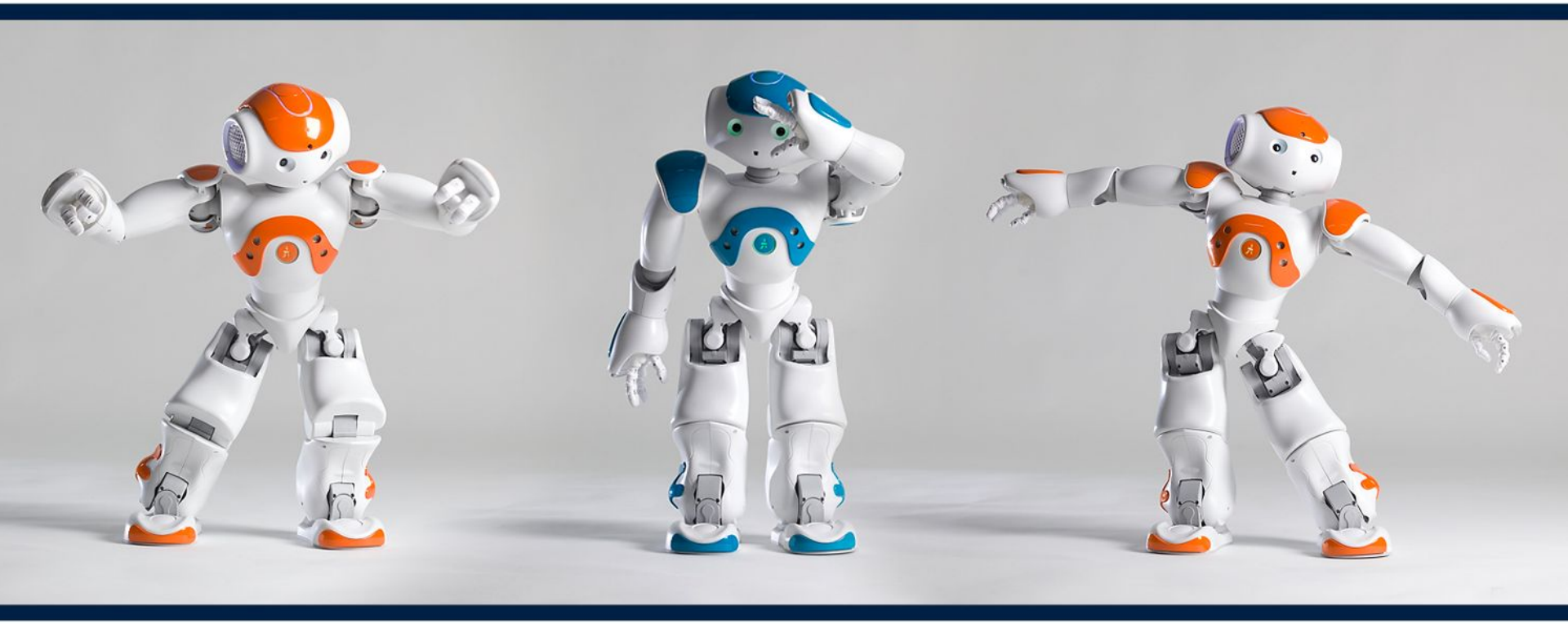


ROS intro

Natalia Lyubova, Vincent Rabaud

31.01.2017

SoftBank Robotics

Outline

PART I

- Description
- Core components
  - Communication
  - Robot-specific features
  - Tools
- Integration with other libraries
- What do we get?

PART II

- ROS basics with NAO

SoftBank Robotics

ROS

- Neither an operating system nor only suited for robots
- flexible framework for programming robots
- developed by Willow Garage in 2007
- aimed at encouraging collaborative robotics software development
- consists of infrastructure, tools, capabilities, and ecosystem

SoftBank Robotics

ROS

Features:

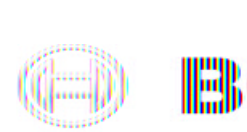
- distributed, modular design
- collaborative environment
- vibrant community
  - research labs, industrial and service robotics
  - 100+ robots
  - 3,000 public packages
  - 300 developers
  - 11K users http://answers.ros.org/questions/
  - Books: http://wiki.ros.org/Books
  - 200 special interest groups
  - ROScon (>400 persons)

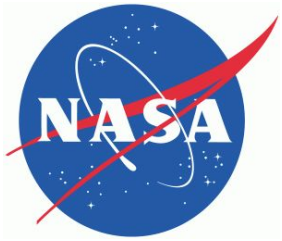
ROS

http://www.ros.org/about-ros

SoftBank Robotics

Built with Industries





Outline

- Description
- Core components
  - Communication
  - Robot-specific features
  - Tools
- Integration with other libraries
- What do we get?

