Visual-Based Navigation
J. Stückler, V. Usenko, L. von Stumberg
Winter Semester 2017/2018

Computer Vision Group
Department of Informatics
Technical University of Munich

Sheet 1

Topic: Robot Operating System (ROS) and Image Processing Basics

Submission deadline: Monday, 30.10.2017, 23:59
Hand-in via email to visnav_ws2017@vision.in.tum.de

General Notice

To be admitted to the final exam, every student has to submit solutions to all exercise sheets. All exercises should be done in teams of two to three students. Each student in a team must be able to present the solution to the tutors during the exercise sessions. If you have not yet done so, please register yourself together with your team members on the team list in room 02.05.14.

We will use ROS Kinetic on Ubuntu 16.04 in this lab course. It is already installed on the lab computers. If you want to use your own laptop, you will need to install these versions of Ubuntu and ROS.

Introduction

The goal of this exercise is to familiarize yourself with the ROS operating system. You should also get acquainted with the basic ROS tools for image processing, camera calibration, pose representation, point cloud processing, and visualization.

Exercise 1 (4 points):

Please follow the following tutorials:

- "Beginner Level" ROS tutorials on `http://wiki.ros.org/ROS/Tutorials`
- C++ tutorials of the TF library `http://wiki.ros.org/tf/Tutorials`
- The "Markers" and "Interactive Markers" tutorials of the rviz package: `http://wiki.ros.org/rviz/Tutorials`. Also read about the basic marker types `http://wiki.ros.org/rviz/DisplayTypes`.

To confirm what you have learned, please answer the following questions:

(a) What is a ROS node?

(b) What is the difference between ROS messages, services, and parameters?

(c) What does TF stand for, and what is a TF tree? What do you need to implement in your node to get relative poses between two frames at a specific time through TF?

(d) What are the possibilities to visualize point clouds using RViz? Give at least two.

Exercise 2 (4 points):

Download the following two bag files containing camera calibration datasets:

- `https://vision.in.tum.de/rgbd/dataset/freiburg1/rgbd_dataset_freiburg1_rgb_calibration.bag`
- `https://vision.in.tum.de/rgbd/dataset/freiburg2/rgbd_dataset_freiburg2_rgb_calibration.bag`

Replay the bag files and confirm that you can subscribe to and visualize the images in RViz.

(a) Use ROS camera_calibration to obtain the intrinsic parameters of the RGB camera on both datasets. Compare your results with the calibration results given on this website `https://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats#calibration_of_the_color_camera`. The checkerboard has 8x6 internal corners and a square side length of 0.02m.

(b) Write a C++ node to undistort the images with your parameters and the ROS default parameters (see `https://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats#calibration_of_the_color_camera`).

   - Use cv_bridge to convert images to the OpenCV image format.
   - Use image_geometry::PinholeCameraModel to compute an undistorted image (in OpenCV format).
   - Publish the undistorted image on a ROS topic and visualize it in RViz.

   What are your observations of the effect of the differences between both parameter sets?

Exercise 3 (8 points):

Download the following bag file for use in this exercise:

- `https://vision.in.tum.de/rgbd/dataset/freiburg2/rgbd_dataset_freiburg2_desk.bag`

(a) Write a C++ node that reprojects the RGB-D images in the bag file to 3D point clouds with RGB information for each point. Use the PCL PointCloud class to represent the point clouds, and use the ROS Default camera calibration. Visualize the point clouds with RViz. An explanation of the ROS topics in the bag file is given here: `https://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats#alternate_file_formats`

(b) Augment the node to transform the point clouds directly in your node into the "world" frame given in the tf messages. You can use TF and pcl_ros packages for this task. Visualize the point cloud and the ground truth pose (through TF) in RViz. How has the ground truth pose been recorded in the benchmark dataset? What is your observation on the precision of the RGB-D point clouds and the ground truth pose?

Submission instructions

A complete submission consists both of a PDF file with the solutions/answers to the questions on the exercise sheet and a ZIP file containing the source code that you used to solve the given problems. Note all names and matriculation numbers of your team members in the PDF file. Make sure that your ZIP file contains all files necessary to compile and run your code, but it should not contain any build files or binaries. Please submit your solution via email to `visnav_ws2017@vision.in.tum.de`.