Assignment 4.2

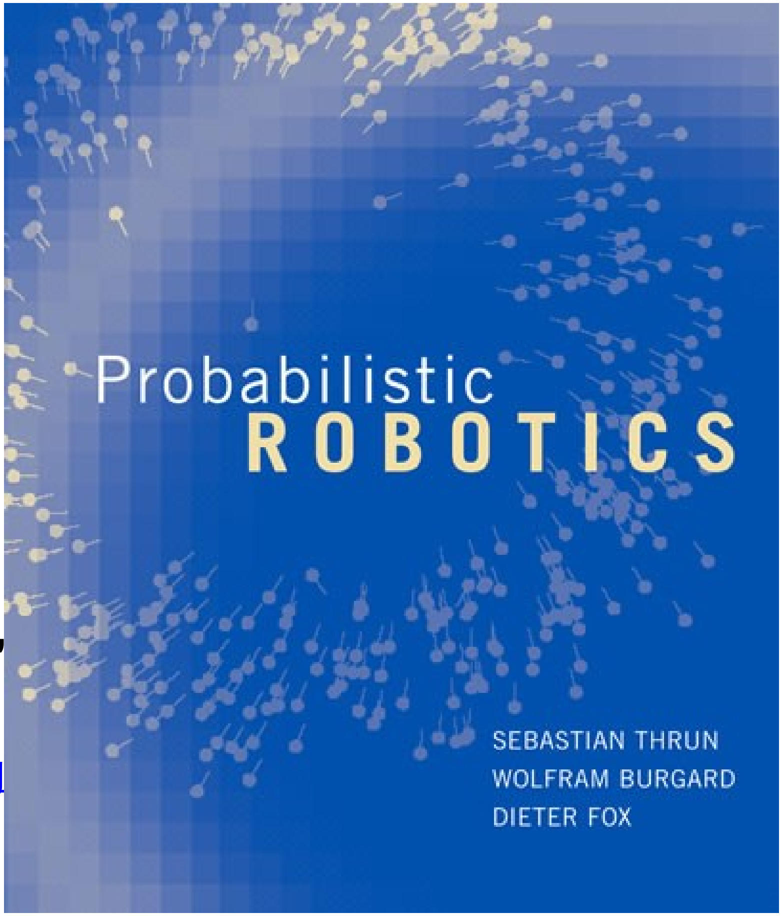
Mapping with Known Poses

Technische Universität Berlin

Recommended Book

Probabilistic Robotics

- Sebastian Thrun, Wolfram Burgard, Dieter Fox
- MIT Press, 2005



Some slides have been adapted from the course “**Introduction to Mobile Robotics”** at Universität Freiburg: http://ais.informatik.uni-freiburg.de/teaching/ss11/robotics/

Prob.Rob. pp. 281

Why Mapping?

- Learning maps is one of the fundamental problems in mobile robotics

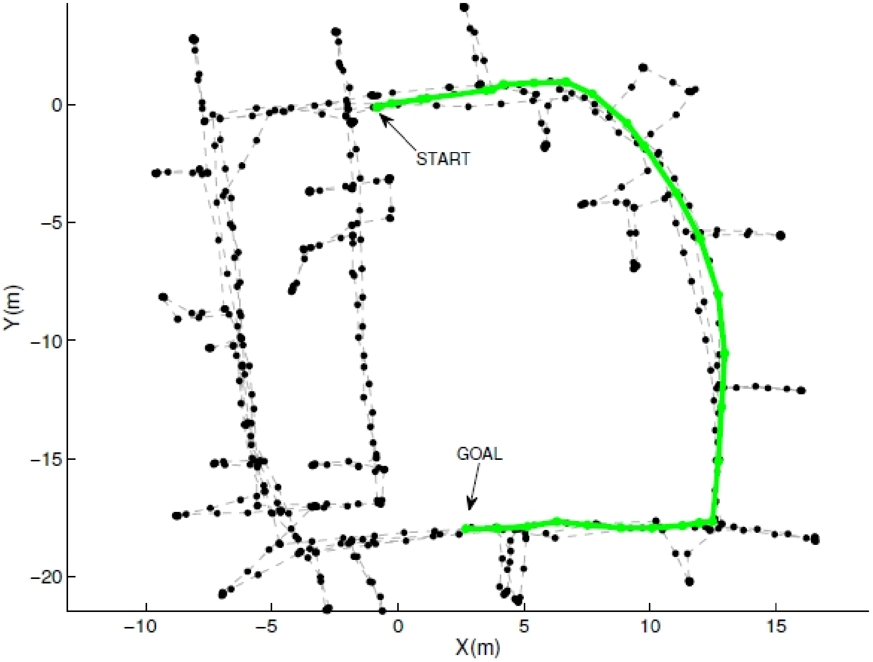
- Maps allow robots to efficiently carry out their tasks, allow localization …