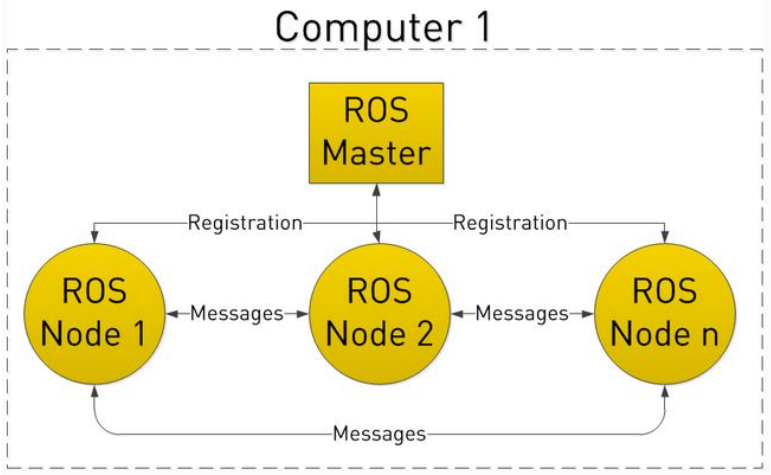
SoftBank Robotics

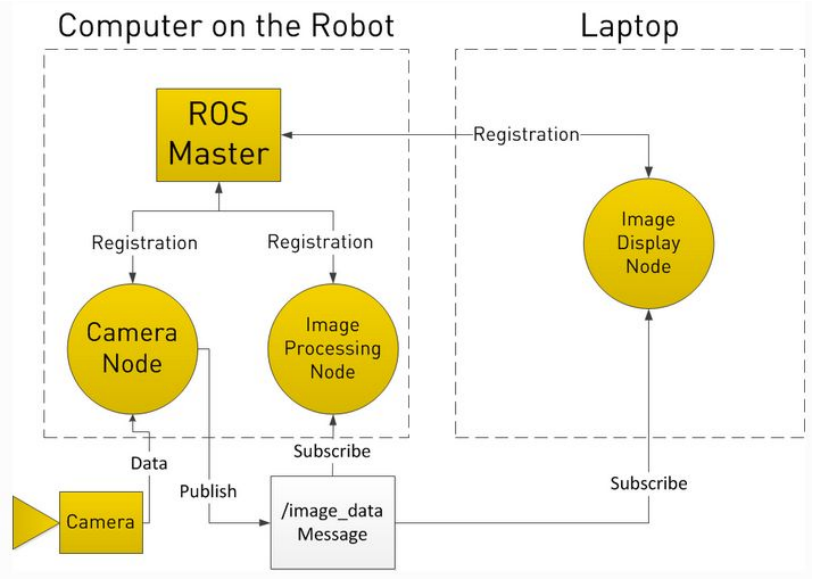
Communication infrastructure

- independent nodes which communicate through publish/subscribe messages

- nodes do not have to be
  - on the same system
  - of the same architecture

- Communication:
  - ROS starts with a master
  - master sets up connection between nodes



