

# 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/Rekleitisl/
```

Move your random\_walk node under that directory and also unzip the grid\_mapper under the same directory:

```
>catkin_ws/src/Rekleitisl/random_walk  
>catkin_ws/src/Rekleitisl/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. (100%) 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:

```
>roslaunch grid_mapper grid_mapper 60 60
```

The last two parameters indicate the dimensions of the world. This is just an example scale your map to look similar to the stage window.

### **3. (20%) Bonus: 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.

Additional Comment: Use a line drawing algorithm to mark unoccupied space, for example the Bresenham's line drawing algorithm.

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.