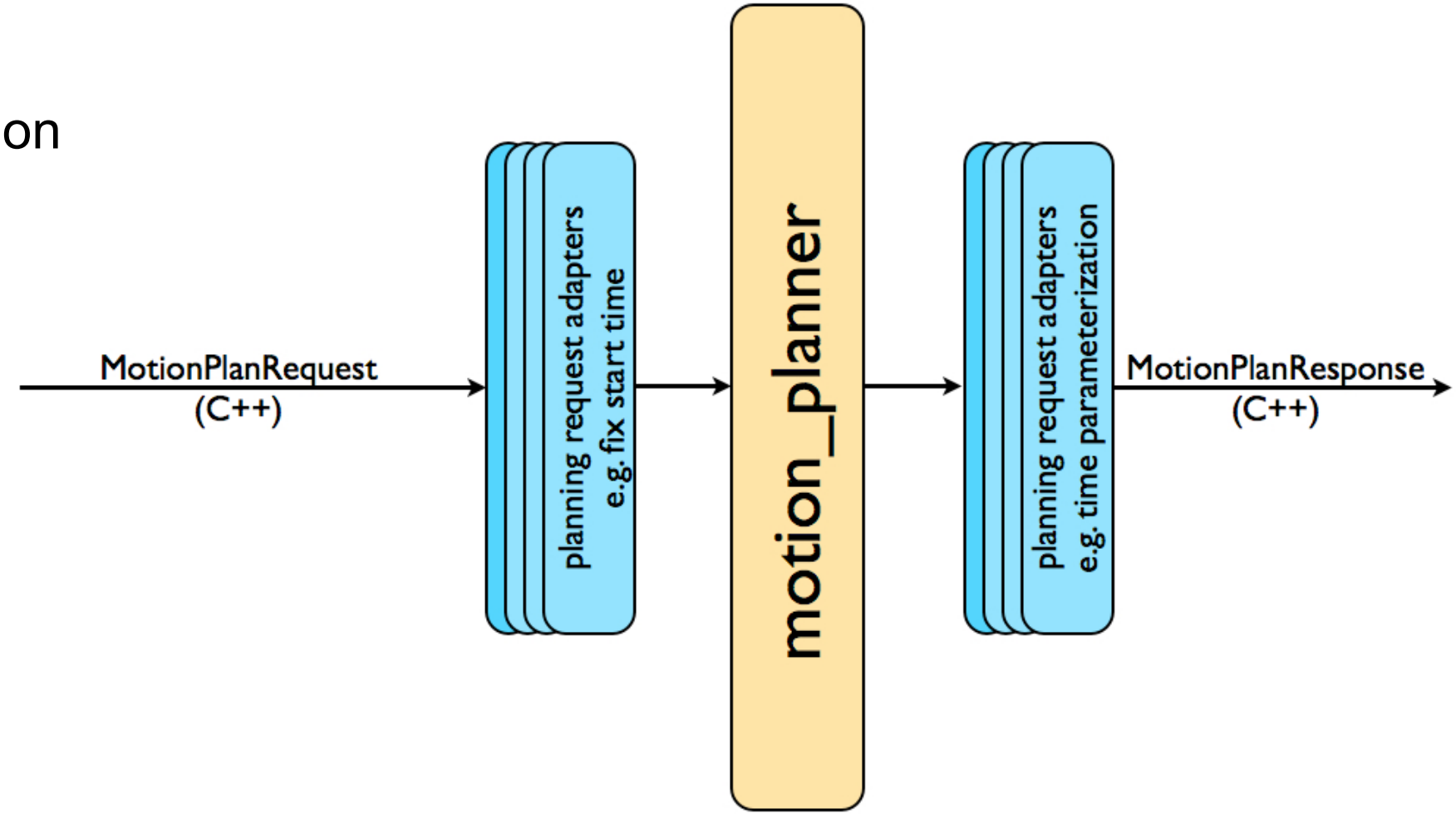
System Architecture

https://moveit.picknik.ai/main/doc/concepts/concepts.html



Motion Planning

- Mainly:
- OMPL (Open Motion Planning Library)

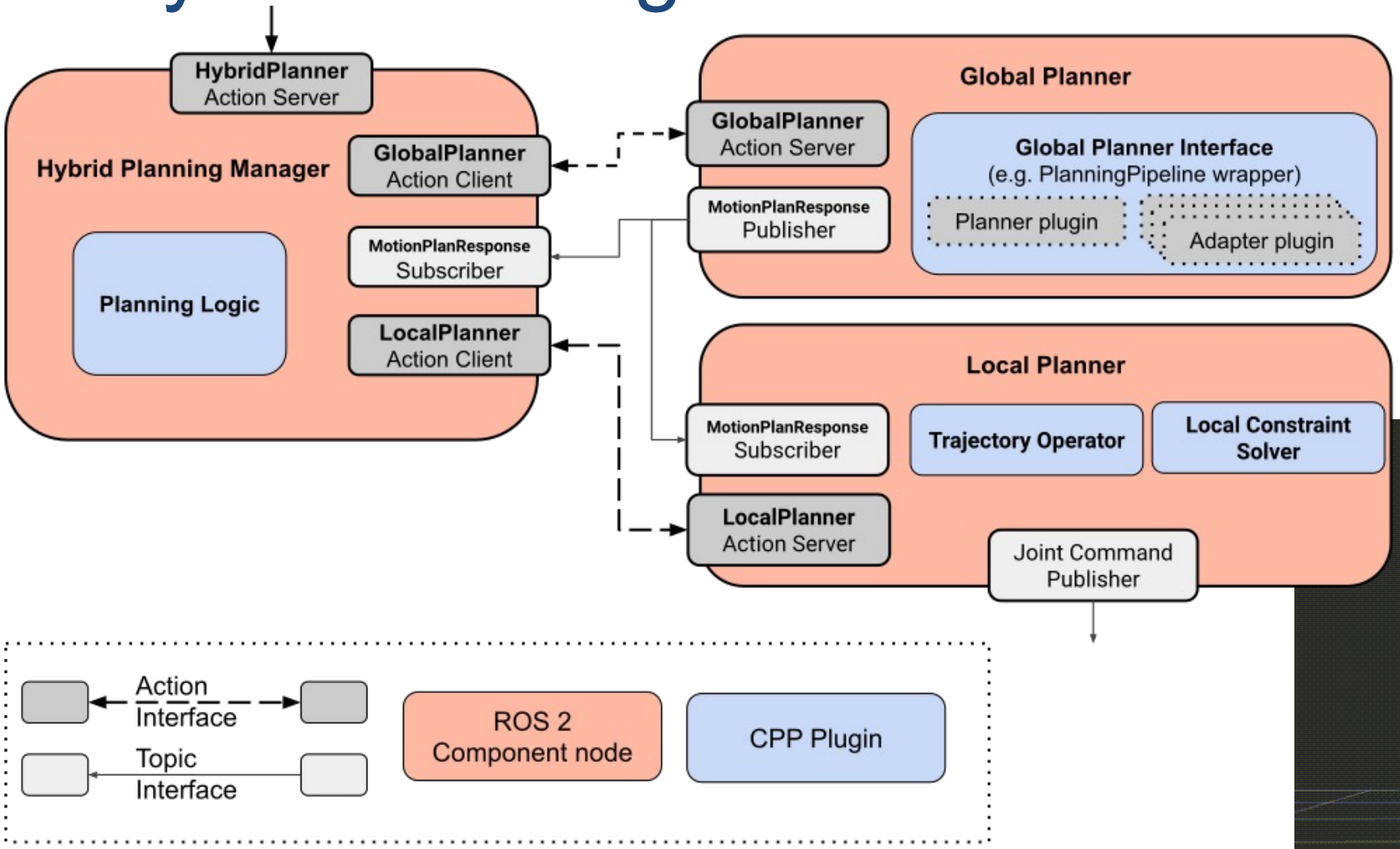


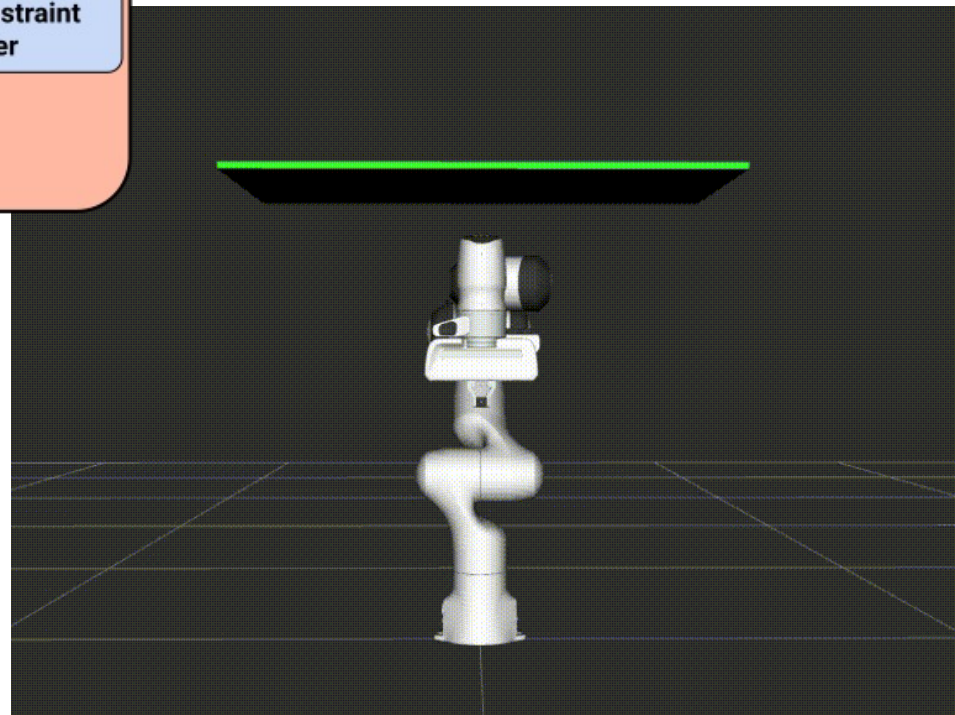
Hybrid Planning Global and Local

- Online motion planning: The global planner creates an initial global solution and continuously optimizes it. Simultaneously, the local planner executes the reference trajectory and blends updated trajectory segments into it.
- Reactive Motion: The global planner is used to fix invalidated solutions (replanning) while the local planner slows down or halts before collisions
- Adaptive Motion: The local planner is used to adapt a global solution to dynamic conditions like keeping steady tool contact with an uneven surface

