Mobile Robotics, Localization: Occupancy Grid Maps

Material based on the book Probabilistic Robotics (Thrun, Burgard, Fox) [PR]; Chapter 4.2, 9.1, 9.2

Part of the material is based on lectures from Cyrill Stachniss

Summary

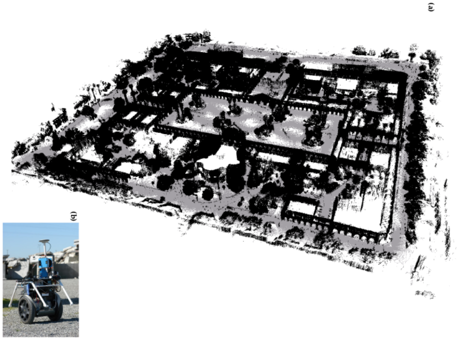
- Introduction to mapping
- Occupancy grid maps [Chapter 9.1]
- Static-State Binary Bayes Filter [Chapter 4.2]
- Occupancy Grid Mapping algorithm [Chapter 9.2]

Introduction to mapping

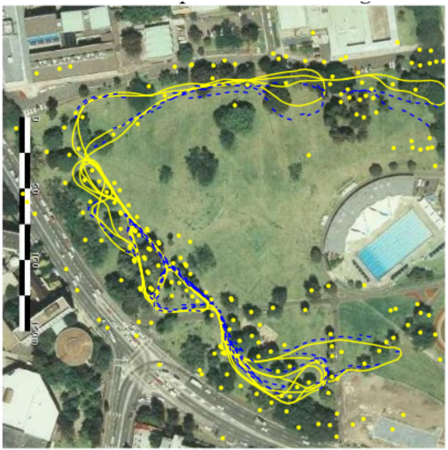
Mapping

- Map: representation of the environment around the robot
- Maps required for most tasks:
  - Localization
  - Obstacle avoidance, Path-Planning
  - Activity planning
  - ...
- Building maps is a fundamental task for robotics

Map representations: features vs. volumetric



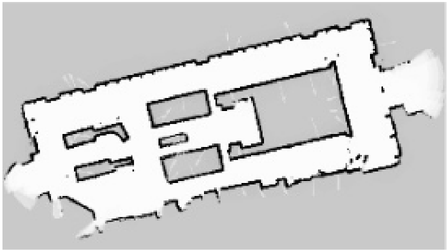
volumetric map of Standford campus (source [PR], courtesy: Montemerlo)



feature map of Victoria Park (source [PR], courtesy: Nebot)

Occupancy grid maps

Raw range data, position by odometry (source [PR]



Occupancy Grid Map (source [PR])

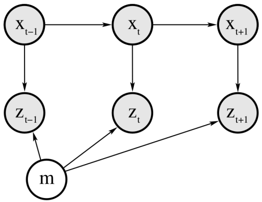
Mapping task

- Compute the most likely map given a stream of sensor data

$$m^* = \arg\max_m P(m | u_{1:t}, z_{1:t})$$

- Mapping with **known poses**

$$m^* = \arg\max_m P(m | x_{1:t}, z_{1:t})$$



Graphical model of mapping with known poses (source [PR])

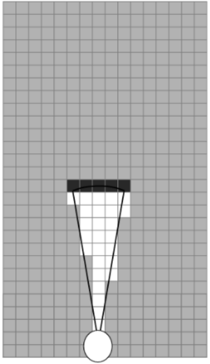
Grid maps

- Map models the occupancy of the space
- Represent the environment by using a grid
  - Large maps may require significant memory (particularly in 3D)
  - trade-off with resolution
  - but they do not rely on a feature detection
- Each cell can be either **occupied** or **free**
- Non-parametric model

Occupancy Probability Grid Maps

- **Assumption 1**: The area of a cell is either completely free or completely occupied

- Every cell is a **binary** random variable
- m_i a single cell
- $P(m_i)$ probability that cell m_i is **occupied**
  - Cell is occupied $P(m_i) = 1$
  - Cell is not occupied $P(m_i) = 0$
  - **No Knowledge** $P(m_i) = 0.5$



Portion of occupancy grid map (source [PR])

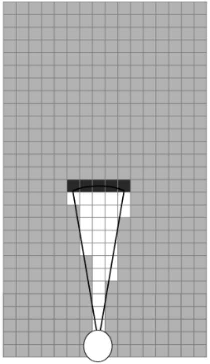
Occupancy Probability Grid Maps

◇ **Assumption 2**: the environment is **static**

- an occupied cell remains occupied, a free cell remains free

◇ **Assumption 3**: the random variables representing cells are **independent** of each other

