ESE650 Project 4: SLAM

Due Date: ***3/22/2016 at 1:20pm*** *on Canvas, and in class*

In this project, you will implement the structure of mapping and localization in an indoor environment using information from an IMU and range sensors. You will integrate the IMU orientation and odometry information from a walking humanoid with a 2D laser range scanner (LIDAR) in order to build a 2D occupancy grid map of the walls and obstacles in the environment. After this, you will then integrate additional camera and depth imagery from a Kinect One sensor to build a textured map. You will first implement your 2D localization and mapping system before moving on to the RGBD part. Training sets of odometry, inertial, and range measurements from a THOR-OP humanoid robot will be provided for this project.

1. DOWNLOAD DATA. A set of IMU data will be available for you at https://upenn.box.com/2016ese650-project4-train.These data sets contain timestamped sensor values, corresponding to the raw sensor readings. Download these files and be sure you can load and interpret the file formats. You can find the 'config.pdf' file that explains about the data. If we need to provide additional information, we will announce them on Piazza. Please read or make a post on Piazza, if you need to ask questions regarding the data.

2. SLAM: First, you should run your robot using pure walking odometry measurements and yaw gyro readings. Make a 2D map using this data before correcting using range readings. You should then be able to provide a visualization of the motion of the robot within a 2D map. Up to this point, you will have done the first phase of this project. Next, you will need to simultaneously localize the robot pose, and construct the surrounding 2D map using a pose filter and occupancy grid algorithm. Some cpp files will be provided to help you experiment with 2D LIDAR scan matching. (You can convert them to Matlab mex files.)

3. TEXTURED MAP : The second portion of this project will be to integrate the RGB and depth images from the Kinect to build a textured visualization of the environment. You may work with dense point clouds, or with a sparser set of tracked features for this part. You will do this with pure odometry, and with your full SLAM routine.

4. TESTING. You should then make sure that your program can take input sensor readings from unknown environment. We will release the test data in the morning in order to give some time to run your program with test data. You should be able to show your results with proper visualization from an unknown test dataset.

5. SUBMISSION. You will upload to Canvas a written description of your algorithm in PDF form and a zip file of your code. Use the naming convention "project4_[YourPennKeyName].pdf" and "project4_[YourPennKeyName].zip".
   Please do include your result map images, but do not include data when you submit.

6. PRESENTATION. For the presentation in class you are expected to bring your own laptop or use the classroom computer. The classroom computer has MATLAB installed but no compatibility is guaranteed. The projector has a VGA port and you may need a VGA adaptor for your laptop. During the presentation, you will be asked to present your algorithm and run your code on a set of test data. The test data will be released both online and on a USB flash disk prior to the presentations. Clearly presenting your approach and having good algorithm performance are equally important.