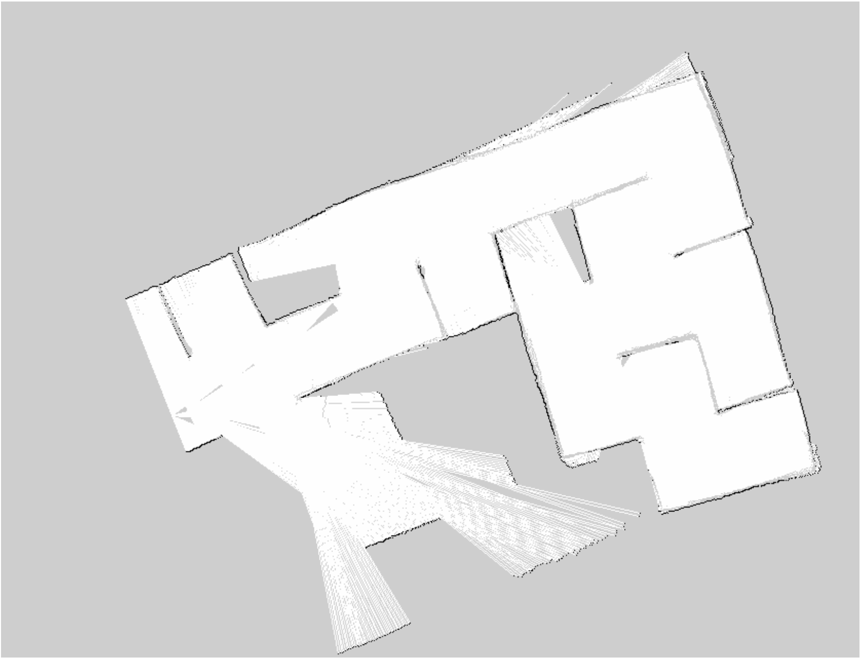
- Successful robot systems rely on maps for localization, path planning, activity planning etc.

Mapping as a Chicken and Egg Problem

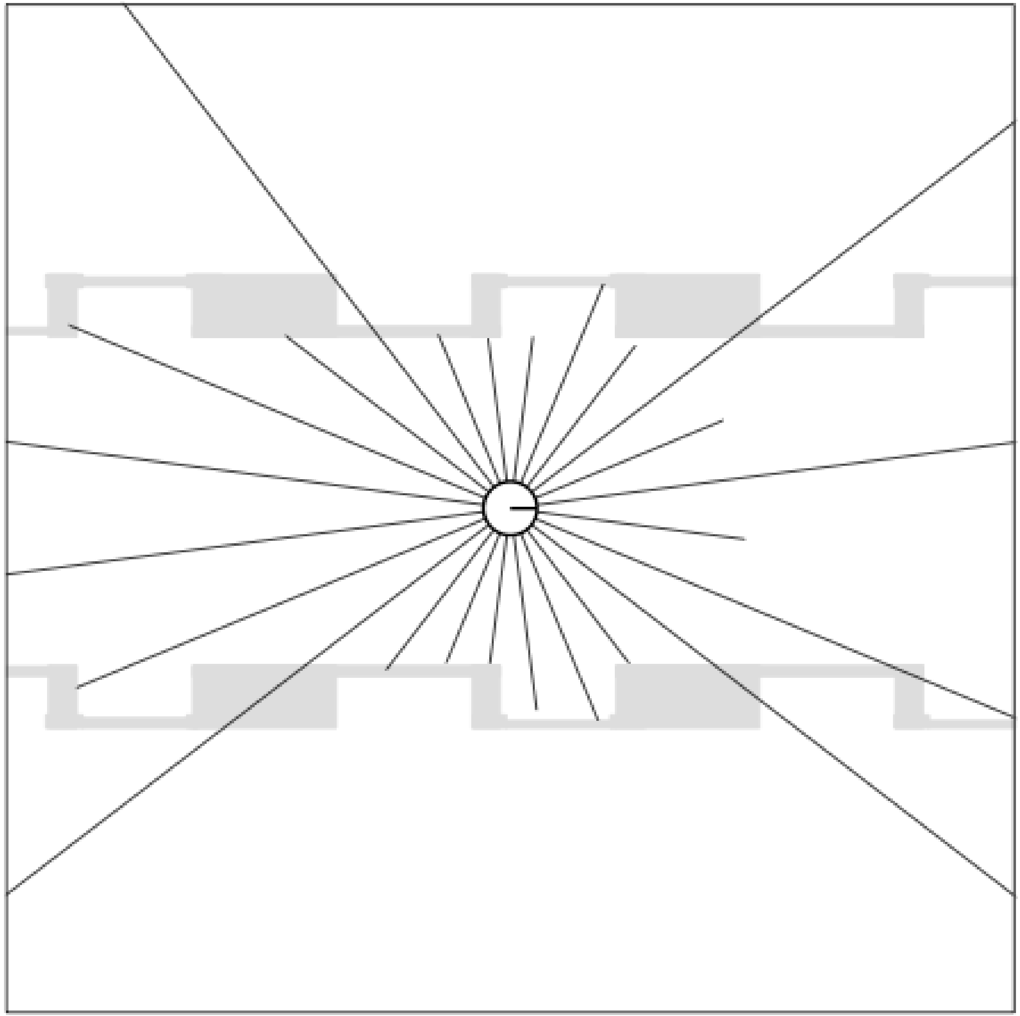
- Mapping involves to simultaneously estimate the pose of the vehicle and the map.
- The general problem is therefore denoted as the *simultaneous localization and mapping problem* (SLAM).
- Throughout this section we will describe how to calculate a map given we know the pose of the vehicle.

