Fall 2019
CS 480/580: Intelligent Mobile Robotics
Assignment 4
Due Date: 11/15, 11:59PM

GOAL

Students will learn to write code in ROS that based on visual input (RGB-D) to control robot platforms.

For technical questions, please first post on Piazza and leave it there for at least 24 hours. You are indirectly helping other people by doing so. If no one knows the answer, please email the instructor.

ACCESS TO ROBOTS

To complete this assignment, students will need to work on real robots (Turtlebots) that are available in the teaching lab. Due to the limited number of robots, we have to distribute the resources in a first-come-first-serve manner.

The robots are only used for testing, i.e., software development should be done offline using your own laptop/desktop machines. You may want to host your code in the cloud (github, bitbucket, etc), so that you can work on it anytime anywhere.

Students will use their own account – don't share accounts!

AFTER USING A ROBOT

After your use of a robot, make sure to turn off both the laptop and the robot base, and put both back for charging. This is very important to the next student who uses the robot.

INSTRUCTIONS

Students will create code to enable the robot to follow ball in red (or any specific) color while keeping a distance of 1m from the ball.

1. The following command will start the driver of Turtlebot base:

```
roslaunch turtlebot_bringup minimal.launch --screen
```

2. The following command activates the rgb camera. After running this command, a window will pop up showing the rgb frames.

```
roslaunch usb_cam usb_cam-test.launch -screen
```

It happens that the laptop's camera is used (instead of the Astra camera). In this case, the following parameter needs to be changed to video0 or video1, whichever works, in the launch file of

```
usb_cam-test.launch
<param name="video_device" value="/dev/video1" />
```

3. The following command starts the depth camera:

```
roslaunch astra_launch astra.launch --screen
```

4. You can use the tool of image_view to see if the rgb and/or depth cameras work or not.

```
rosrun image_view image_view image:=/topic/of/images
```

5. Finally, students will create code to, based on the rgb and depth frames, drive the robot platform to follow a ball of red color while maintaining a distance of one meter.

6. To help students get used to processing rgb and depth frames, the instructor created a ROS package that finds the "most red" pixel in RGB frames while drawing a circle around it. The code is available on Github:

https://github.com/shiqizhang6/robotics_teaching.git

7. The following video shows the expected behaviors of your robot (While the code associated to the video is available, please be cautious that the code is very outdated and starting from there may require more effort than developing your own code directly):

https://youtu.be/FO-vBwJLUiY

8. Bonus points (of up to 20%) will be given, if the student' solution does not require the RGB stream of the visual sensor. This can be easily tested by covering the RGB camera, turning off the light, or changing the ball color.

   Please let the instructor and the TA know, if your code does not require the RGB sensor readings.

WHAT TO TURN IN

Students will submit a Youtube link (in the readme file) that points to a video showing the robot follows a ball while maintaining a 1-meter distance.

Students also need to turn in a single file (in tar/zip/rar format), and name it as your last name (initial in uppercase) followed by the initial of your first name (uppercase). For example, the file name should be "ZhangS.tar" for the instructor.

IMPORTANT: Your program will (possibly) be run in two steps:

1. Place the compressed file under the path of 'catkin_ws/src' and extract the file there.
2. The following command (and **only this command**) will be run under an arbitrary directory, where the package name will be changed based on student names.

```
roslaunch ZhangS assign6.launch
```

Please do not expect the instructor to figure out the problems in your code (even if the fix can be very straightforward).

Upload your solution to Blackboard.

If you have any questions about this assignment, feel free to ask the instructor. Not strictly following this submission instruction will reduce your points by 20%.