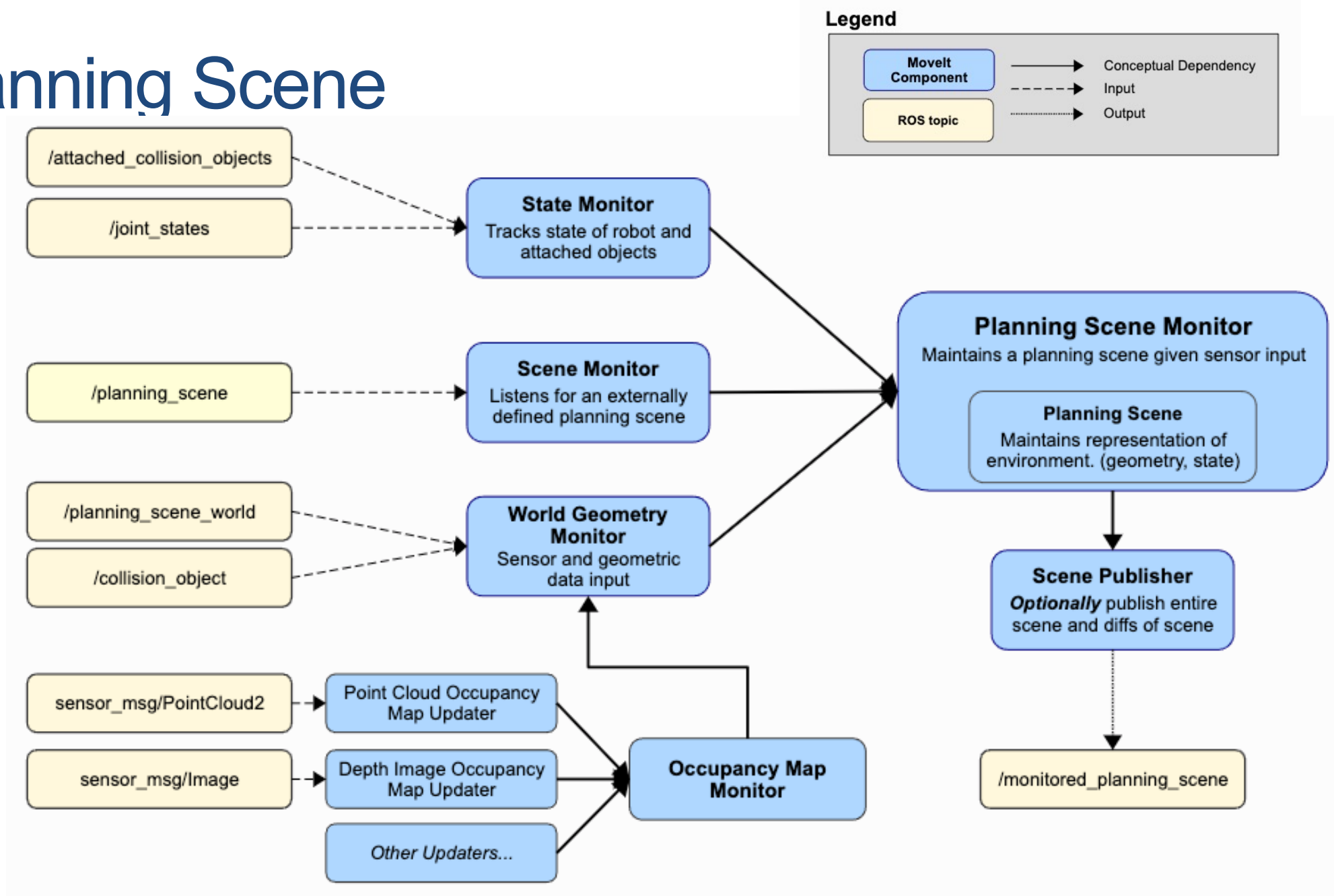
Global Planner	Local Planner
• Solve global solution trajectory • Optimize trajectory path (continuously)	• Follow global reference trajectory • Solve local problem constraints • May process sensor input • Optimize solution locally • Compute controller commands
• Complete • No restricted computation time • Not real-time safe • Not necessarily deterministic	• Can get stuck in local minima • Low computation time • Realtime-safe (depends on solver) • Deterministic
• OMPL planner • STOMP • TrajOpt • Cartesian motion planner • Pilz Industrial Motion Planner • MTC	• IK solver, Jacobian • Potential field planner • Trajectory optimization algorithm • Model Predictive Control (MPC) • Sensor-based Optimal Control

Hybrid Planning: Global and Local





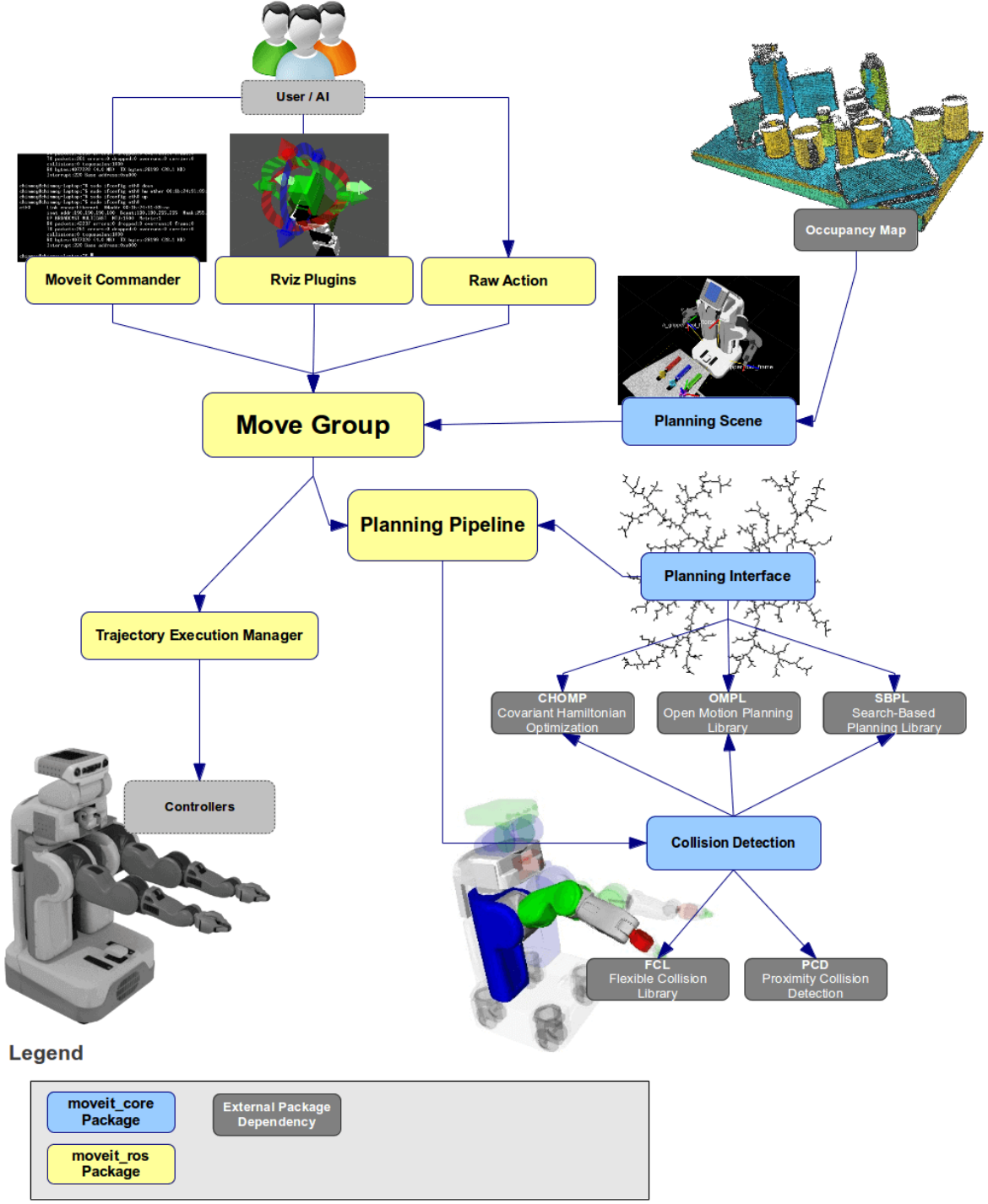
Planning Scene



MoveIt!

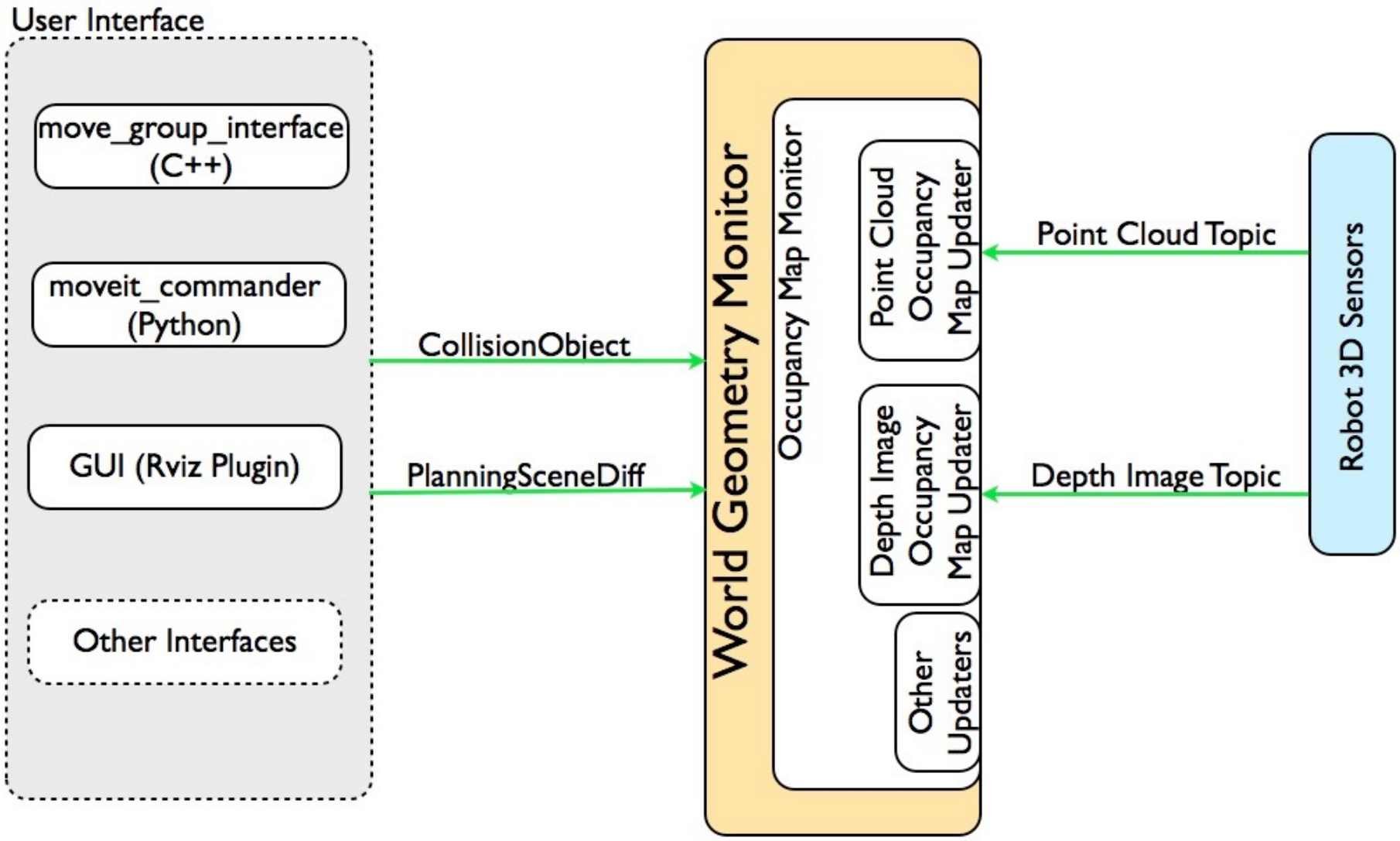
- Configuration:

```
my_robot_moveit_config
    config/
        kinematics.yaml
        joint_limits.yaml
        *_planning.yaml
        moveit_controllers.yaml
        moveit_cpp.yaml
        sensors_3d.yaml
        ...
    launch/
    .setup_assistant
    CMakeLists.txt
    package.xml
```



