HOCHSCHULE BONN-REHIN-SIEG

SDP PCL report

Team members: Iuri Araujo, Shehzad Ahmed, Oscar Lima

November 23, 2013

1 PROJECT TASK DESCRIPTION

Our task for this semester was to understand, and apply different components of PCL (point cloud library) as well as the agentification of the used components.

Other goals involved in the project were the constructions of the UML diagrams such as class diagram, sequence diagram and state machine.

This will be done by designing and constructing a useful program, implementing all this mentioned requirements.

Our program contains a menu which presents the feasible actions such as visualise a point cloud, read a pcd file (point cloud file) and filter application, downsampling or passthrough filter.

Later on asks the user to select an option so the program can perform the desired action.

2 SHORT DESCRIPTION OF THE WORK DONE SO FAR

First we installed PCL library [1], then we went through the tutorials. We developed 8 examples:

[1] http://pointclouds.org

- pcd_read Reads pcd files and prints in console x, y, z coordinates of each point
- pcd_write - Generates file_name.pcd files out of a given pointcloud
- pcl_viewer - Visualize a given pointcloud
- donwsampling - Reduces the size of the cloud computing the centroids of a fixed sized cubes
- passtrough_filter - cuts, or takes slices out of the cloud around a given axis
- kinect_io - Visualise live data from microsoft kinect[2] sensor device
- kinect_snapshot - Visualise live data from kinect and if user press spacebar then write pcd file

After that we integrate this examples into a main program as described in the task description.

Finally we generated the UML diagrams of the program. (class, sequence and state machine)

All the code is public and available trough github website[3] inside MAS-GROUP repository.

3 EFFORT SPENT ON THE PROJECT

- Each of us spent 7 hours per week (in average) for the project
- Effort was divided into:
- Understanding pcl library
- pcl implementation
- Bug fixing
- Solving github issues
- Installing kinect driver openni[4]
- CMake[5] usage and eclipse[6] project creation

[2] http://www.microsoft.com/en-us/kinectforwindows
[3] https://github.com/mas-group/sdp_ws2013_pcl
[4] http://www.openni.org
[5] http://www.cmake.org
[6] http://www.eclipse.org

4 Positive experiences that went well in the first phase

- We got a general overview understanding of the plc library
- We learn how does the downsampling and passtrought filters works
- We improve our git[7] skills on a 3 member project
- We improve our skills in C++ language
- We learn CMake basics
- We got the first insights in ROS[8] plc usage

5 A list of items in need of improvement

- There are still some bugs around the program
- Agentification must be improved
- Diagrams are having errors, we need to correct them

6 Goal suggestions for the second phase

- Current program migration to ROS platform
- Proper agentification of the program
- Segmentation implementation
- Outlier filter implementation

[7] http://git-scm.com
[8] http://www.ros.org