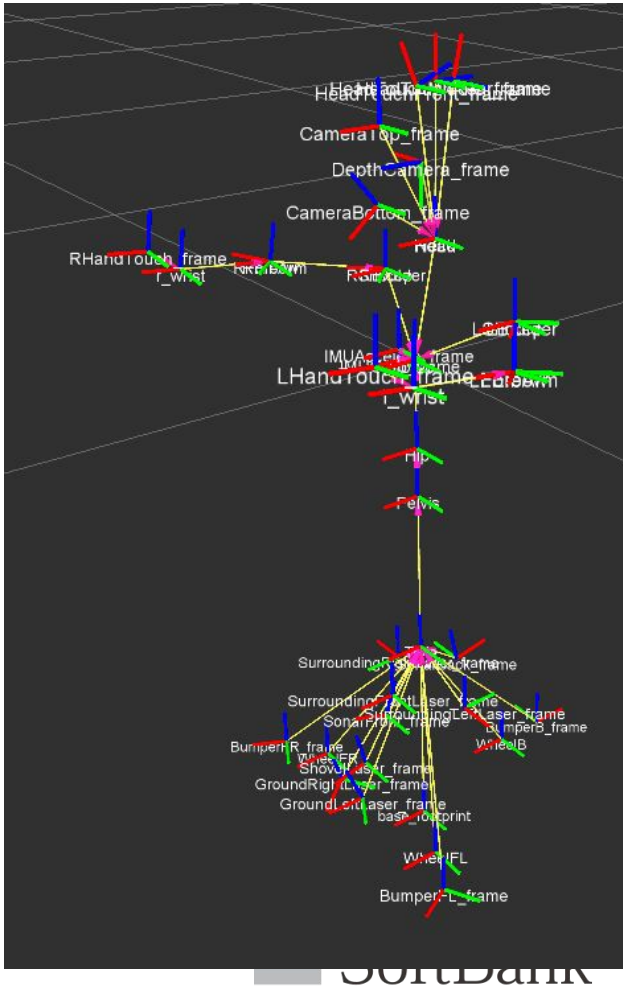


Outline

- Description
- Core components
  - Communication
  - Robot-specific features
    - Robot Description Language : URDF
    - Robot Geometry Library : TF
    - Standard Message Definitions
    - Preemptable Remote Procedure Calls : actions
    - Diagnostics : analysis, troubleshooting and logging
    - Pose Estimation, Localization, Navigation
  - Tools
- Integration with other libraries
- What do we get?

Robot Description Language (URDF)

- XML docs, config via xacro
- describes the robot's physical properties
  - from lengths of limbs and sizes of wheels,
  - locations of sensors,
  - visual appearances of the robot parts
- basis for:
  - low level (kinematics, hardware abstraction)
  - high level (planning, navigation, grasping)
  - GUIs



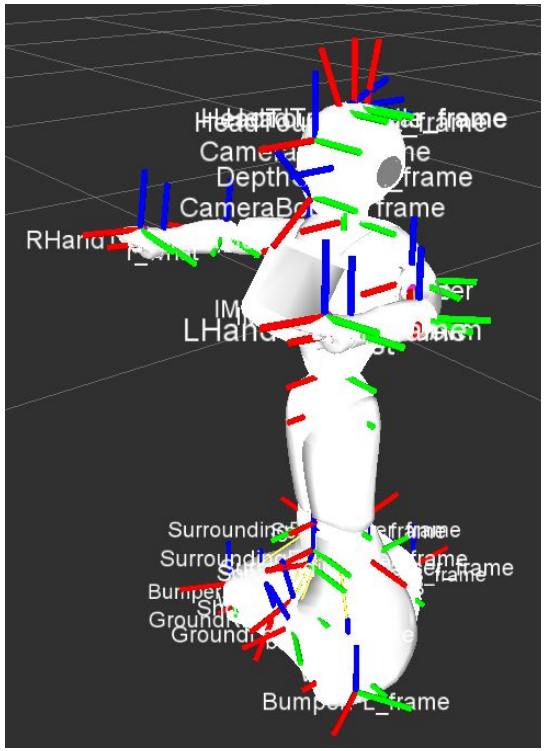
Robot Geometry Library (TF)

Why?

- A robot system has many 3D coordinate frames that change over time

Idea:

- tf keeps track of all coordinate frames over time
- maintains the relationship between frames in a tree structure buffered in time
- let transform points, vectors, etc between frames at any time



Allows to know:

- Print the coordinate transform tree
- Where is the base frame in the map?
- What is the pose of the object in my gripper?

SoftBank Robotics

Standard Message Definition / IDL

- standard types for most commonly used robot utilities
  - actuators
  - sensors (> 300 supported)
  - diagnosis
  - navigation
  - grasping

- What if your device is not supported yet ?
  - can be custom-made
  - bindings for C++, Python, Ruby, C#, Julia, Matlab ...
  - basis for robotics libs

CameraInfo
ChannelFloat32
CompressedImage
FluidPressure
Illuminance
Image
Imu
JointState
Joy
JoyFeedback
JoyFeedbackArray
LaserEcho
LaserScan
MagneticField
MultiDOFJointState
MultiEchoLaserScan
NavSatFix
NavSatStatus
PointCloud
PointCloud2
PointField
Range