3D Perception & Collision Checking

- Octomap



Simple C++

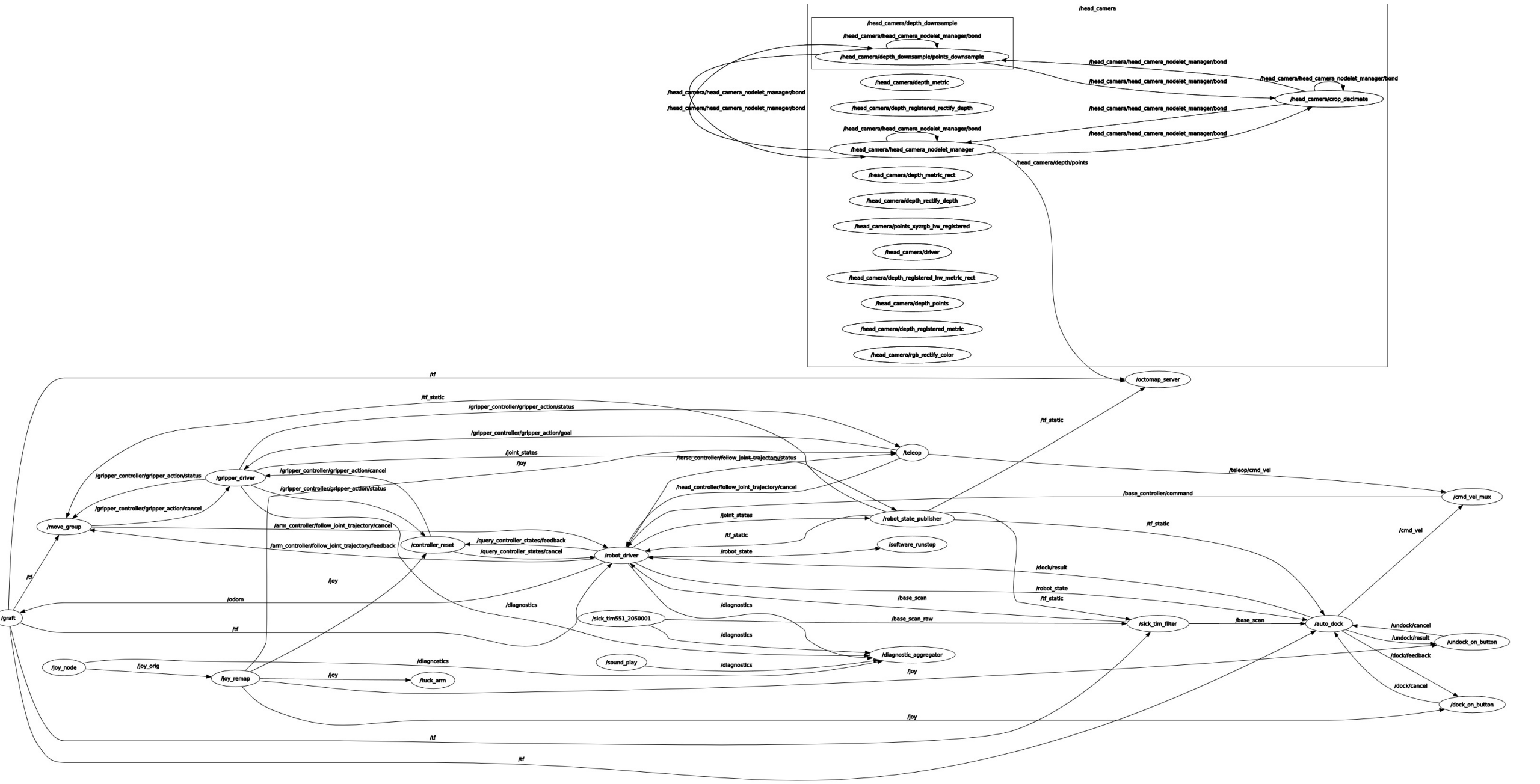
```cpp
// Create the MoveIt MoveGroup Interface
using moveit::planning_interface::MoveGroupInterface;
auto move_group_interface = MoveGroupInterface(node, "manipulator");

// Set a target Pose
auto const target_pose = []{
  geometry_msgs::msg::Pose msg;
  msg.orientation.w = 1.0;
  msg.position.x = 0.28;
  msg.position.y = -0.2;
  msg.position.z = 0.5;
  return msg;
}();
move_group_interface.setPoseTarget(target_pose);

// Create a plan to that target pose
auto const [success, plan] = [&move_group_interface]{
  moveit::planning_interface::MoveGroupInterface::Plan msg;
  auto const ok = static_cast<bool>(move_group_interface.plan(msg));
  return std::make_pair(ok, msg);
}();

// Execute the plan
if(success) {
  move_group_interface.execute(plan);
} else {
  RCLCPP_ERROR(logger, "Planning failed!");
}
```

EXAMPLE

Fetch Robot; ROS 1

- Examples from running robot

- Without localization, SLAM, mapping navigation, task or mission planning


/head_camera
/head_camera/depth_downsample
/head_camera/head_camera_nodelet_manager/bond
/head_camera/depth_downsample/points_downsample
/head_camera/depth_metric
/head_camera/depth_registered_rectify_depth
/head_camera/head_camera_nodelet_manager
/head_camera/depth_metric_rect
/head_camera/depth_rectify_depth
/head_camera/points_xyzrgb_hw_registered
/head_camera/driver
/head_camera/depth_registered_hw_metric_rect
/head_camera/depth_points
/head_camera/depth_registered_metric
/head_camera/rgb_rectify_color
/head_camera/crop_decimate
/head_camera/depth/points
/octomap_server
/tf
/tf_static
/gripper_controller/gripper_action/status
/gripper_controller/gripper_action/goal
/gripper_controller/gripper_action/cancel
/gripper_controller/gripper_action/cancel
/joint_states
/joy
/torso_controller/follow_joint_trajectory/status
/head_controller/follow_joint_trajectory/cancel
/teleop
/teleop/cmd_vel
/cmd_vel_mux
/base_controller/command
/gripper_driver
/move_group
/arm_controller/follow_joint_trajectory/cancel
/arm_controller/follow_joint_trajectory/feedback
/controller_reset
/query_controller_states/feedback
/query_controller_states/cancel
/robot_driver
/robot_state_publisher
/software_runstop
/robot_state
/cmd_vel
/graft
/odom
/diagnostics
/sick_tim551_2050001
/base_scan_raw
/sick_tim_filter
/base_scan
/auto_dock
/dock/result
/undock/cancel
/undock/result
/undock_on_button
/dock/feedback
/dock/cancel
/dock_on_button
/joy_node
/joy_orig
/joy_remap
/tuck_arm
/sound_play
/diagnostic_aggregator

Nodes

- 35 different nodes running
- Without any navigation for the mobile base! (at least +15)

```
/auto_dock
/cmd_vel_mux
/controller_reset
/diagnostic_aggregator
/dock_on_button
/graft
/gripper_driver
/head_camera/crop_decimate
/head_camera/depth_downsample/points_downsample
/head_camera/depth_metric
/head_camera/depth_metric_rect
/head_camera/depth_points
/head_camera/depth_rectify_depth
/head_camera/depth_registered_hw_metric_rect
/head_camera/depth_registered_metric
/head_camera/depth_registered_rectify_depth
/head_camera/driver
/head_camera/head_camera_nodelet_manager
/head_camera/points_xyzrgb_hw_registered
/head_camera/rgb_rectify_color
/joy_node
/joy_remap
/move_group
/octomap_server
/robot_driver
/robot_state_publisher
/rosout
/rviz_1713876364443101616
/sick_tim551_2050001
/sick_tim_filter
/software_runstop
/sound_play
/teleop
/tuck_arm
/undock_on_button
```

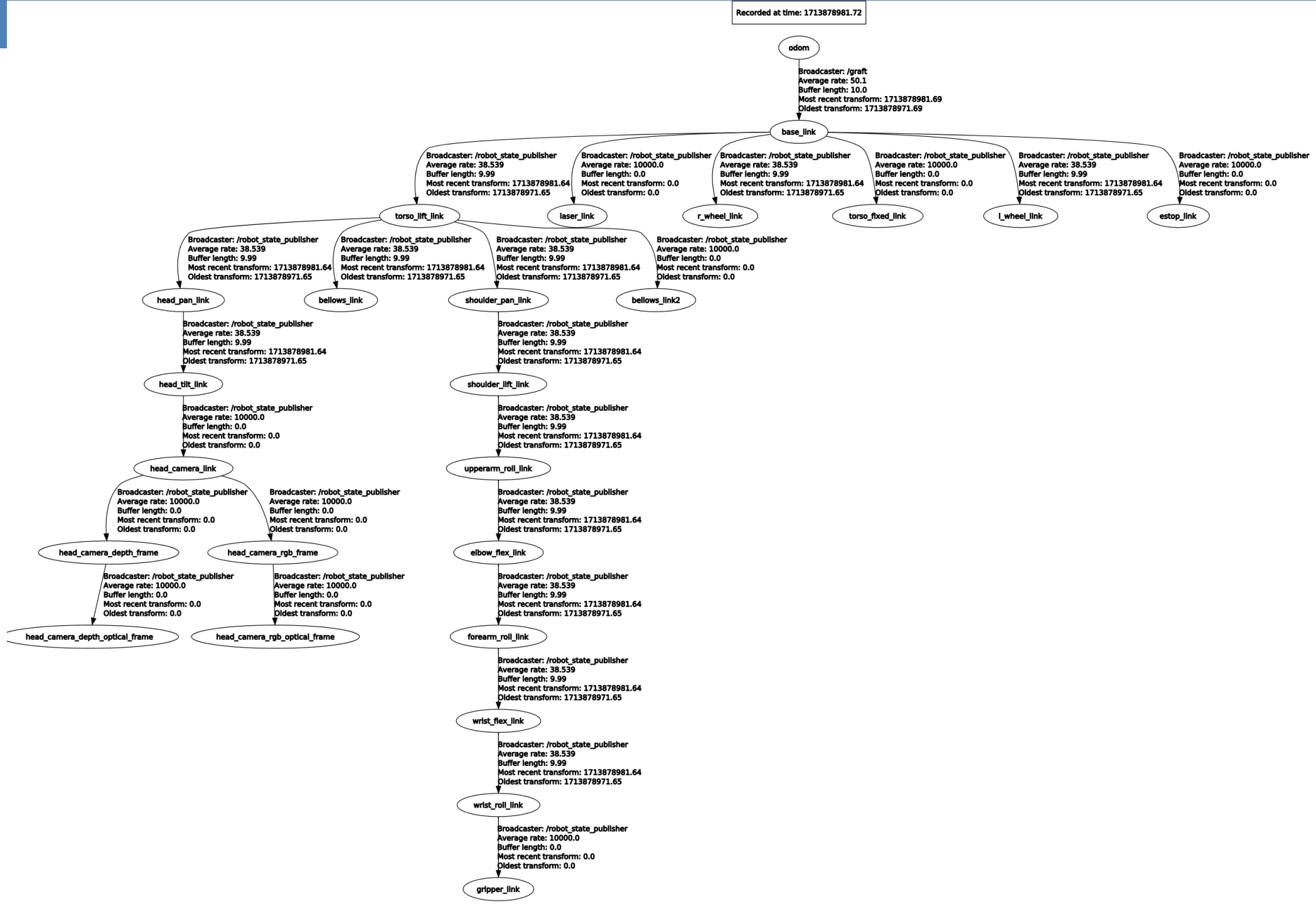
Topics

- 311 different topics
- w/o navigation!