- dramatically simplifies the model



Portion of occupancy grid map (source [PR])

Joint Probability Distribution

- Joint probability distribution for a map configuration

$$P(m) = P(m_1, m_2, \ldots, m_N)$$

- Exploiting independence we can decompose the joint probability into a product

$$P(m) = \prod_{i=1}^{N} P(m_i)$$

Estimating a Map from data

- Given sensor data $z_{1:t}$ and robot poses $x_{1:t}$
- Estimate the posterior probability for the map

$$P(m|z_{1:t}, x_{1:t}) = \prod_{i=1}^{N} P(m_i|z_{1:t}, x_{1:t})$$

- Need to estimate the value for each binary random variable based on measurements
- **Idea**: use a Binary Bayes filter (for a static state)

Static State Binary Bayes Filter I

$$P(m_i|z_{1:t}, x_{1:t}) = \frac{P(z_t|m_i, z_{1:t-1}, x_{1:t})P(m_i|z_{1:t-1}, x_{1:t})}{P(z_t|z_{1:t-1}, x_{1:t})}$$ **Bayes Rule**

$$P(m_i|z_{1:t}, x_{1:t}) = \frac{P(z_t|m_i, x_t)P(m_i|z_{1:t-1}, x_{1:t-1})}{P(z_t|z_{1:t-1}, x_{1:t})}$$ **Markov** + **Indep.**

$$P(z_t|m_i, x_t) = \frac{P(m_i|z_t, x_t)P(z_t|x_t)}{P(m_i|x_t)}$$ **Bayes rule** on first term of right hand equation

$$P(m_i|z_{1:t}, x_{1:t}) = \frac{P(m_i|z_t, x_t)P(z_t|x_t)P(m_i|z_{1:t-1}, x_{1:t-1})}{P(m_i|x_t)P(z_t|z_{1:t-1}, x_{1:t})}$$ **Substitute**

$$P(m_i|z_{1:t}, x_{1:t}) = \frac{P(m_i|z_t, x_t)P(z_t|x_t)P(m_i|z_{1:t-1}, x_{1:t-1})}{P(m_i)P(z_t|z_{1:t-1}, x_{1:t})}$$ **Indep.**

Static State Binary Bayes Filter II

- Same derivation but considering the probability of cell being free $p(\neg m_i)$

$$P(\neg m_i|z_{1:t}, x_{1:t}) = \frac{P(\neg m_i|z_t,x_t)P(z_t|x_t)P(\neg m_i|z_{1:t-1},x_{1:t-1})}{P(\neg m_i)P(z_t|z_{1:t-1},x_{1:t})}$$

- Compute the **ratio** of both probabilities we have

$$\frac{P(m_i|z_{1:t},x_{1:t})}{P(\neg m_i|z_{1:t},x_{1:t})} = \frac{\frac{P(m_i|z_t,x_t)P(z_t|x_t)P(m_i|z_{1:t-1},x_{1:t-1})}{P(m_i)P(z_t|z_{1:t-1},x_{1:t})}}{\frac{P(\neg m_i|z_t,x_t)P(z_t|x_t)P(\neg m_i|z_{1:t-1},x_{1:t-1})}{P(\neg m_i)P(z_t|z_{1:t-1},x_{1:t})}}$$

$$\frac{P(m_i|z_{1:t},x_{1:t})}{P(\neg m_i|z_{1:t},x_{1:t})} = \frac{P(m_i|z_t,x_t)P(m_i|z_{1:t-1},x_{1:t-1})P(\neg m_i)}{P(\neg m_i|z_t,x_t)P(\neg m_i|z_{1:t-1},x_{1:t-1})P(m_i)}$$

$$\frac{P(m_i|z_{1:t},x_{1:t})}{1-P(m_i|z_{1:t},x_{1:t})} = \frac{P(m_i|z_t,x_t)}{1-P(m_i|z_t,x_t)} \frac{P(m_i|z_{1:t-1},x_{1:t-1})}{1-P(m_i|z_{1:t-1},x_{1:t-1})} \frac{1-P(m_i)}{P(m_i)}$$

From ratio to probability I

- Odds for an event: ratio between the event happening vs. not happening
- can be used to turn the ratio we compute to probability
  - $Odds(x) = \frac{P(x)}{1-P(x)}$
  - $P(x) = Odds(x) - Odds(x)P(x)$
  - $P(x)(1 + Odds(x)) = Odds(x)$
  - $P(x) = \frac{Odds(x)}{1+Odds(x)}$
  - $P(x) = \frac{1}{1+\frac{1}{Odds(x)}}$