SoftBank Robotics

Outline

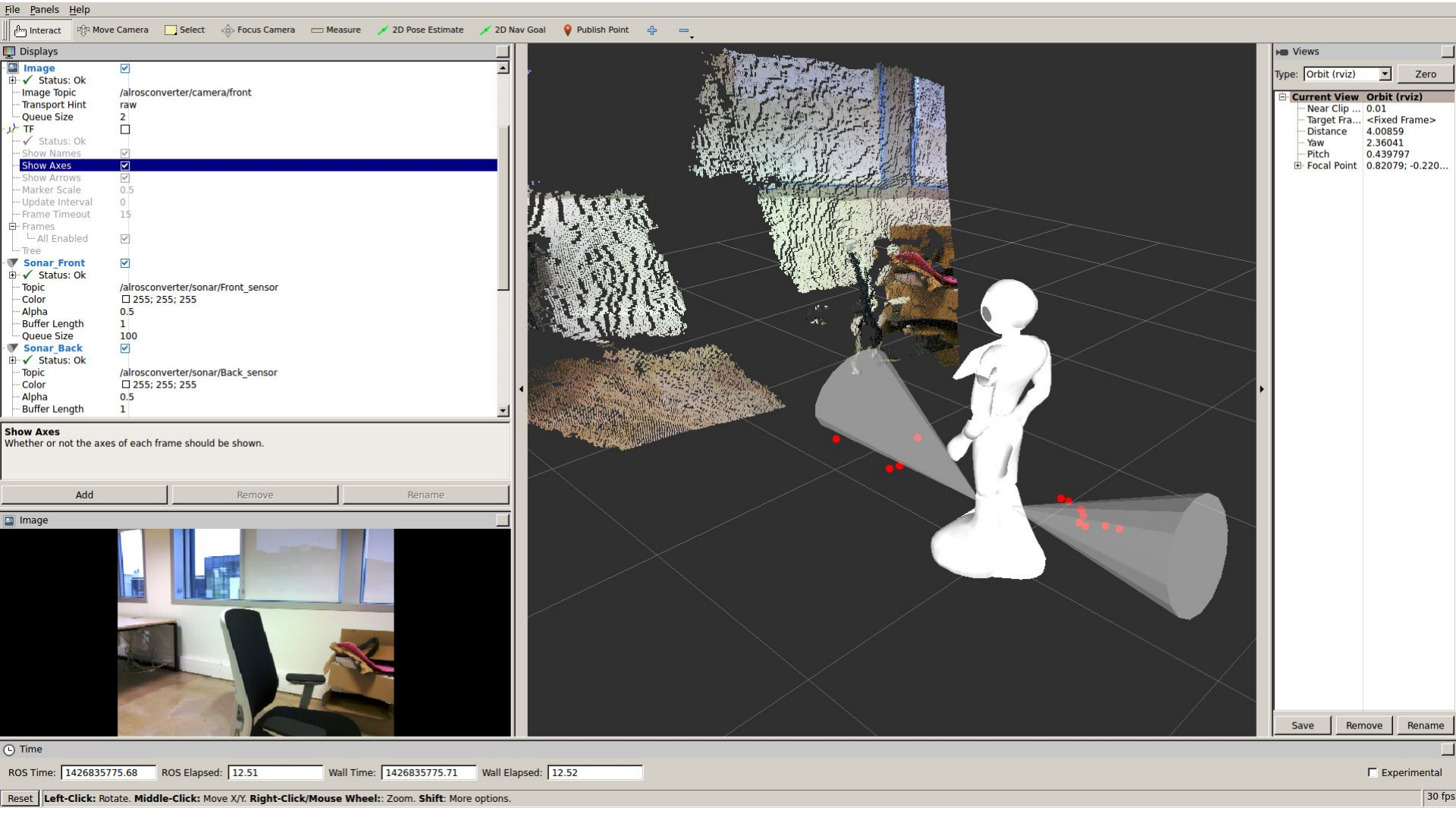
- Description
- Core components
  - Communication
  - Robot-specific features
  - Tools
    - Command-Line Tools
    - RviZ
    - rqt
- Integration with other libraries
- What do we get?