Feature/Landmark-based vs. location-based maps

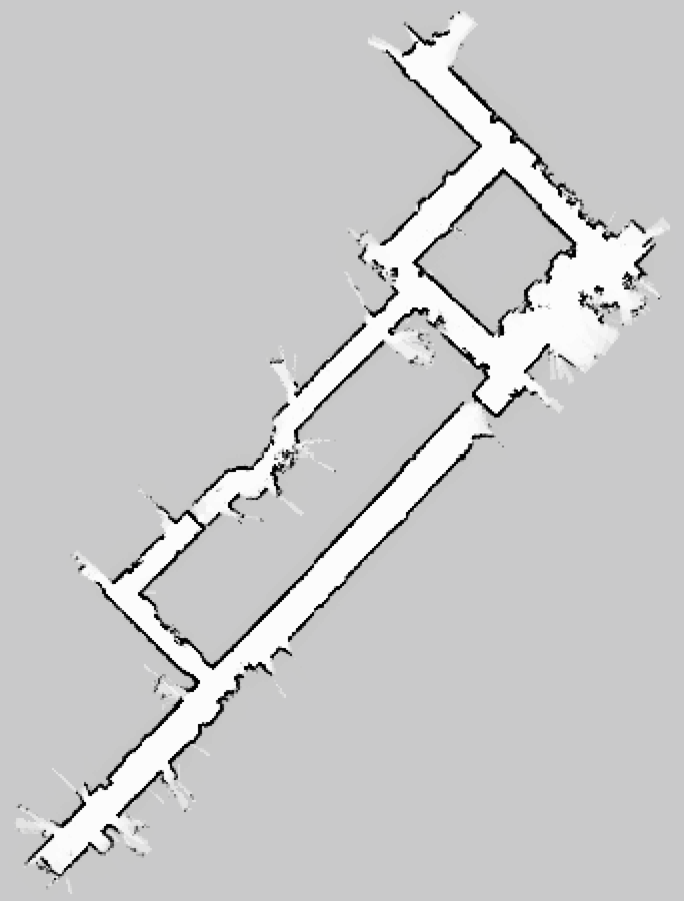




We Need Sensors!



Occupancy Grid Maps



- Discretize the world into equally spaced cells
- Each cell stores the probability that the corresponding area is occupied by an obstacle
- Each cell is assumed to be independent from all others:

$$bel(m_t) = \prod_{x,y} bel(m_t[x, y])$$

- If the pose of the robot is known, mapping is easy!

Simple Counting Method (what we'll implement)

- For every cell count
  - *hits(x,y)*: number of cases where a beam ended at *(x,y)*
  - *misses(x,y)*: number of cases where a beam passed through *(x,y)*

$$bel(m[x,y]) = \frac{hits(x,y)}{hits(x,y) + misses(x,y)}$$

- Results in reflection map (≠ occupancy map)

ASSIGNMENT

Preliminaries

- Download the workspace from ISIS
  - Dell laptops: /home/create/ws_assignment4/
- Build the localization node, mapping node and the visualization nodes:

```
$ cd  /home/create/ws_assignment4/
$ catkin_make
```

ROS Launch File

```
roslaunch mapping mapping.launch
```

- Launches:
  - **mapper**: node that you have to implement
  - **map_view**: visualization tool
  - **rosbag**: replays recorded test data