From ratio to probability II

- Using $P(x) = [1 + Odds(x)^{-1}]^-1$ we get:

$$P(m_i|z_{1:t}, x_{1:t}) = [1 + Odds(m_i|z_{1:t}, x_{1:t})^{-1}]^{-1}$$

$$P(m_i|z_{1:t}, x_{1:t}) = \left[1 + \frac{1-P(m_i|z_t,x_t)}{P(m_i|z_t,x_t)} \frac{1-P(m_i|z_{1:t-1},x_{1:t-1})}{P(m_i|z_{1:t-1},x_{1:t-1})} \frac{P(m_i)}{1-P(m_i)}\right]^{-1}$$

Log odds notation

- To improve efficiency usually log odds is used

$$l(x) = \log\left(\frac{P(x)}{1-P(x)}\right)$$

- we can retrieve $P(x)$ as

$$P(x) = 1 - \frac{1}{1+\exp l(x)}$$

- Going back to our map estimation:

$$\frac{P(m_i|z_{1:t},x_{1:t})}{1-P(m_i|z_{1:t},x_{1:t})} = \frac{P(m_i|z_t,x_t)}{1-P(m_i|z_t,x_t)} \frac{P(m_i|z_{1:t-1},x_{1:t-1})}{1-P(m_i|z_{1:t-1},x_{1:t-1})} \frac{1-P(m_i)}{P(m_i)}$$

$$l_{t,i} = l(m_i|z_{1:t},x_{1:t}) = log\left(\frac{P(m_i|z_{1:t},x_{1:t})}{1-P(m_i|z_{1:t},x_{1:t})}\right)$$

Occupancy mapping in Log odds form

- The product turns into a sum

$$\frac{P(m_i|z_{1:t},x_{1:t})}{1-P(m_i|z_{1:t},x_{1:t})} = \frac{P(m_i|z_t,x_t)}{1-P(m_i|z_t,x_t)} \frac{P(m_i|z_{1:t-1},x_{1:t-1})}{1-P(m_i|z_{1:t-1},x_{1:t-1})} \frac{1-P(m_i)}{P(m_i)}$$

$$l(m_i|z_{1:t}, x_{1:t}) = l(m_i|z_t, x_t) + l(m_i|z_{1:t-1}, x_{1:t-1}) - l(m_i)$$

- More compact form

$$l_{t,i} = \text{inverse\_sensor\_model}(m_i, z_t, x_t) + l_{t-1,i} - l_0$$

Occupancy mapping algorithm

- Highly efficient, just need to compute sums

Algorithm occupancy_grid_mapping($\{l_{t-1,i}\}, x_t, z_t$)**:**

    for all cells $\mathbf{m}_i$ do

        if $\mathbf{m}_i$ in perceptual field of z_t then

            $l_{t,i} = l_{t-1,i} + \textbf{inverse\_sensor\_model}(\mathbf{m}_i, x_t, z_t) - l_0$

        else

            $l_{t,i} = l_{t-1,i}$

        endif

    endfor

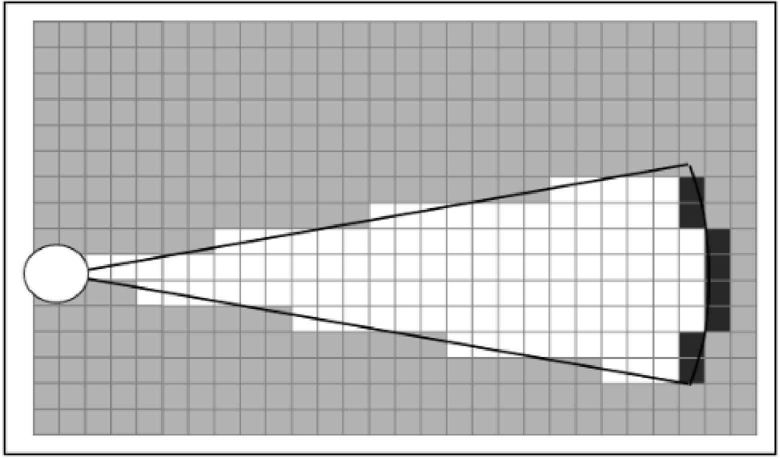
    return $\{l_{t,i}\}$

Source [PR]

Inverse sensor model for laser range finder

- Laser is extremely precise
- Laser has a very high resolution typically more than the cell
- Overall Idea:
  - cell is occupied if a laser beam stops inside the cell
  - cell is free is a laser beam goes through the cell
  - cell is unknown otherwise

Inverse sensor model for sonar range sensors



Source [PR]