SoftBank Robotics

Command-line tools

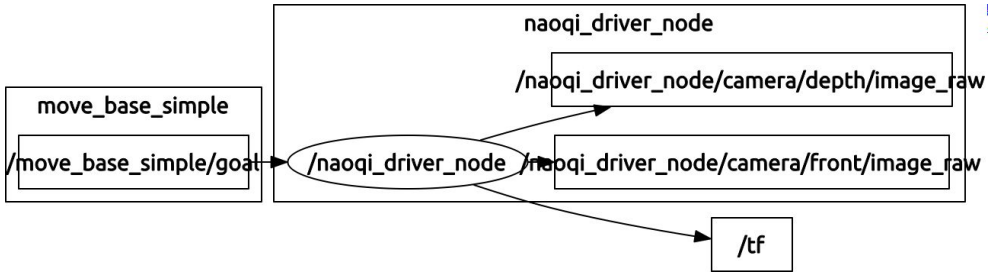
- ROS can be used 100% without GUI
- >45 command line tools:
  - launching groups of nodes:
    - roscore
    - rosrun
    - roslaunch
  - introspecting topics, services, and actions
    - rostopic list
    - rostopic echo <topic>
    - roswtf
    - rosrun tf tf_monitor
    - rosrun tf tf_echo <frame1> <frame2>
  - recording and playing back data
    - rosbag record -a
    - rosbag play bagname.bag

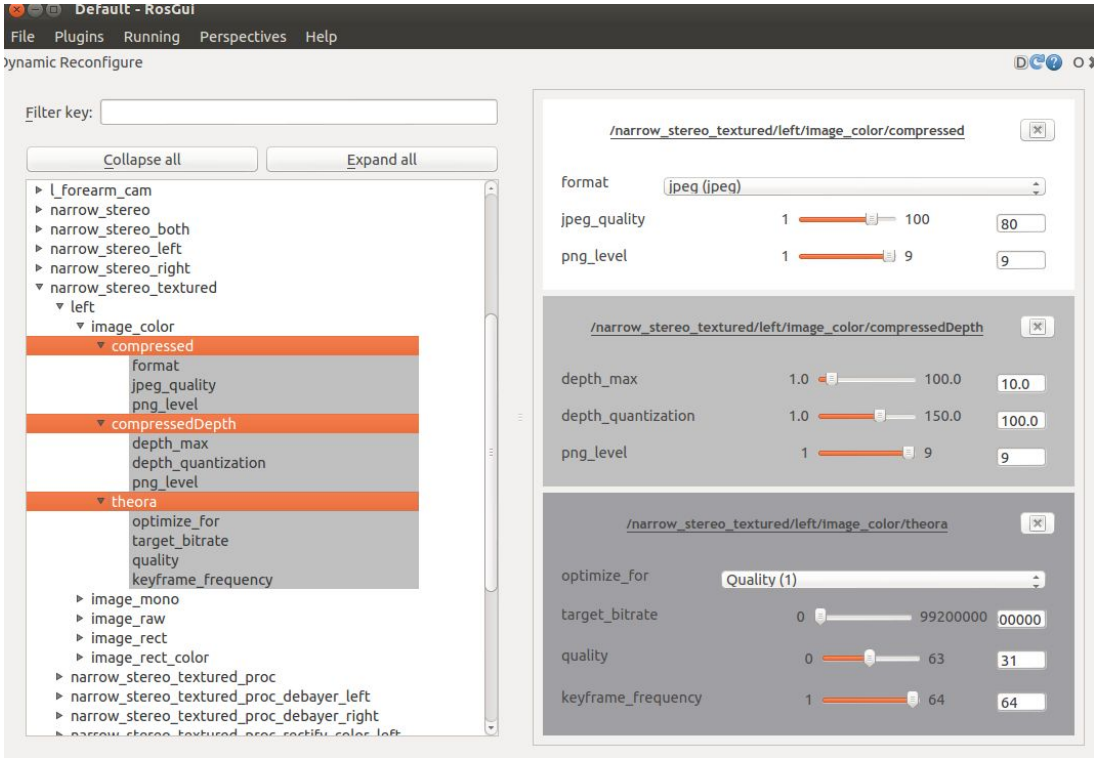
RviZ visualizer



rqt

- Qt-based framework for developing GUI
  - rqt_graph
  - rqt_plot
  - rqt_topic, rqt_publisher
  - rqt_bag
  - dynamic reconfigure
  - logging





Outline

- Description
- Core components
  - Communication
  - Robot-specific features
  - Tools
- Integration with other libraries

