Assignment 2

1. Introduction

The primary goal of this assignment is to work on the problem of mapping, we will use an occupancy grid as the underlying map structure. In addition you will have to implement a

A change from the last assignment, please create a directory under **catkin_ws/src** using the following naming convention **LastnameInitial** for example for the instructor the directory should be:

>catkin_ws/src/RekleitisI/

Move your random_walk node under that directory and also unzip the grid_mapper under the same directory:

>catkin_ws/src/RekleitisI/random_walk
>catkin_ws/src/RekleitisI/grid_mapper

Compile your code using catkin_make and ensure to run the command from the ROS tutorial:

>. ~/catkin_ws/devel/setup.bash

to ensure ROS can find your executables.

2. (70%) Occupancy grid mapping

Create a program that would implement Occupancy Grid Mapping. Use the random walk implemented in question 1 to have a single robot wander around the environment. Collect the laser data and use them to implement an occupancy grid. Use the provided code as a guideline on how to store and display an occupancy grid in the form of an image. In addition display the path of the robot. Remember to consider *Occupied*, *Unoccupied* and *Unknown* Cells.

Please find the complete package under grid_mapper.zip. Compile the code using catkin_make. The grid_mapper code can be run as:

>rosrun grid_mapper grid_mapper 60 60

The last two parameters indicate the dimensions of the world.

3. (30%) Particle Filter Propagation

Using the example code we produced in class implement the following behaviours:

- Initialize a particle filter:
  - At a fixed position (0,0) with random orientation
  - Uniformly over a 10 m by 10 m area with random orientation
  - Around a fixed position (0,0) but with uncertainty along the X and Y axis following a Normal distribution with sigma=0.3 m. Fixed orientation at 90 degrees.
- Propagate for 100 sec discretized at 1 sec intervals for the following motion strategies
  - Move with fixed angular velocity V=2 m/sec and angular velocity w=0.1 rad/sec. No noise
  - Move with fixed angular velocity V=2 m/sec and random angular velocity w ranging between -0.2 and 0.2 rad/sec (w=[-0.2,0.2] rad/sec). Select a new angular velocity every 10 sec. Linear velocity is corrupted with zero mean Gaussian noise with sigmaV=0.1 m/sec. Angular velocity is corrupted with zero mean Gaussian noise with sigmaW=0.001 m/sec.

No obstacles in the environment, the robot moves through free space. The above should produce six experiments; prepare a report with sample images for the six behaviours. Clearly identify which screenshot corresponds with which behaviour; add a short comment on what you observed.

You can place your code and the report in a separate directory named ParticleFilter.

>LastnameInitial/ParticleFilter

Note: Refer to the particle filter tutorial located at:

http://www.cim.mcgill.ca/~yiannis/ParticleTutorial.html

4. (20%) For Graduate and Honour Students (bonus for the rest of the class) Occupancy grid mapping using real data

Use the provided logged data publisher instead of a stageros simulation to construct and display an occupancy grid. The data are from an early experiment with a hokuyo laser range finder mounted on top of a TurtleBot 2 robot. The robot pose as published comes also from real data and thus is corrupted by odometric error. In addition display the path of the robot.

The real data are store in a bag file. Run:

> rosbag play left-corridor-mapping.bag

Use the rostopic list command to see the topics published during the play of the bag. The topics of interest for the grid mapper are:

> \odom

or

> \odom_combined
> \scan

Run the mapper for the two different odometry measurements and comment on the differences.

Note:

There is a difference between image coordinates and Stage coordinates. See sample code and implement accordingly.
You need to have opencv 2.x and some boost packages installed. For the sample code:

- press SPACEBAR to save snapshot of occupancy grid canvas into folder where user opened the executable in.
- press X or x to quit.
- read over the code, and pay attention to the 2 TODO comments especially

Other new things in the project, i.e. why did I have to include a zip of the entire project:

- CMakeLists.txt has some lines needed to link project with Boost and OpenCV libraries
- package.xml has ros_dep line that asks project to depend on EXTERNALLY INSTALLED OpenCV library (recommended by ROS, see http://www.ros.org/wiki/opencv2)

For all questions prepare a written report with a screen (images saved in your program; see attached code) that document the progress of the mapping.

What to submit:
Prepare a report presenting screenshots of your work, provide a brief overview on how the program works and also discuss the choices made. Place the report in the base directory with your name and zip everything into a single file. Submit in the dropbox.