- Each action has topics:
  cancel; feedback; goal; result; status

```
/arm_controller/cartesian_twist/command
/arm_controller/follow_joint_trajectory/cancel
/arm_controller/follow_joint_trajectory/feedback
/arm_controller/follow_joint_trajectory/goal
/arm_controller/follow_joint_trajectory/result
/arm_controller/follow_joint_trajectory/status
/arm_with_torso_controller/follow_joint_trajectory/cancel
/arm_with_torso_controller/follow_joint_trajectory/feedback
/arm_with_torso_controller/follow_joint_trajectory/goal
/arm_with_torso_controller/follow_joint_trajectory/result
/arm_with_torso_controller/follow_joint_trajectory/status
/attached_collision_object
/base_controller/command
/base_scan
/base_scan_no_self_filter
/base_scan_raw
/base_scan_tagged
/battery_state
/charge_lockout/cancel
/charge_lockout/feedback
/charge_lockout/goal
/charge_lockout/result
/charge_lockout/status
/clicked_point
/cmd_vel
/cmd_vel_mux/selected
/collision_object
/diagnostics
/diagnostics_agg
/diagnostics_toplevel_state
/dock/cancel
/dock/feedback
/dock/goal
/dock/result
```

tf tree


Recorded at time: 1713878981.72
odom
Broadcaster: /graft
Average rate: 50.1
Buffer length: 10.0
Most recent transform: 1713878981.69
Oldest transform: 1713878971.69
base_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
torso_lift_link
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
laser_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
r_wheel_link
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
torso_fixed_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
l_wheel_link
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
estop_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
head_pan_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
head_tilt_link
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
head_camera_link
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
head_camera_depth_frame
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
head_camera_depth_optical_frame
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
head_camera_rgb_frame
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
head_camera_rgb_optical_frame
bellows_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
shoulder_pan_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
shoulder_lift_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
upperarm_roll_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
elbow_flex_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
forearm_roll_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
wrist_flex_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
wrist_roll_link
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0
gripper_link
Broadcaster: /robot_state_publisher
Average rate: 38.539
Buffer length: 9.99
Most recent transform: 1713878981.64
Oldest transform: 1713878971.65
bellows_link2
Broadcaster: /robot_state_publisher
Average rate: 10000.0
Buffer length: 0.0
Most recent transform: 0.0
Oldest transform: 0.0

SENSORS FOR COLLISION CHECKING

Sensor

- Requirements:
  - Range information
  - Dense data
  - Ideally: also have RGB data to combine computer vision with 3D range sensing

- Main sensor approaches:
  - LiDAR
  - Time of flight cameras
  - Structured light
  - Stereo cameras

Intel RealSense L515

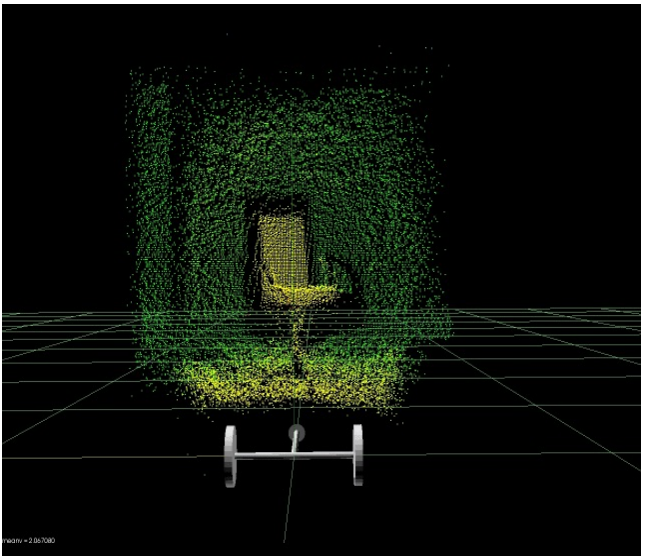
- 9m distance
- Depth: 1024 x 786 pixel @ 30Hz => 23mill pts per second
- RGB: 1920 × 1080 @ 30Hz
- Depth FOV: 70° × 55° (±2°)
- Weight: 100g
- With IMU

- Solid state laser with RGB camera

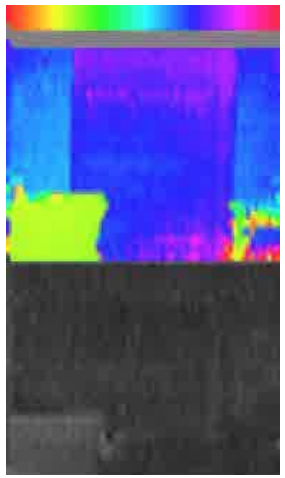
- Sensitive to ambient infrared light => Does NOT work outdoors!

3D Range Sensor: Time Of Flight (TOF) camera

- 3D information with high data rate (100 Hz)
- Compact and easy to manage
- High, non-uniform measurement noise
- High outlier rate at jump edges
- Wrap-around error (phase ranging)
- Sensitive to ambient infrared light => Does NOT work outdoors!



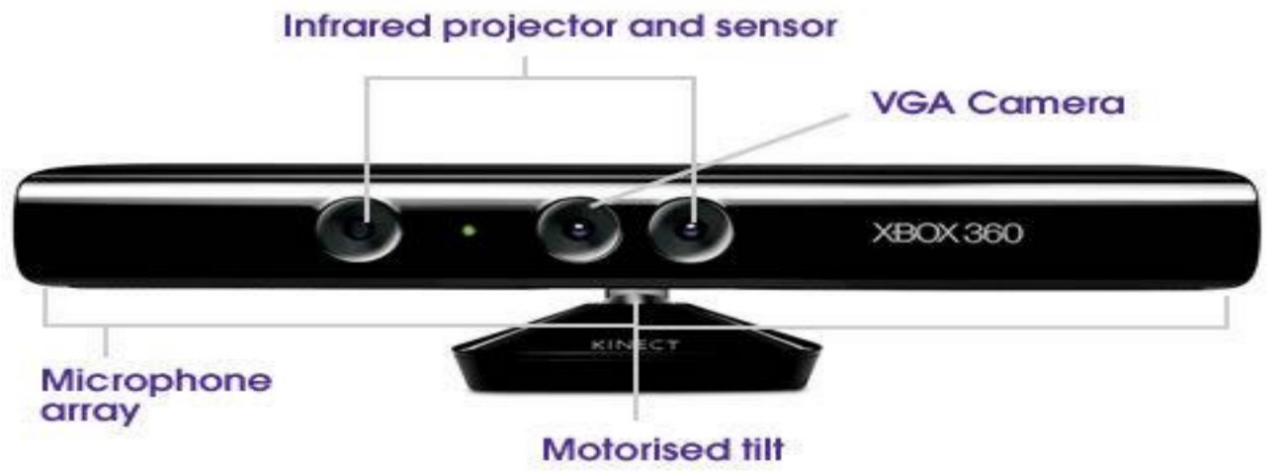
Swiss Ranger 3000
(produced by MESA)



- Kinect 2
- Resolution 1920x1080 pixels
- Field of view: 70 deg (H), 60 deg (V)
- Claimed accuracy: 1 mm
- Claimed max range: 6 meters

PrimeSense Cameras

- Devices: Microsoft Kinect and Asus Xtion
- Developed by Israeli company PrimeSense in 2010



- Components:
  - IR camera (640 x 480 pixel)
  - IR Laser projector
  - RGB camera (640 x 480 or 1280 x 1024)
  - Field of View (FoV):
    - 57.5 degrees horizontally,
    - 43.5 degrees vertically

IR Pattern

Sensitive to infrared light
=>
Does NOT work outdoors!

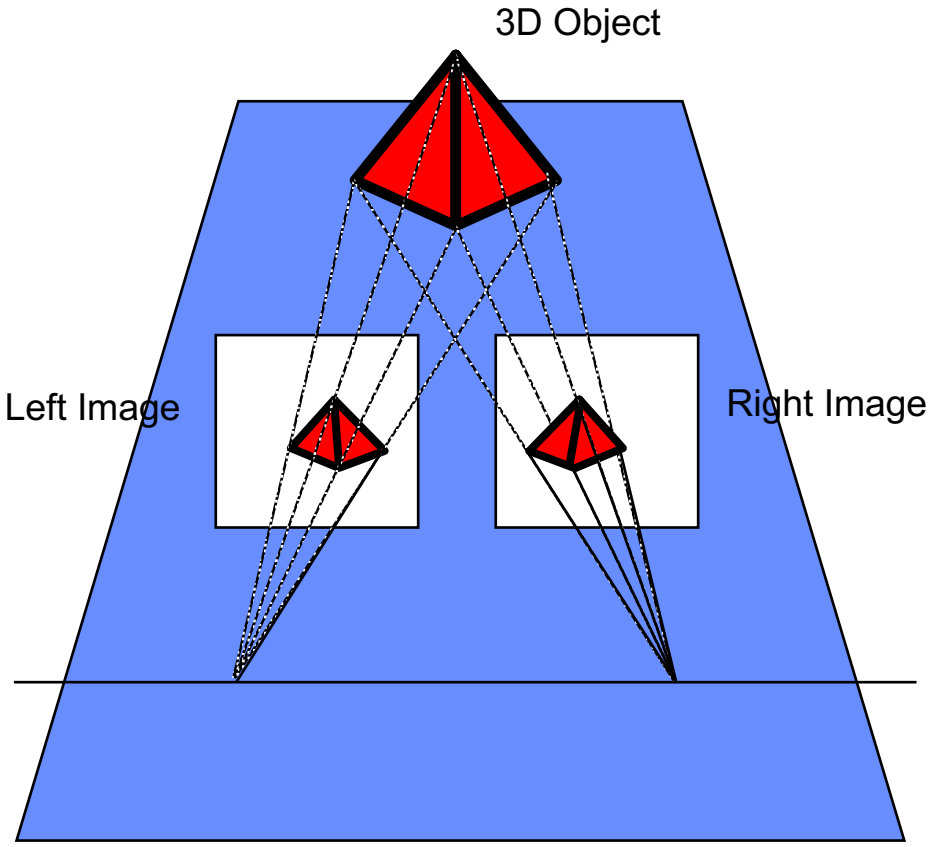
Depth Map

Microsoft Kinect: Depth Computation

- **Depth from Stereo**
  - The Kinect uses an infrared projector and an infrared sensor; it does not use its RGB camera for depth computation
  - The technique of analyzing a known pattern is structured light
  - The IR projector projects a pseudo-random pattern across the surface of the room.
  - The direction of each speckle of the patter is known (from pre calibration during manufacturing) and is hardcoded into the memory of the Kinect
  - By measuring the position of each speckle in the IR image, its depth can be computed

