Mapping

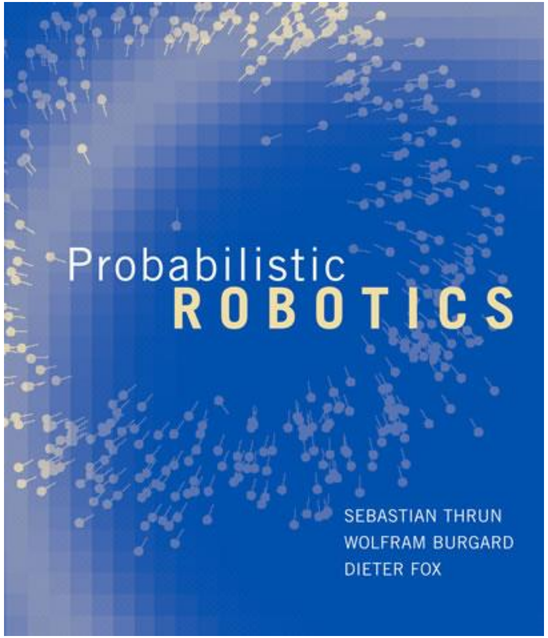
Manuel Baum

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/

Why Mapping?

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

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

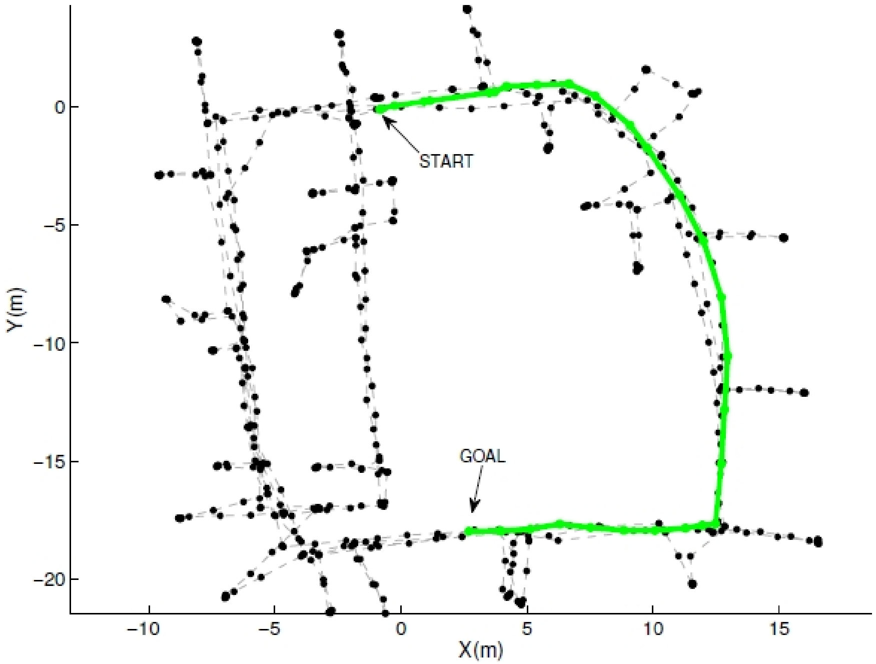
- Successful robot systems rely on maps for localization, path planning, activity planning etc.

Mapping as a Chicken and Egg Problem

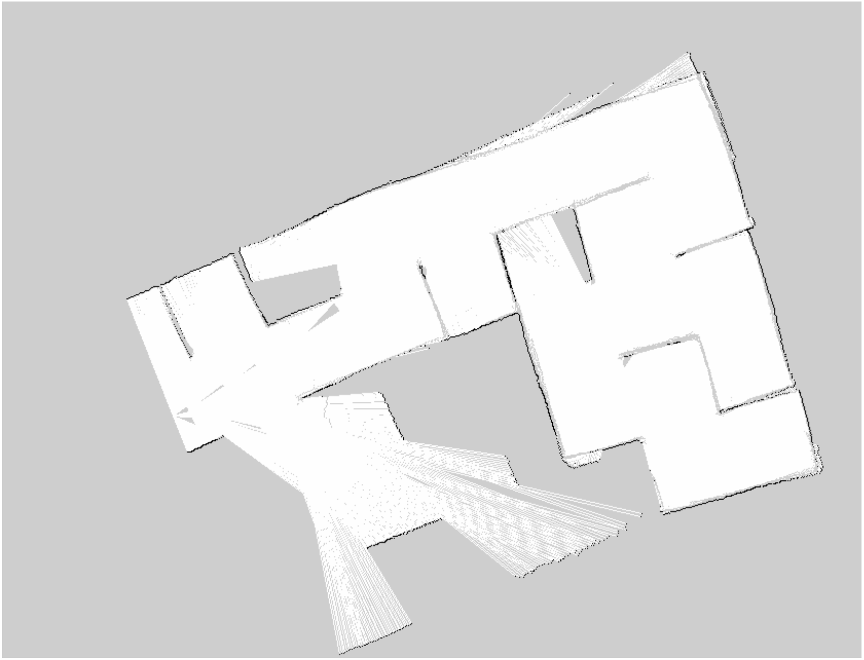
- Mapping involves to simultaneously estimate the pose of the vehicle and the map.
- The general problem is therefore denoted as the *simultaneous localization and mapping problem* (SLAM).
- Throughout this section we will describe how to calculate a map given we know the pose of the vehicle.