Inverse range sensor model algorithm

Algorithm inverse_range_sensor_model(i, x_t, z_t)**:**

  Let x_i, y_i be the center-of-mass of $\mathbf{m}_i$

  $r = \sqrt{(x_i - x)^2 + (y_i - y)^2}$

  $\phi = \text{atan2}(y_i - y, x_i - x) - \theta$

  $k = \text{argmin}_j \, |\phi - \theta_{j,\text{sens}}|$

  if $r > \min(z_{\max}, z_t^k + \alpha/2)$ or $|\phi - \theta_{k,\text{sens}}| > \beta/2$ then

    return l_0

  if $z_t^k < z_{\max}$ and $|r - z_{\max}| < \alpha/2$

    return l_{occ}

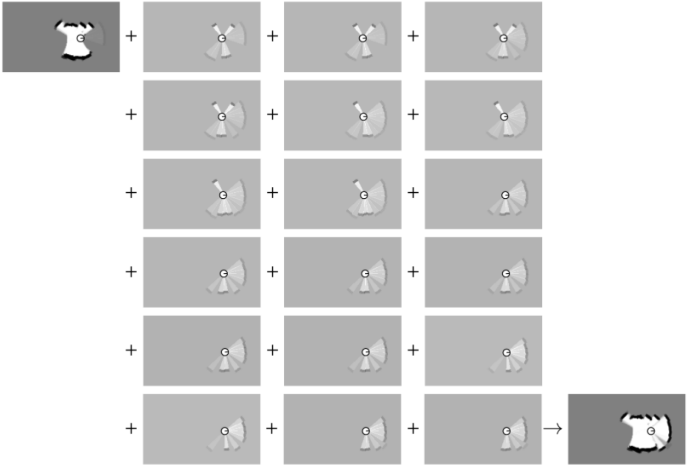
  if $r \leq z_t^k$

    return l_{free}

  endif

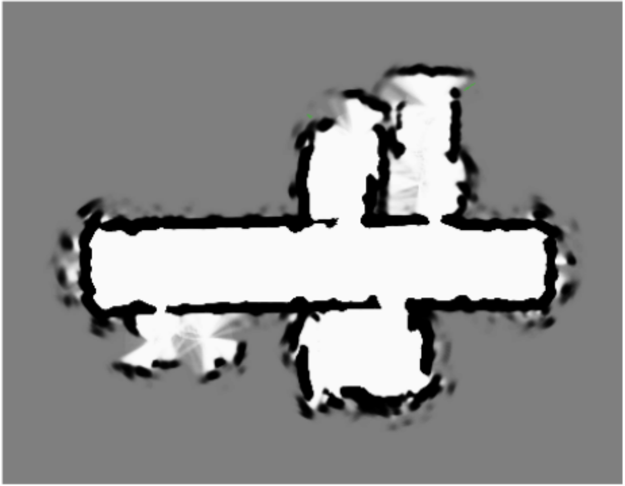
Source [PR]

Incremental update of occupancy grid with sonar



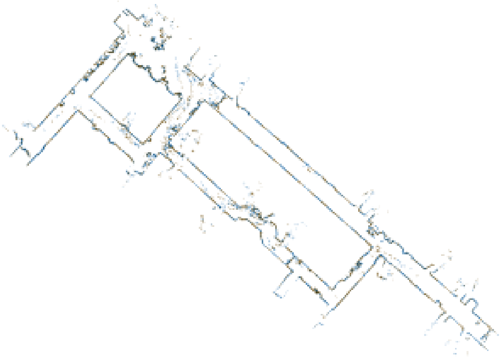
Source [PR], courtesy of Cyrill Stachniss

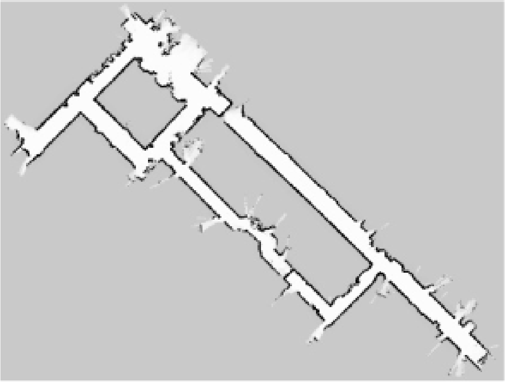
Resulting maps obtained with 24 sonar range sensors



Source [PR], courtesy of Cyrill Stachniss

Resulting maps obtained with laser range sensor





Source [PR], courtesy of Steffen Gutmann

Blueprint vs. Occupancy map

Source [PR]

Summary

- Occupancy grid maps discretize the environment into **independent** cells
- Each cell is modelled as a **binary** random variable
- Estimate maps using a static-state binary Bayes filter for each cell
- Mapping with known poses
- Log odds model very fast to compute
- no need of predefined features