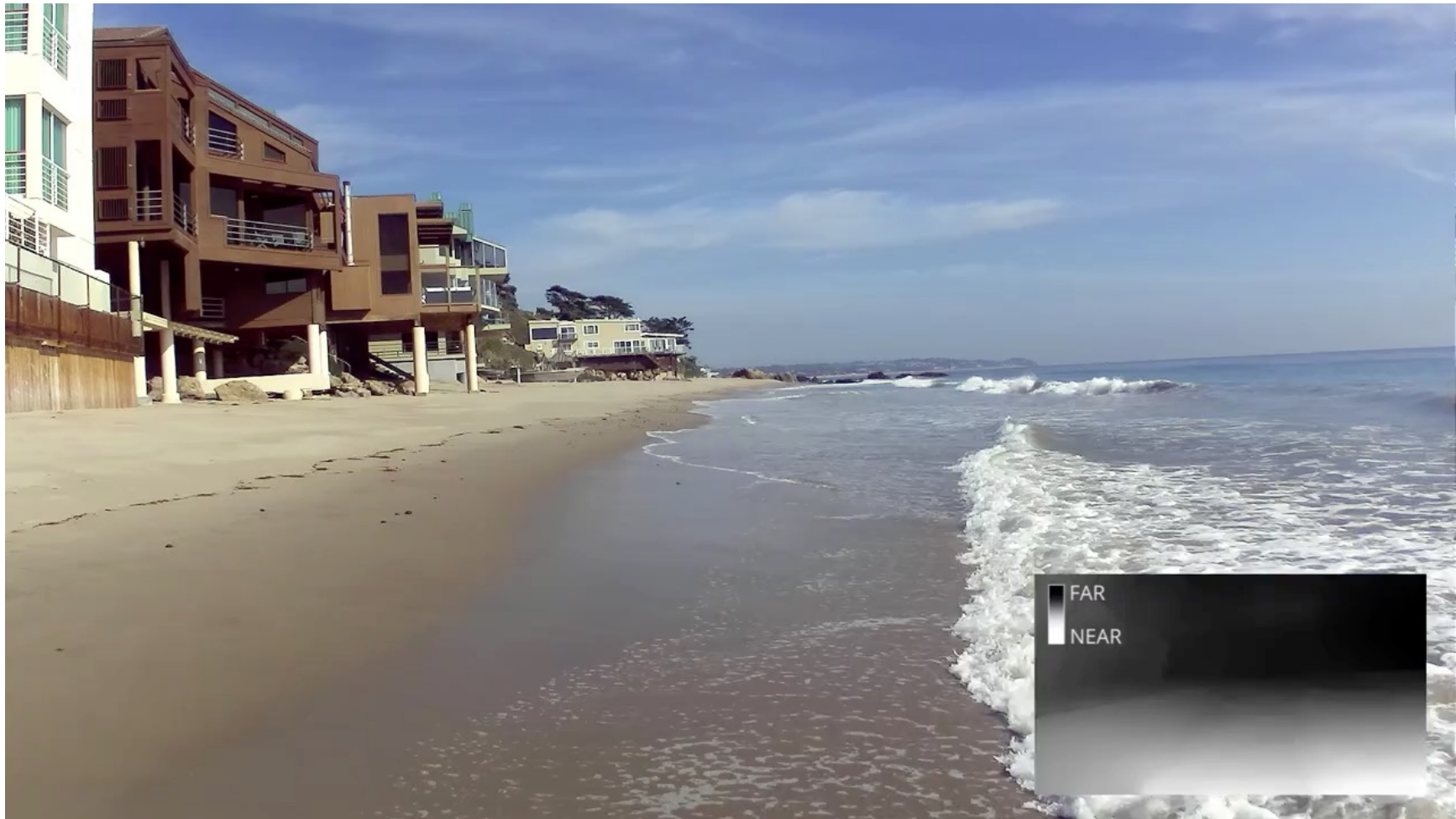
Stereo Cameras

- Theory will be covered in detail in Vision Lecture
- Estimate depth by using 2 cameras



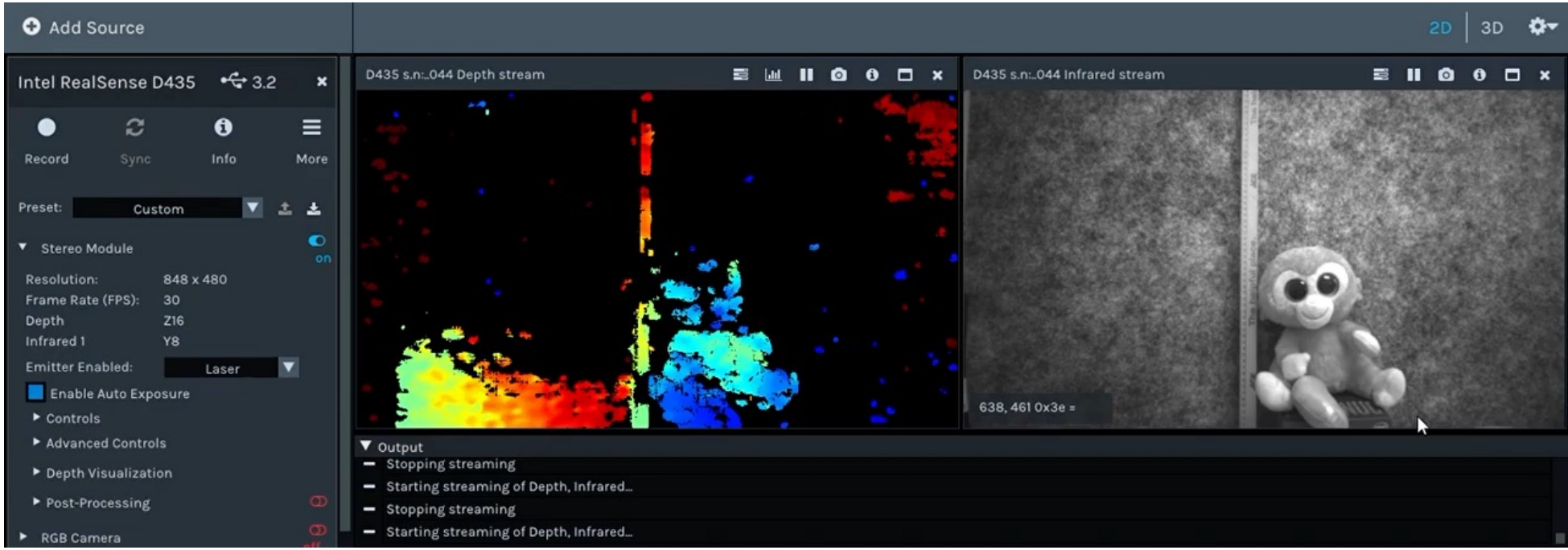
Stereo example: ZED camera

- Dual 4MP Camera @15Hz (lower resolution => higher fps)
- Up to 20m distance
- Passive Sensor
- Doesn’t work on single color surfaces (e.g. white wall)!



