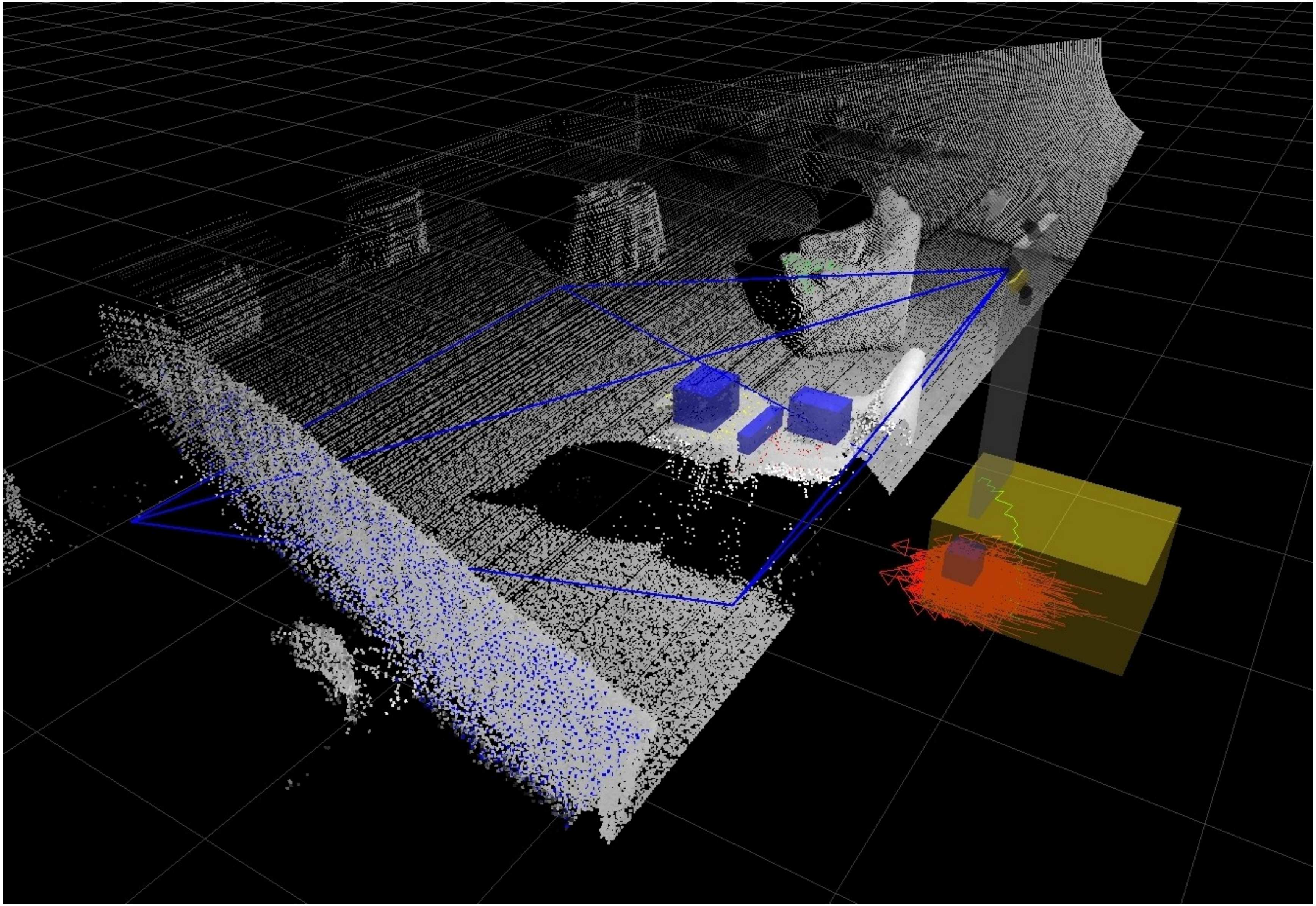
Robot Operating System Tutorial

David Meger
University of British Columbia

CRV 2010 – Tutorial Day
May 30th, 2010

Introduction



What is ROS?

