Lab Course: Robot Vision WS 2012/2013

Philipp Heise, Brian Jensen, Sebastian Klose

Assignment 4 - Due: 17.12.2012

Exercise 1 RGBD-Data Visual Odometry

In this exercise, you have to adapt the code from the previous sheet to work on RGBD-Data. In principle, working with RGBD data eases life a lot, as for (almost) each pixel position, you already get an associated depth value from the sensor. So there's no need for the left-to-right stereo matching scheme anymore. Proceed as follows:

1. *(Adapting Stereo-VO code)*
   - download one of the datasets (bag files!) from http://cvpr.in.tum.de/data/datasets/rgbd-dataset, e.g. freiburg2/desk
   - Create a new node named `rvc_rgbd_vo`
   - subscribe to the rgb and depth image topic, as well as the `camera_info` topic
   - use your temporal matching scheme from assignment 3, to track features throughout consecutive images
   - use your Umeyama method from the last assignment to compute the 6-dof pose of the moving camera
2. *(Evaluating results)*
   - evaluate your motion estimates, by comparing your results with the published `tf` frame in the bag files
   - store your estimated trajectory to a `txt` file and use the online or python evaluation tools provided with the dataset, to evaluate your erros
   - For efficiency, the bag-files do not contain point clouds. At each frame, create the point-clouds from the rgb and depth image. Publish it and use rviz to visualize the cloud. Once visual odometry algorithm is working, you can increase the decay time of the cloud, to see the overlapping clouds in rviz and, up to a certain extent, the full 3D scenery.

Exercise 2 3D Occupancy Grid Mapping

For navigation, a robot needs to be aware of the free-space within an environment. In this exercise you shall use an existing package named octomap.

1. *(Learning about octomap)*
   - Get familiar about whith octomap and what functionality it provides
   - install the required package on your pc and browse the examples
   - Useful packages:
     - http://www.ros.org/wiki/octomap_server

- http://ros.org/wiki/octomap_ros
- http://octomap.sourceforge.net

2. *(Octomap mapping)*
   - setup a launch file, that brings up the necessary nodes, to create a occupancy grid-mapping application
   - As input you can use the result from the previous exercise (pointcloud of rgbd + associated pose) or your results from the last assignment sheet (stereo)
   - play around with the parameters available in the `octomap_server`
3. *(2D-Occupancy Grid)* The `octomap_server` also provides access to a projection of the 3D grid, yielding a 2D occupancy grid which is usefull for navigation of ground robots.
   - Visualize the 2D-grid in rviz

Exercise 3 Other features and descriptors `(optional)`

This exercise is an optional one, if you have time. Refactor your code-base of the last-two assignments, such that you can use the `OpenCV` pipeline for feature detection and descriptor matching. The idea would be, that you can easily try out all kind of different combinations of features and descriptors (or at least a few). This way, it's easier to evaluate, which features and/or descriptors are most suited for the task of visual odoemtry.