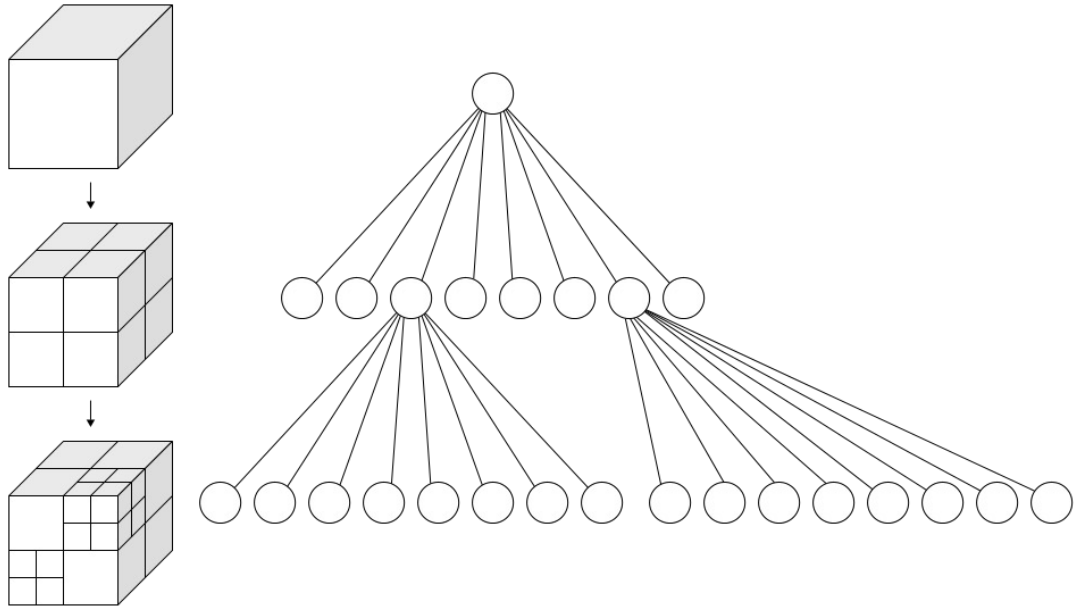
Intel RealSense D435

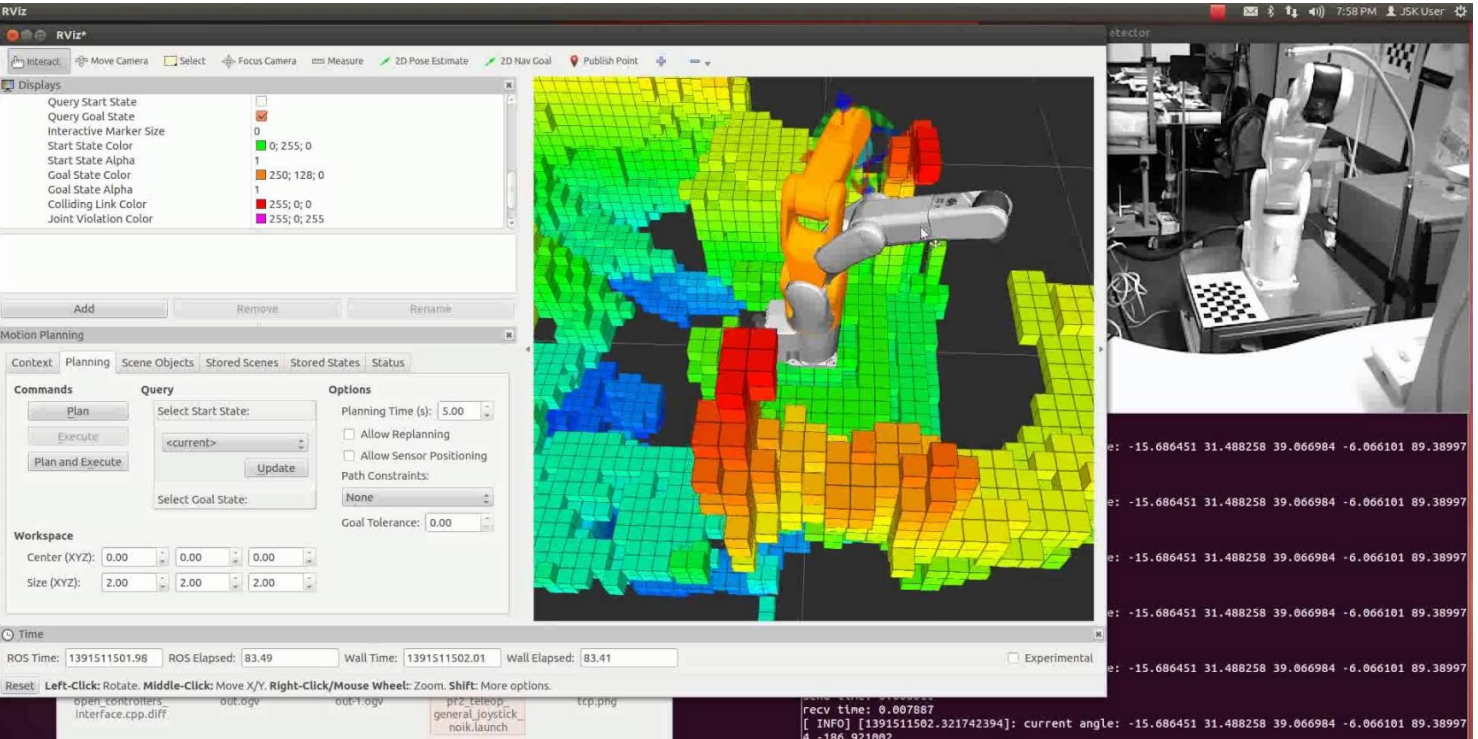
- Stereo Infrared – works indoors and outdoors
- Active pattern (e.g. for white wall) – only works indoors!
- Depth resolution: 1280 × 720
- Depth Field of View (FOV): 86° × 57° (±3°)
- RGB: 1920 × 1080
- Small and l ightweight
- With IMU



Octomap / Octree

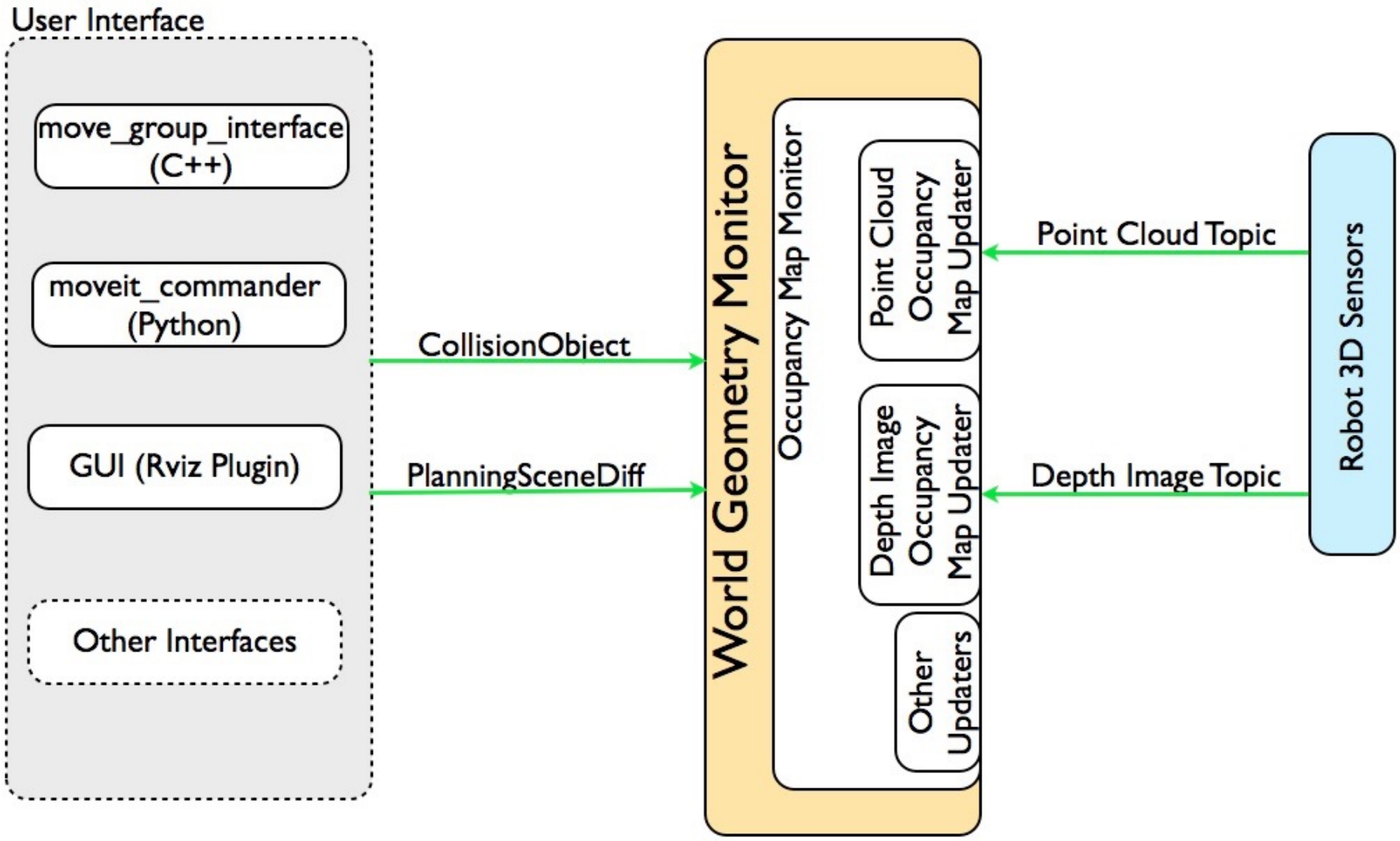
- Put 3d range sensor data in clever datastructure
- http://wiki.ros.org/octomap





3D Perception & Collision Checking

- Octomap



Mobile Manipulation

RViz

Interact | Move Camera | Select | Focus Camera | Measure | 2D Pose Estimate | 2D Nav Goal | Publish Point

Displays

- Global Options
  - Fixed Frame: /BASE
  - Background Color: 48; 48; 48
- Global Status: Ok
  - Fixed Frame: OK
- Grid
- MotionPlanning
  - Status: Ok
  - Robot Description: robot_description

Add | Remove | Rename

Motion Planning

Context | Planning | Scene Objects | Stored Scenes | Stored States | Status

Planning Library

OMPL <unspecified> Publish Current Scene

Warehouse

Host: 127.0.0.1 Port: 33829 Connect

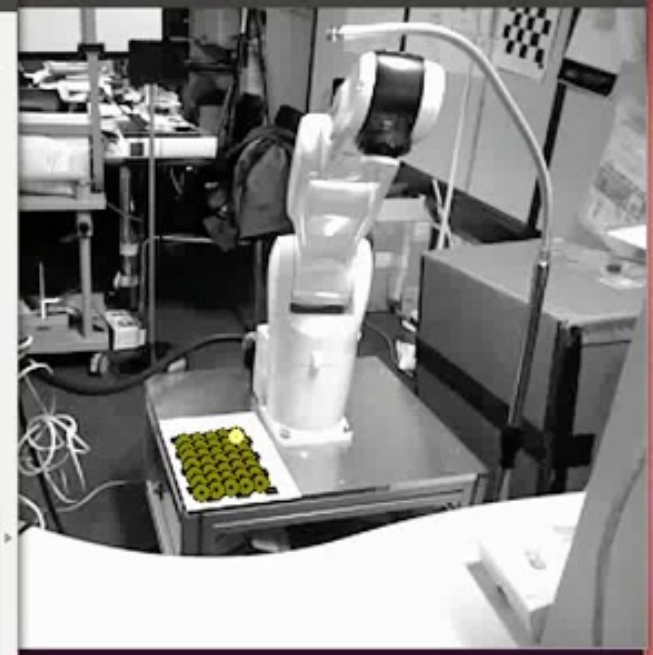
Kinematics

- [x] Use Collision-Aware IK
- [ ] Allow Approximate IK Solutions

Time

ROS Time: 1391511451.87 ROS Elapsed: 33.38 Wall Time: 1391511452.01 Wall Elapsed: 33.38 Experimental

Reset



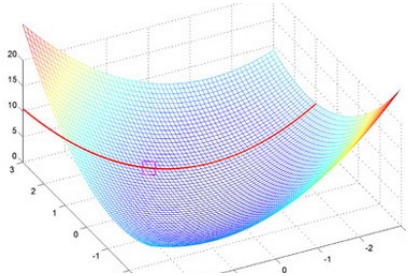
```
e: -15.686451 31.488258 39.066984 -6.066101 89.38997
e: -15.686451 31.488258 39.066984 -6.066101 89.38997
e: -15.686451 31.488258 39.066984 -6.066101 89.38997
e: -15.686451 31.488258 39.066984 -6.066101 89.38997
e: -15.686451 31.488258 39.066984 -6.066101 89.38997
recv time: 0.007818
[ INFO] [1391511452.321107105]: current angle: -15.686451 31.488258 39.066984 -6.066101 89.38997
4 -186.921002
send time: 0.000017
recv time: 0.007829
[ INFO] [1391511452.329157163]: current angle: -15.686451 31.488258 39.066984 -6.066101 89.38997
4 -186.921002
send time: 0.000023
recv time: 0.007888
/rviz_deuscontroller_9320_4971803640700373455
/state_publisher
jskuser@deuscontroller:~/ros/groovy/jsk-ros-pkg/
```

PLANNING

Path Planning: Overview of Algorithms

1. Optimal Control

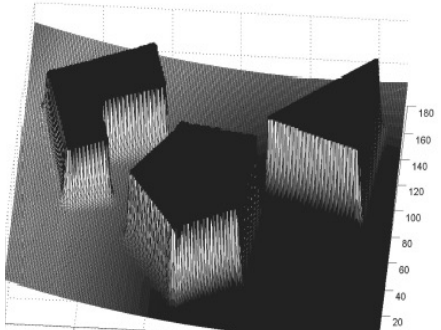
- Solves truly optimal solution
- Becomes intractable for even moderately complex as well as nonconvex problems



Source: http://mitocw.udsm.ac.tz

2. Potential Field

- Imposes a mathematical function over the state/configuration space
- Many physical metaphors exist
- Often employed due to its simplicity and similarity to optimal control solutions