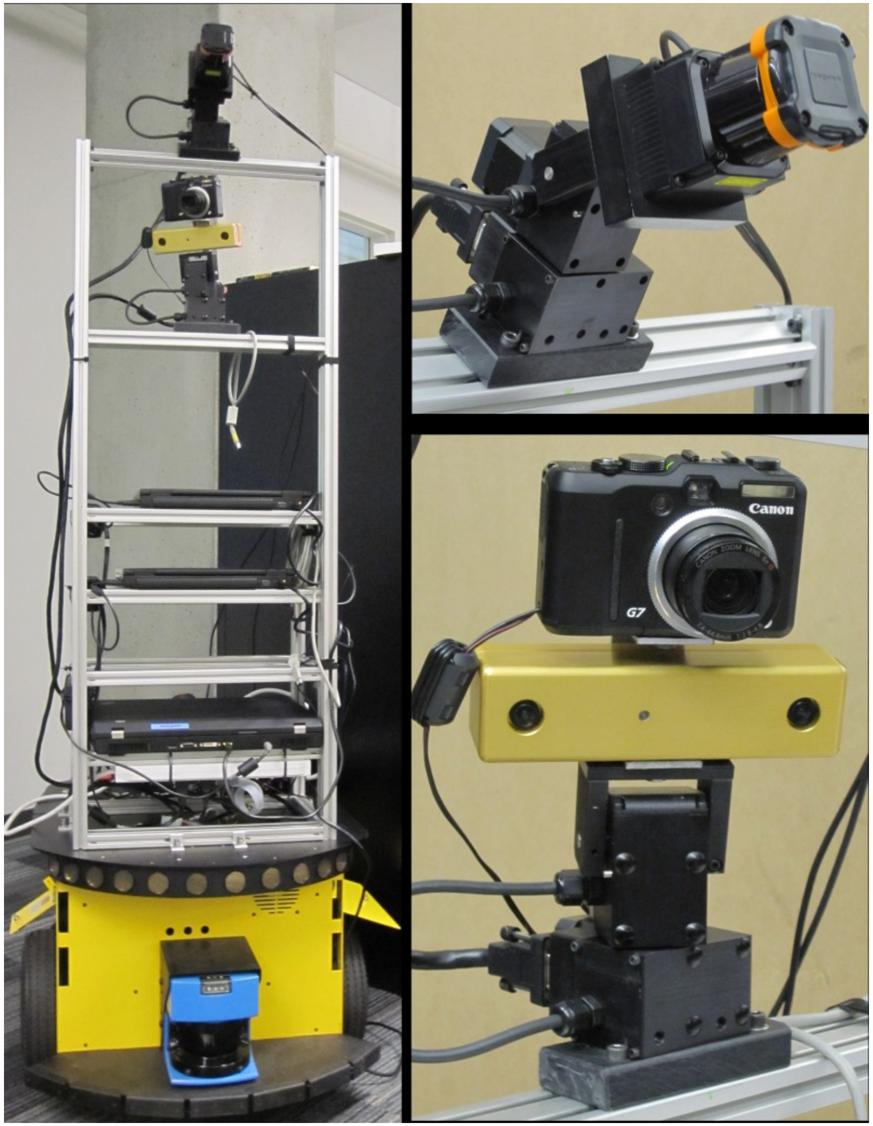
- Short:
  - A ***really*** useful collection of software targeted exactly to help those researching perception, control, and modeling
- Long:
  - An Open-source, meta-operating system for your robot
  - Includes hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management
  - Also provides tools and libraries for obtaining, building, writing, and running code across multiple computers
- Examples:
  - (milestone 2 movie)
  - (towel folding movie)

ROS Components (core)

- Middleware
  - Inter-process communication API
  - Integration of C/C++, Python, Octave, Java, Lisp
  - Standardized packaging and build environment
- Debugging and visualization tools
- Open-source implementation of standard robot applications
  - Robot Control (Player/Stage)
  - Motion planning (mobile robots and manipulators)
  - Localization and Mapping (Gmapping, EKF and AMCL)
  - Computer Vision (OpenCV)
  - Processing 3D information from lasers, stereo, etc...

ROS Components (core)

- (Curious George mapping video)



ROS Components (community)

- All institutions using ROS are encouraged to post their application code (and many do!)
  - CMU, MIT, Stanford, Berkeley, Munich, Bosch, Tokyo, Georgia Tech, Washington, Leuven, all heavily involved
- Improving the “scientific-ness” of robotics research:
  - ICRA 2010 include released version of ROS code that was used to produce their results for repeatability
  - PR2 beta program has provided 11 institutions with identical robots / sensors

Programming ROS - Basics

- Many processes (nodes) communicate via specified message types:
  - Images
  - Point Clouds
  - Laser scans
  - User specified
- Synchronization using one of 3 specified methods:
  - Topics (asynchronous publish/subscribe)
  - Services (synchronous)
  - Actions (asynchronous with feedback monitoring)
- (usb camera driver example)
- (canon camera driver example... if my luggage arrives)

Helpful Commands

- (rostopic {list,echo,info})
- (rosservice {list,call})
- (rosmsg show)
- (rxgraph)
- (rviz)

ROS for CV

- OpenCV is highly integrated via cv_bridge
- ROS Image/CameraInfo types compatible with all tools
    - Stores geometric information for the camera via tf library – excellent for 3D vision
    - Robust and accurate calibration routines included
    - (live image example)

For more info...

- www.ros.org For many many good tutorials and complete ROS API documentation
- Google “Curious George UBC” to find the numerous object recognition approaches implemented at UBC are available as ROS packages:
  - SIFT [Lowe IJVC 2004]
  - Deformable Parts Model [Felzenszwalb et al CVPR 2007]
  - Contour matching
- See poster tomorrow!

Questions