- 

SoftBank Robotics

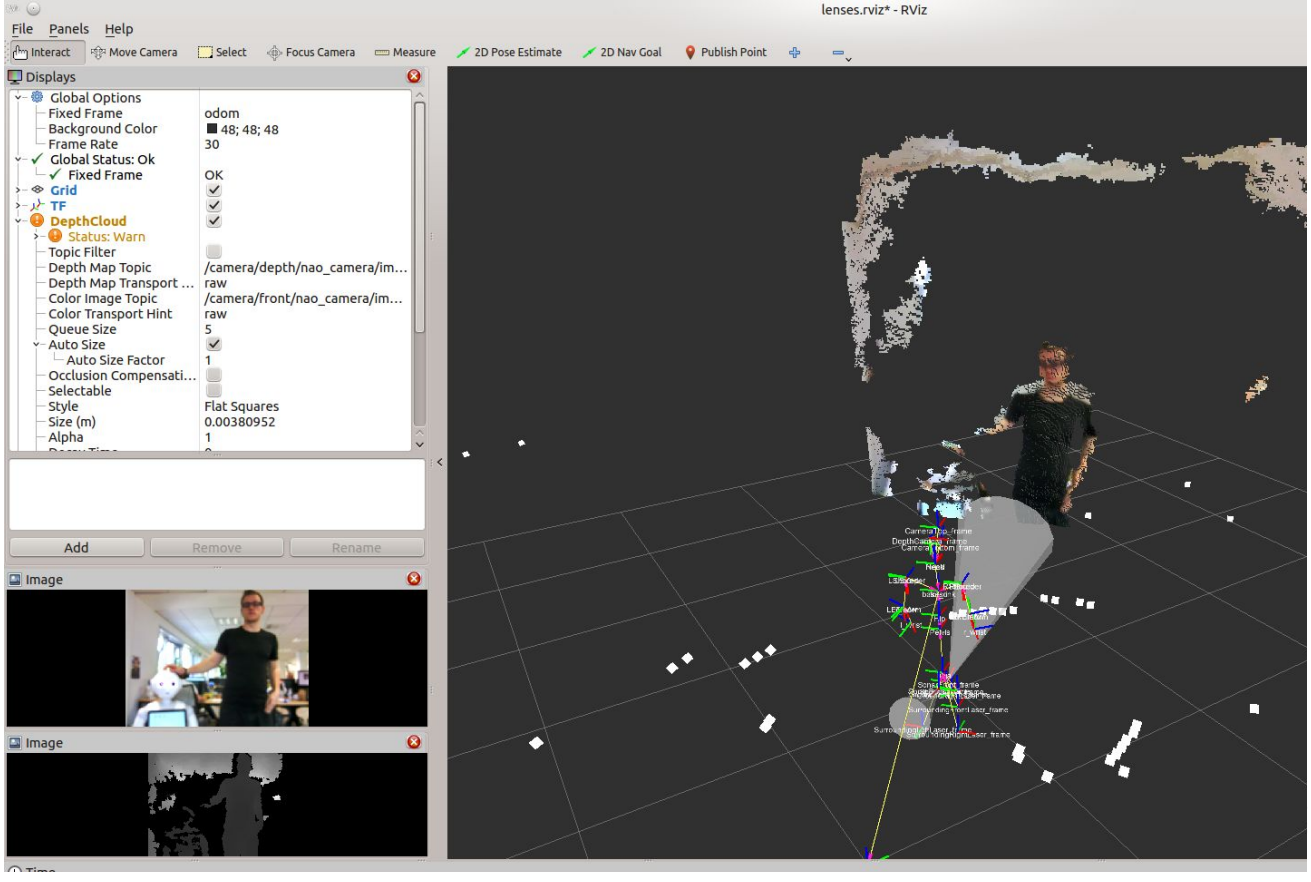
Outline

- Description
- Core components
  - Communication
  - Robot-specific features
  - Tools
- Integration with other libraries
- What do we get?