3. Deterministic Graph Search

- Identify a set edges between nodes within the free space
- Breath/ Depth First Search
- Dijkstra
- A*, D*

4. Sampling-based Motion Planning

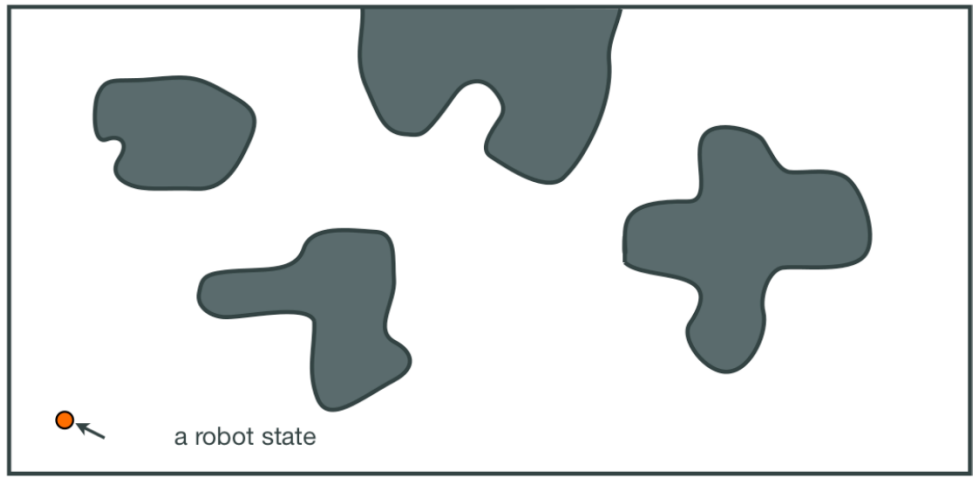
- Randomized Graph Search
- Rapidly exploring Random Tree RRT

Sampling-based Motion Planning

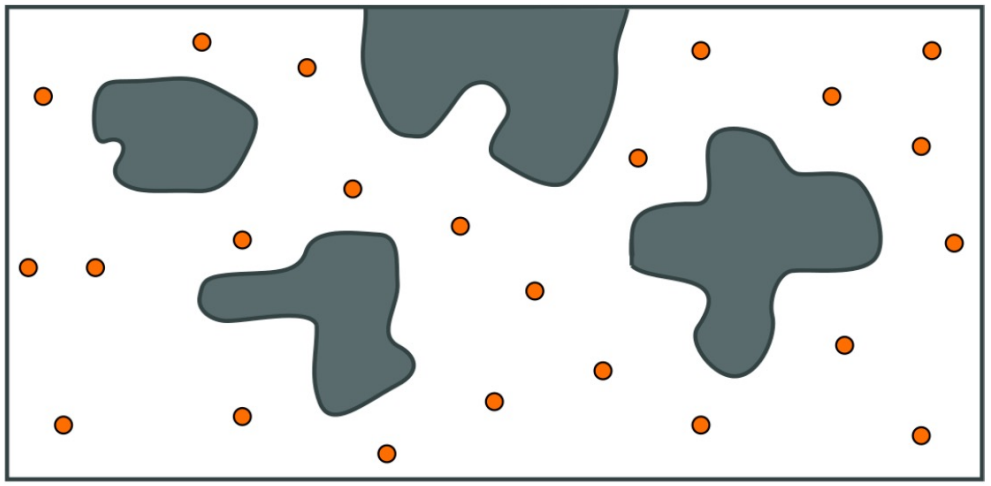
- No need to reason over the entire continuous state space -> no explicit state space representation!
- Well suited for high-dimension search space -> robot arm planning (often 6DoF or more)
- Sample state space of robot (configuration space)
  - for collisions
  - and constrains
    - pose constraints on state (e.g. point gripper up)
    - differential constraints on edges between state:
      - edge represents the motion between two states
      - edges are only added if they adhere to the constraints (e.g. maximum speed; maximum acceleration)
- Probabilistic Roadmap
  - One of first sampling-based motion planner

Material from: Open Motion Planning Library: A Primer; Kavraki Lab; Rice University

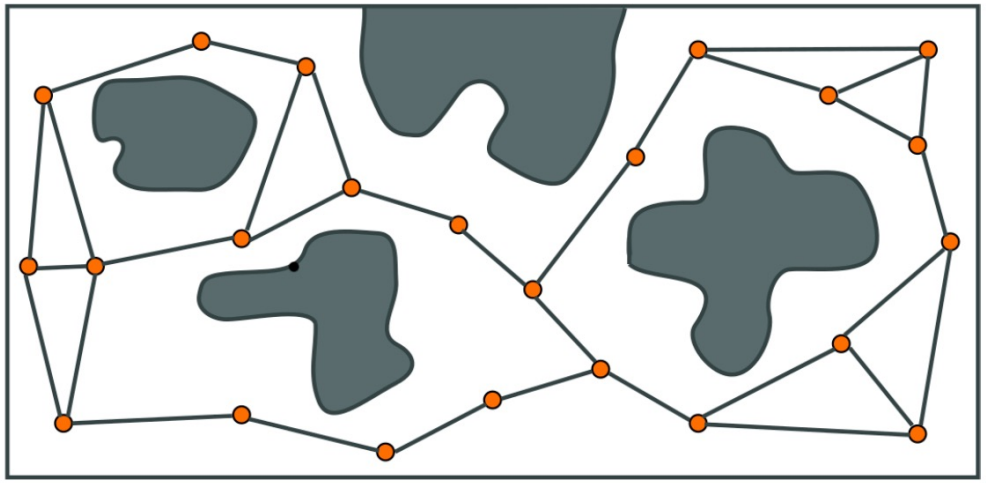
Probabilistic Roadmap



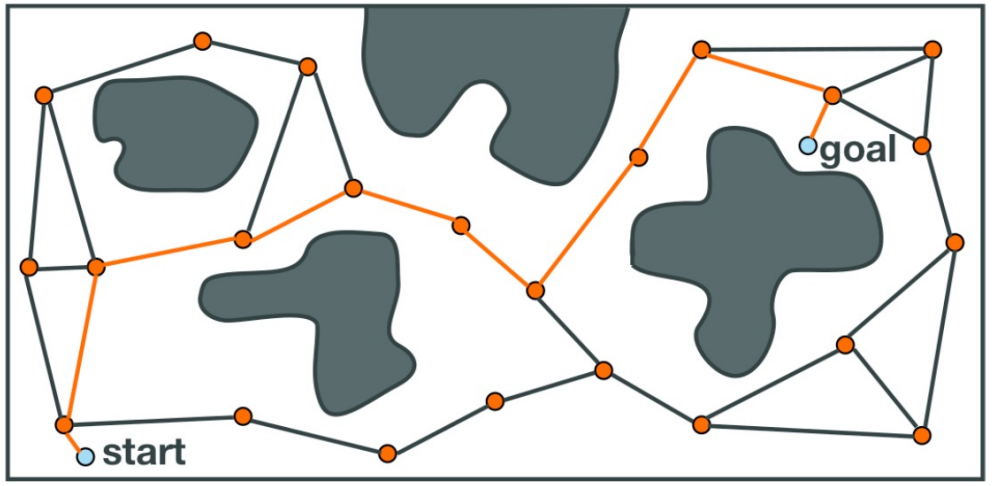
1) Single State (e.g. robot; configuration)



2) Uniform random samples of free state space



3) Connect close samples with straight path



4) Use graph search (e.g. A*) to plan a path

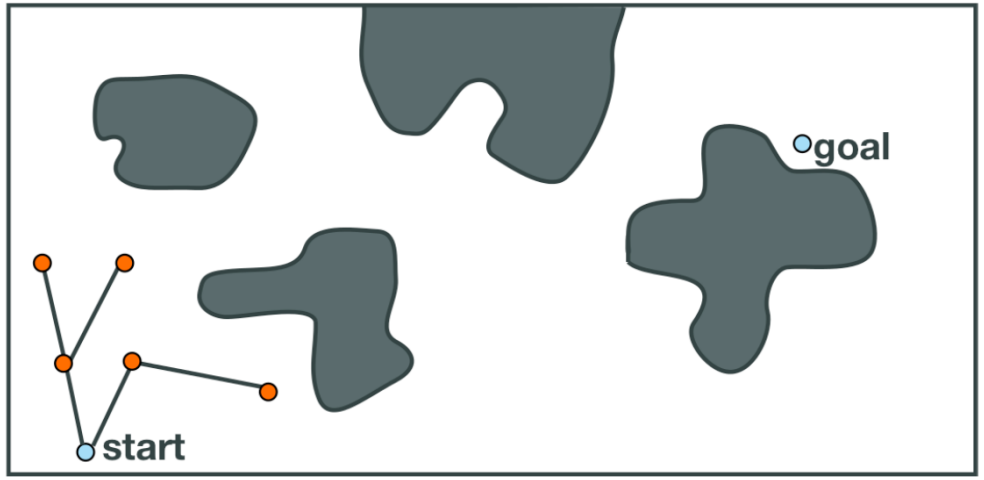
Probabilistic Roadmap vs Tree-based Planner

- Probabilistic Roadmap
  - Goal is used only at last step -> pre-compute graph
  - Need samples distributed over entire space (don’t know where goal is)
  - Too big for high dimension search spaces
  - But: fast planning for small dimension search spaces (e.g. 2D or 3D robot motion planning)

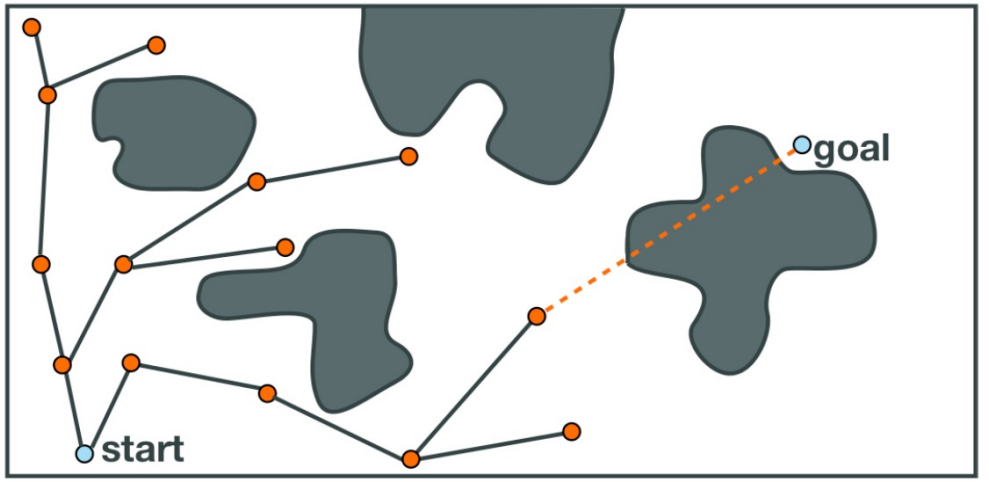
- Tree-based Planning
  - Directed edges -> can encode control commands
    - E.g.: Move up needs more power than move down -> motion is directed
    - Easier to represent differential constraints (e.g. max acceleration of joints)
  - Expand tree on the fly
    - Sometimes with bias towards goal

