Technische Universität München
Institut für Informatik
Prof. Michael Beetz, PhD
Nico Blodow, Mihai Dolha, Zoltan-Csaba Marton, Dejan Pangercic

SS 10
Aufgabenblatt 1
07.05.2010

Übungen zur Vorlesung Sensor-enabled Intelligent Environments

TF, Point Clouds, Visualization

In this session you will learn about the absolute basics of every robotic system - coordinate transforms. It is very important to keep track of multiple coordinate frames over time and over spatial space. ROS's "TF" (`http://www.ros.org/wiki/tf`) library reaches this by using a tree-like structure buffered in time. Additionally a PointCloud message data and an "rviz", the tool for visualization will be discussed. . First lets checkout some bleeding-edge ROS tools not included in the ROS stable version yet:

1. "svn co" following packages into directory contained in the ROS_PACKAGE_PATH variable of your ROS installation:
   (a) `https://code.ros.org/svn/ros-pkg/stacks/point_cloud_perception/trunk/flann/`
   (b) `https://code.ros.org/svn/ros-pkg/stacks/point_cloud_perception/trunk/ann/`
   (c) `https://code.ros.org/svn/wg-ros-pkg/trunk/stacks/semantic_mapping/point_cloud_mapping/`
   (d) `https://code.ros.org/svn/ros-pkg/stacks/point_cloud_perception/trunk/cminpack/`.

2. Compile "point_cloud_mapping" package using rosmake command.

3. Download the precreated package **sie_two** by running a "wget `http://ias.cs.tum.edu/~pangerci/sie_two.tar`" command and untar it into the directory contained in the ROS_PACKAGE_PATH variable of your ROS installation.

4. In the package's directory "data" you will find a dump of a PointCloud message of the household scene named "test.pcd". Publish it on a topic using "pcd_to_msg_node" program from the "point_cloud_mapping" package and visualize it using "rviz" and its PointCloud widget (to find the topic name use "rostopic list" command). Create a screenshot. "rviz" is a 3D visualization environment described here: `http://www.ros.org/wiki/rviz`.

5. Write a program named "cutter.cpp" (place it in package **sie_two**) that subscribes to the topic previously mentioned "test.pcd" is being published to, cuts out points within a box with dimensions of [1mx1mx1m] and publishes these points on the topic "/box". Visualize the latter in "rviz" and create a screenshot.

6. Now we will create a simple robot with 2 coordinate frames: i)"base_link" and ii)"head" using "tf's" "static_transform_publisher" tool (rosrun tf static_transform_publisher ...). The offset between both frames is as following: X:0m Y:0m Z:1m R:90deg P:0deg Y:0deg.

7. Visualize this simple transform (aka our trivia robot) in "rviz". Create a screenshot.

8. Now we will use this transform to transform the "test.pcd" from "base_link" frame to "head" frame[1]. Write a program named "transform.cpp" that subscribes to the topic "test.pcd" is being published to, creates a "tf::TransformListener listener" object (see `http://www.ros.org/wiki/tf/Tutorials/Writing%20a%20tf%20listener%20(C%2B%2B` for an example) and deploys a "transformPointCloud()" method (see `http://www.ros.org/wiki/tf/Overview/Using%20Published%20Transforms`) to actually transform the PointCloud message. Visulize the transformed PointCloud

[1] Note: test.pcd is being published in the frame base_link by the pcd_to_msg_node

in “rviz” and create the screenshot.

9. Submit the package **sei_two** and the screenshots to blodow@cs.tum.edu by 13.5.2010.

10. Bonus camera calibration: drop-in on Monday, 10.05.2010 @18.00 in our **NEW** lab in Karlstrasse 45, 3rd floor, room 3001 (ask for Dejan).