ME 133(b) Lab #3: 1 - Mapping & SLAM

Prelab

Released: Thursday, 02/28/2019 Due: Wednesday, 03/13/2019

In Lab #3, you will learn how to construct an occupancy grid map, and perform simultaneous localization and mapping (SLAM). You will first use *OctoMap,* a public domain program to build a 3-dimensional occupancy map of a terrain. In this case, the terrain will be the CAST flight arena (with some obstacles included). The theory follows closely from our class review of occupancy grids. Details of the algorithm can be found at the OctoMap web site, introduced below. Then you will use the *gmapping* package/procedure to build a 2-dimensional occupancy grid, but also perform simultaneous localization. You will compare the accuracy of gmapping’s localization process with the ground truth recorded by OptiTrack.

This prelab is aimed at getting you familiar with the Velodyne Lidar and the OctoMap system. You will set up the OctoMap program and Velodyne Lidar in ROS, and the apply these methods to a bag file.

# Building an Occupancy Grid using OctoMap (30 Points)

**Warm-Up.** To start this prelab, please go the the OctoMap *GitHub* site: <https://octomap.github.io/> . If you want to dive in deeply, you can read the original paper that describes the theory behind OctoMap. But at a minimum, you need to read the basic documentation, and in particular, the “OctoMap in ROS” page ( <http://wiki.ros.org/octomap> ).

* 1. After understanding the basic OctoMap operation and ROS package, look at the *Installation Instructions* page. The *octomap\_mapping* stack link (<http://wiki.ros.org/octomap_mapping> ) describes the simplest way to download the key packages and libraries. Make sure that you look at the *octomap\_server* documentation ( <http://wiki.ros.org/octomap_server> )
  2. Note that the cloud\_in topic will need to be remapped to the velodyne\_pointcloud topic. See details below.
  3. Next you will need to download the Velodyne driver. You’ll want to start by reading the <http://wiki.ros.org/velodyne/Tutorials/Getting%20Started%20with%20the%20Velodyne%20VLP16> ROS “Getting Started with Velodyne VLP-16”. Pay particular attention to Installation items 2 and 3 (“Installing ROS dependencies” and “Installing the VLP16 driver”).
  4. Next download a ROS Bag file of Velodyne data, which will be used for this pre-lab. The [Husky bagfile](https://drive.google.com/file/d/1Vx2cJNFkb3UV9uwk3fyuSGDwUrmhBuaM/view?usp=sharing) can be found at this link. A link can also be found on the course web site. This data was gathered by a *Husky* robot mounted with a velodyne lidar. The velodyne data (and all the other Husky commands being published) are included in the bag file. You should look for the topics /velodyne\_points and /tf (for transforms).
  5. In the [octomap\_server](http://wiki.ros.org/octomap_server) package, look at the [octomap\_mapping.launch](https://github.com/OctoMap/octomap_mapping/blob/kinetic-devel/octomap_server/launch/octomap_mapping.launch) file, and change the frame\_id to the appropriate fixed frame and remap cloud\_in to velodyne\_points
  6. You should now be able to use RViz to visualize the octomap made from the Velodyne data by visualizing /occupied\_cells\_vis\_array.

**Deliverable.** To complete this prelab, capture a rendering of the OctoMap output, and submit this picture to the lab T.A.s during your laboratory session.