Tree-based Planner

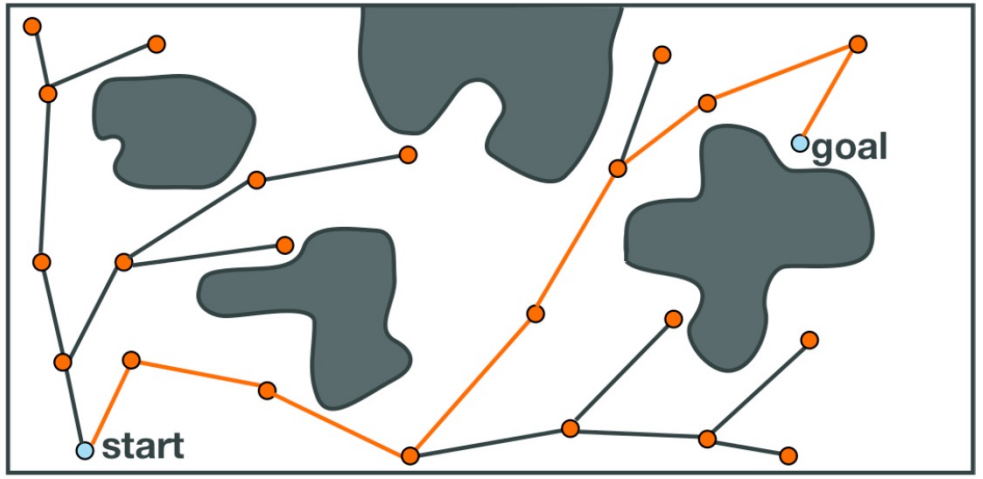
- Tree:
  - no cycles
  - Directed edges
- Grow from start state
- Expansion heuristic to select next state in tree



1) First few samples expanded randomly using expansion heuristic



2) Goal is close to a node, but cannot be connected because collision check of edge fails!



3) Goal is close to a node and can be connected

Rapidly Exploring Random Tree: RRT

- Most famous/ widely used planner concept for Robot Manipulations

- Monte-Carlo based method with
- Bias towards “big empty space” (largest Voronoi region)

Kuffner, J. J., & LaValle, S. M. (2000, April). RRT-connect: An efficient approach to single-query path planning. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation.* IEEE.

Animation from iteration 0 to 10,000

RRT Components

1. **State Space:** A topological space, X
2. **Boundary Values:** $x_{init} \in X$ and $X_{goal} \subset X$
3. **Collision Detector:** A function, $D : X \rightarrow \{true, false\}$, that determines whether global constraints are satisfied from state x. This could be a binary-valued or real-valued function.

- X = Configuration Space or more general
- X = C-space + Velocity + Acceleration + ….

RRT Components

4. **Inputs:** A set, U, which specifies the complete set of controls or actions that can affect the state.

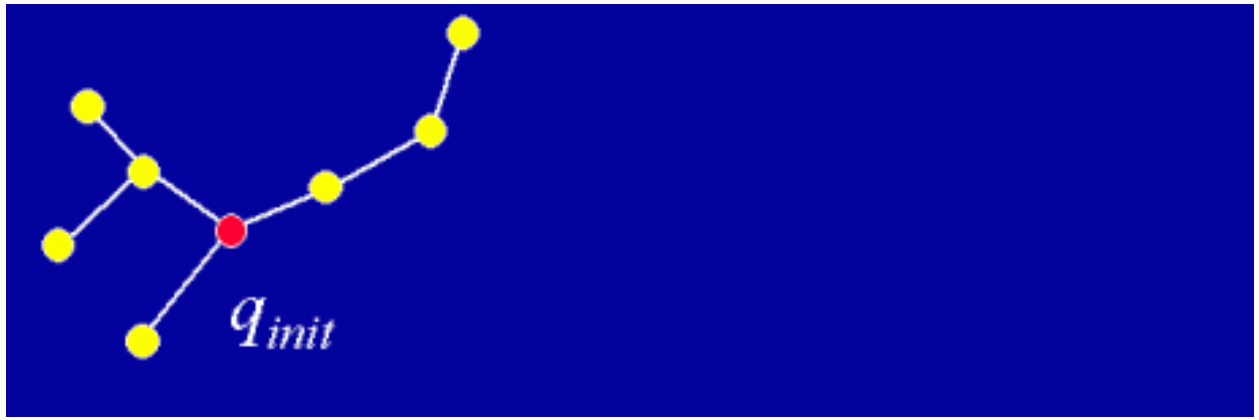
5. **Incremental Simulator:** Given the current state, $x(t)$, and inputs applied over a time interval, $\{u(t') | t \leq t' \leq t + \Delta t\}$, compute $x(t + \Delta t)$.

6. **Metric:** A real-valued function, $\rho : X \times X \rightarrow [0, \infty)$, which specifies the distance between pairs of points in X.

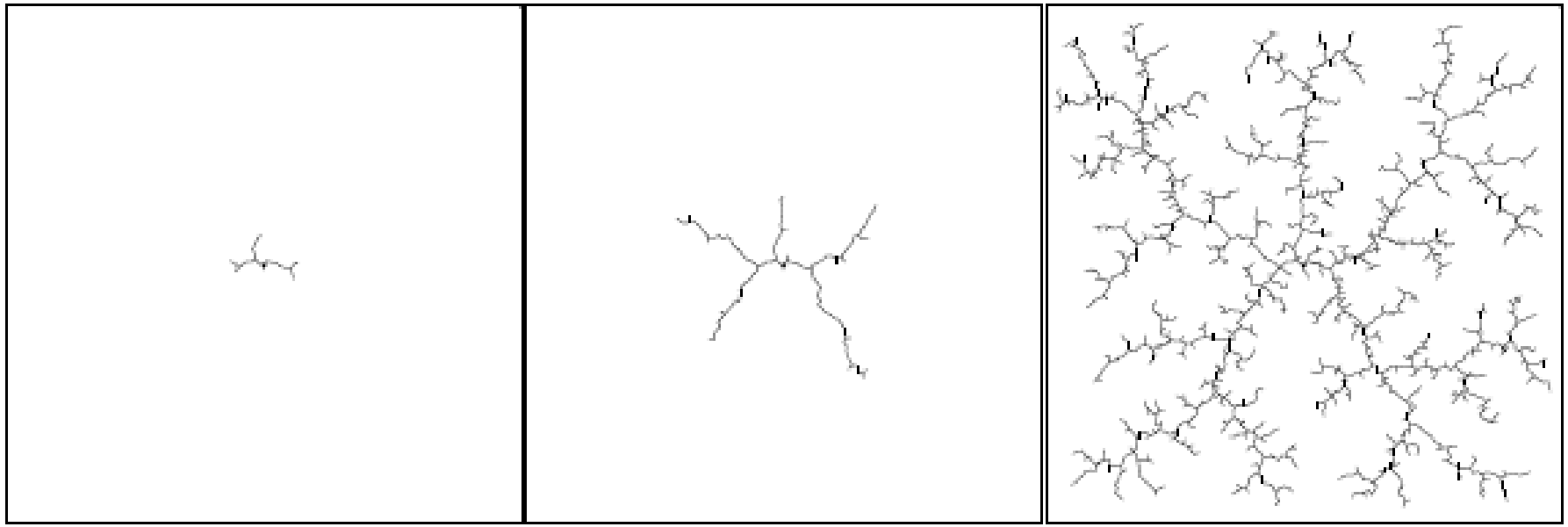
RRT

```
BUILD_RRT(q_init)
 1  T.init(q_init);
 2  for k = 1 to K do
 3      q_rand ← RANDOM_CONFIG();
 4      EXTEND(T, q_rand);
 5  Return T
```

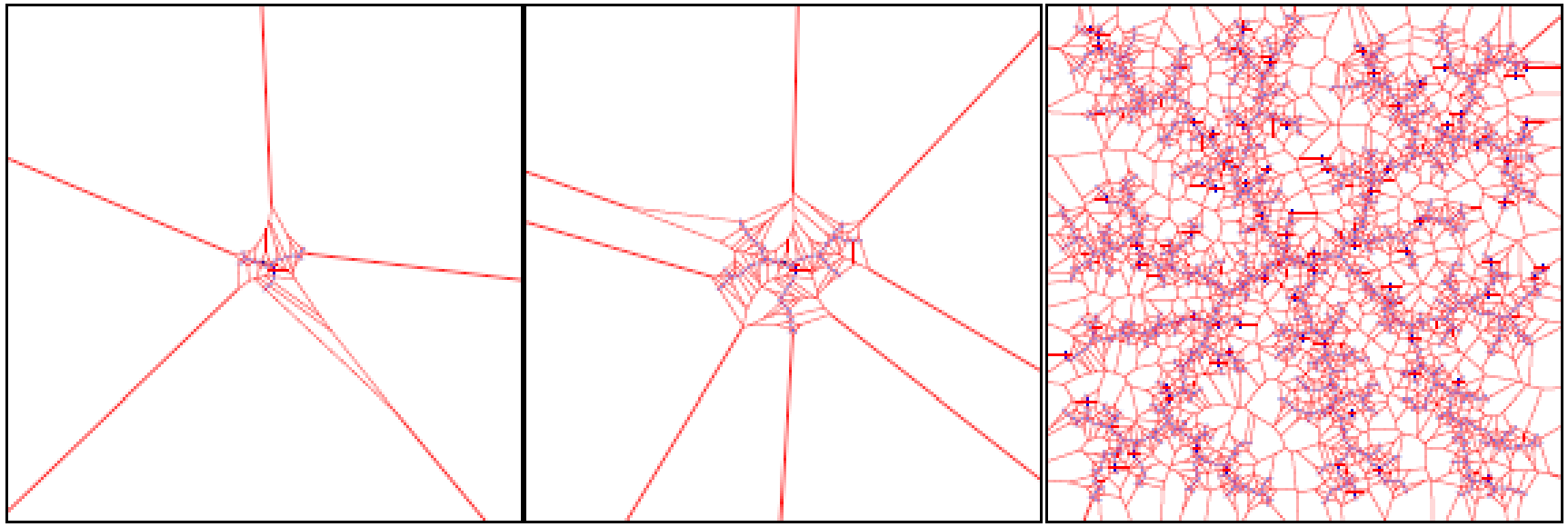
```
EXTEND(T, q)
 1  q_near ← NEAREST_NEIGHBOR(q, T);
 2  if NEW_CONFIG(q, q_near, q_new) then
 3      T.add_vertex(q_new);
 4      T.add_edge(q_near, q_new);
 5      if q_new = q then
 6          Return Reached;
 7      else
 8          Return Advanced;
 9  Return Trapped;
```



Example in holonomic empty space



Why “Rapidly Exploring”?



- What is the probability that a vertex will be extended?
  - Proportional to the area of its Voronoi region
- If just choose a vertex at random and extend, then it would act like random walk instead
  - Biased towards start vertex