SoftBank Robotics

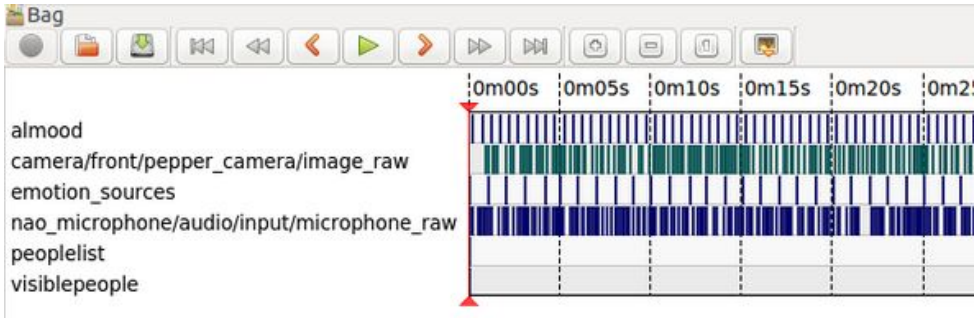
What we get from URDF / IDL

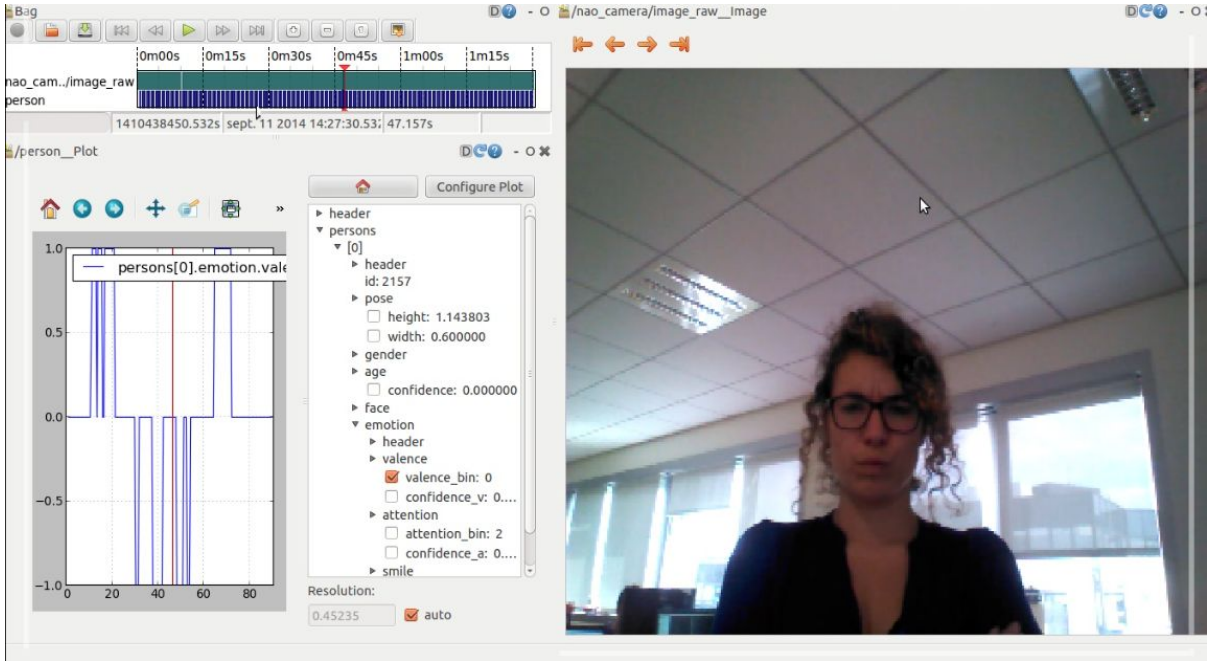
- low level libraries (OpenCV conversion, world representation ...)
- data introspection with ROS GUIs