Different kinds of maps

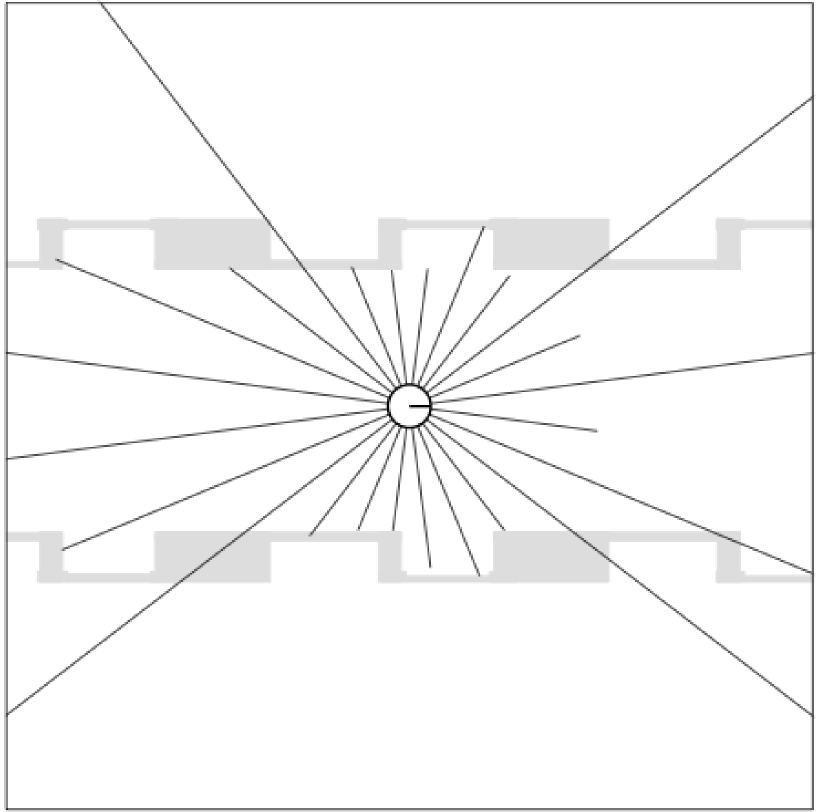
Feature/Landmark-based



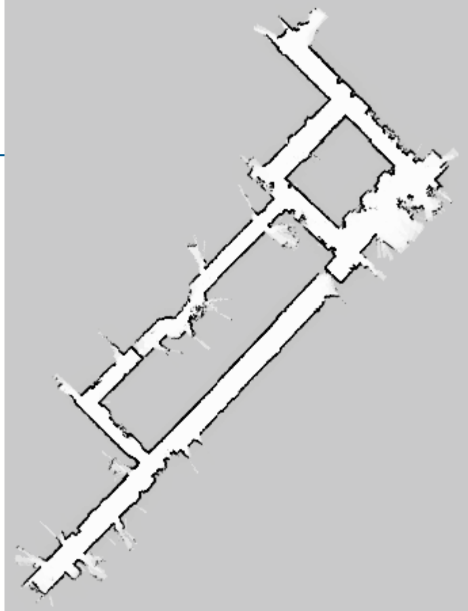
Occupancy Grid



We need sensors!



Occupancy Grid Maps



- Discretize world into (equally spaced) cells
- Each cell stores probability that area is occupied by an obstacle
- Cells are assumed to be independent from each other

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

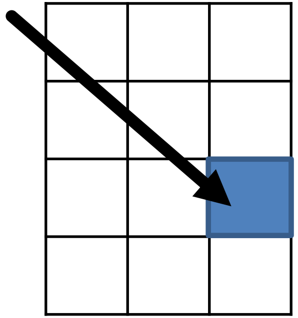
- If pose of robot is known, then mapping is easy!

Simple Counting Method (what we'll implement)

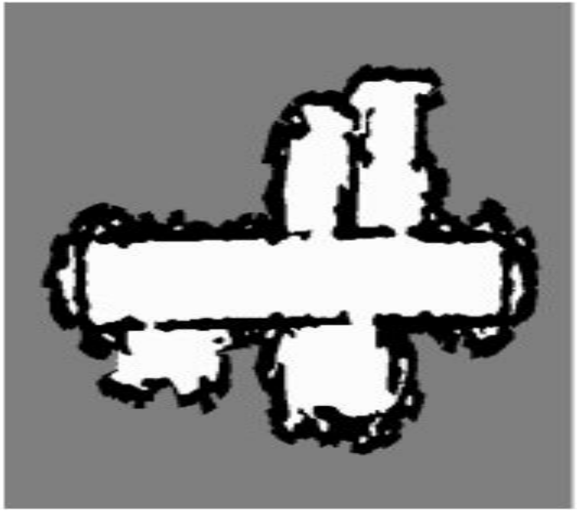
- For every cell count
  - *hits(x,y)*: number of beams that ended at *(x,y)*
  - *misses(x,y)*: number of cases when a beam passed *(x,y)*

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

- Results in reflection map / occupancy map



Getting the Maximum Likelihood Map



The maximum likelihood map is obtained by clipping the occupancy grid map at a threshold of 0.5

Running the Assignment Code

- Download workspace from ISIS
- Build the mapping node, visualization node (and localization node)

```
$ cd /home/username/workspace_assignment4/
$ catkin_make
```

- Source the workspace [better: put into .bashrc]

```
$ source devel/setup.bash
```

- Launch the mapping node

```
$ roslaunch mapping mapping.launch
```

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

What the result should look like