CS-7630 Autonomous Robotics

Homework – Perception

Support: cedric.pradalier@georgiatech-metz.fr
Office hours: 10:00 to 18:00.

This homework assumes that you have completed the teleoperation homework.

Submit the two ros packages (tar.gz or zip) by email by next Monday, 10:00 to cedric.pradalier@georgiatech-metz.fr

Step 1: Collision Avoidance

Scene: `rosControlExample.ttt`

Write a small ROS package that uses the laser point cloud to prevent collisions when driving manually. The suggested algorithm is the following (but it is only a suggestion):

- Only consider points in the direction of motion, less than a maximum distance away and not too far on the side.
- Set a minimum distance below which maximum velocity is set to 0.0
- Between that and the maximum distance implement a smooth linear scaling.

To read the point cloud in C++, you need to use the point-cloud library (ros package pcl_ros). Here is a example:

```
#include <pcl_ros/point_cloud.h>
#include <pcl/point_types.h>
void pc_callback(const sensor_msgs::PointCloud2ConstPtr msg)
{
     pcl::PointCloud<pcl::PointXYZ> pc;
     pcl::fromROSMsg(*msg, pc);
     for (i=0;i<pc.size();i++) {
          printf("x %f y %f z %f\n",pc[i].x,pc[i].y,pc[i].z);
     }
}
```

As a suggestion, you can start from the `collision_avoidance_base package` in `/cs-share/pradalier/vrep_ros_ws`. Don't forget to use the launch file provided, after adapting it to your needs, and in particular launching your teleop node.

Step 2: Face detection

Scene: `rosControlStabilisation.ttt`

A face detection example is provided in the package `face_detect_base` in the `vrep_ros_ws` workspace in `/cs-share/pradalier`. Try it using the image provided by V-REP (alternatively, you can also use a node to open a standard web-cam to provide the image).

Modify the package so that it publishes the detected faces. It is suggested to publish the detected faces using the RegionOfInterest standard message. However, because there might be several faces in one image, you might have to create your own message to publish an array of regions of interest.

Publish the image with the marked faces on second ros topic.

As an option, you can try publishing 3D markers in Rviz when faces are detected. Check the visualization tutorials on http://www.ros.org/wiki/visualization/Tutorials (Useful for later, recommended!)

Step 3: TurtleBots

The turtlebot are similar in principle to the bubbleRob used in V-Rep. Select one turtlebot that will be yours for the duration of the semester. You now have 2 tasks, which should involve NO programming (editing an XML file is not programming).

- Go through the turtlebot tutorials to get to the point where you can use the default turtlebot teleop to move the robots around.
- Create a launch file that will instantiate your teleoperation and collision avoidance on the turtlebots, instead of the default one.

When you're done, make sure you put your turtlebot back on its base station so that it can charge its battery (base station green or blinking green). Switch the turtlebot base off and make sure the laptop is plugged and suspended.

Your turtlebot is yours for the semester, please be careful, considerate and gentle with it.