What we get from URDF / IDL

- data recording (rosbag)
- retrospection
- data annotation
- data injection

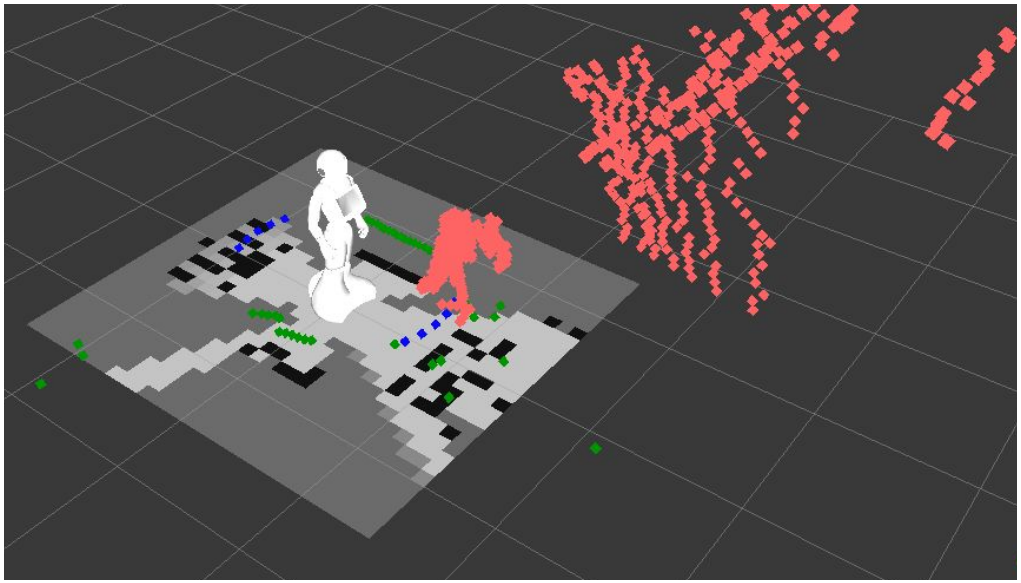


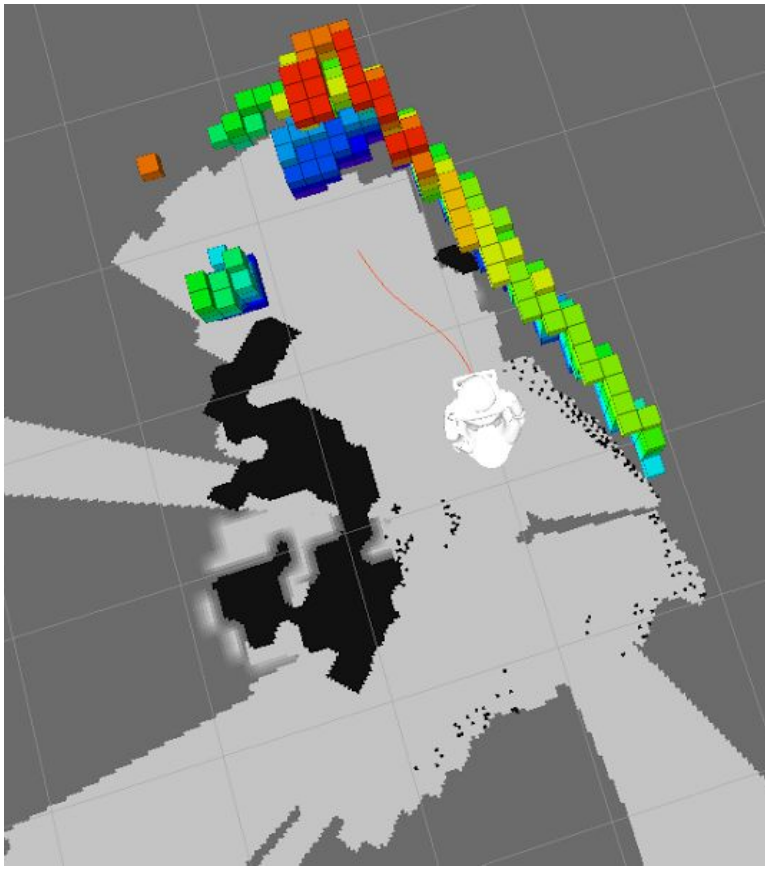


SoftBank Robotics

What we get from URDF / IDL

- localisation
- obstacle detection

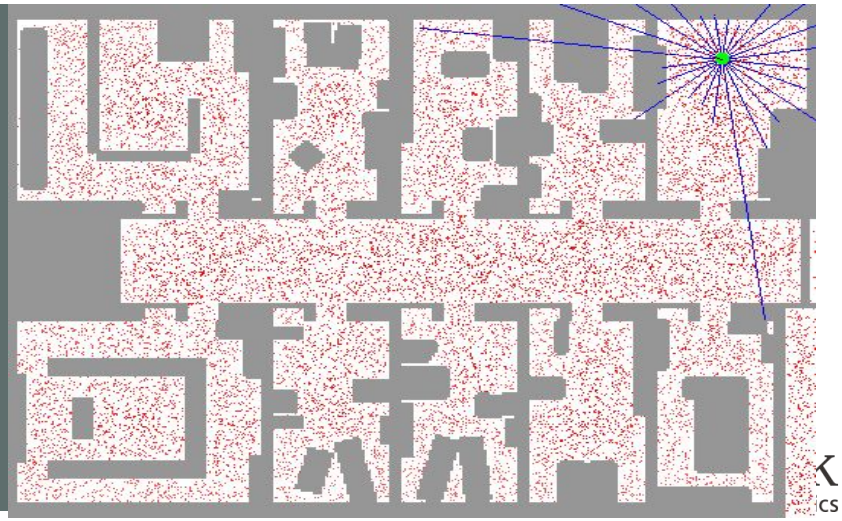




SoftBank Robotics

What we get from communication

- more accurate live GUIs (sonars, lasers ...)
- possibility to try out low/high level nodes (compression, navigation ...)



Outline

PART I

- Description
- Core components
  - Communication
  - Robot-specific features
  - Tools
- Integration with other libraries
- What do we get?

PART II

SoftBank Robotics