Introduction to Mobile Robotics

Grid Maps and Mapping With Known Poses

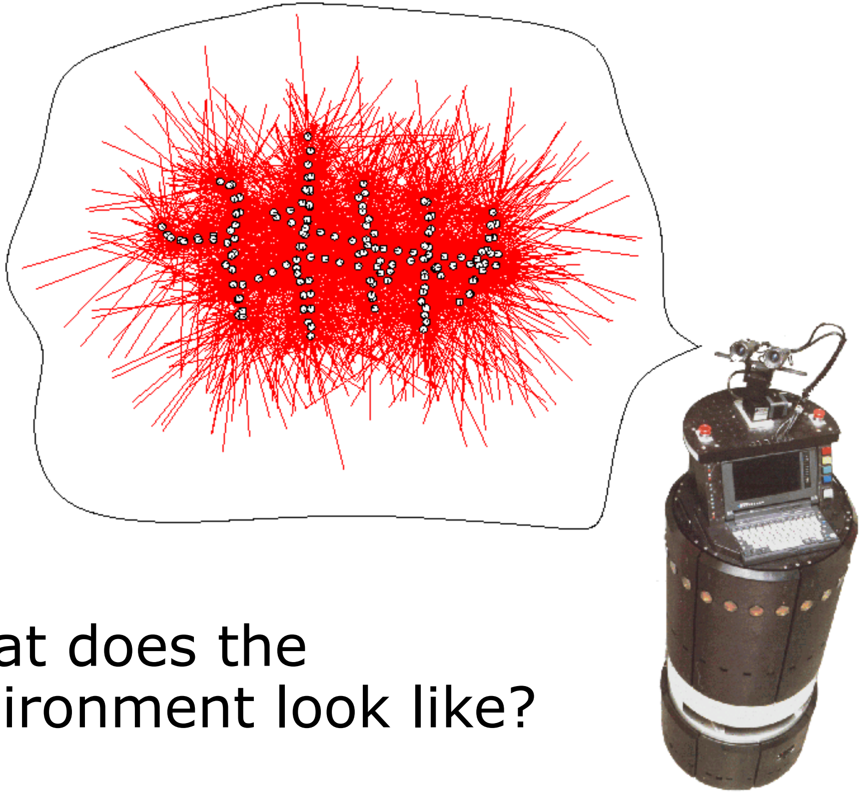
Wolfram Burgard, Tanja Kaiser

UTN

Why Mapping?

- Learning maps is one of the fundamental problems in mobile robotics.
- Maps allow robots to efficiently carry out their tasks, allow localization, path planning, and much more.
- Many successful robot systems and autonomous cars heavily rely on maps.

The General Problem of Mapping



What does the
environment look like?

The General Problem of Mapping

- Formally, mapping involves, given the sensor data

$$d = \{u_1, z_1, u_2, z_2, \ldots, u_t, z_t\}$$

- to calculate the most likely map

$$m^\star = \mathrm{argmax}_m P(m \mid d)$$

The General Problem of Mapping

Formally, mapping involves, given the sensor data

$$d = \{u_1, z_1, u_2, z_2, \dots, u_t, z_t\}$$

to calculate the most likely map

$$m^\star = \operatorname{argmax}_m P(m \mid d)$$

Today we describe **how to calculate a map given the poses $x_1, \dots, x_t$ of the robot**.

The General Problem of Mapping with Known Poses

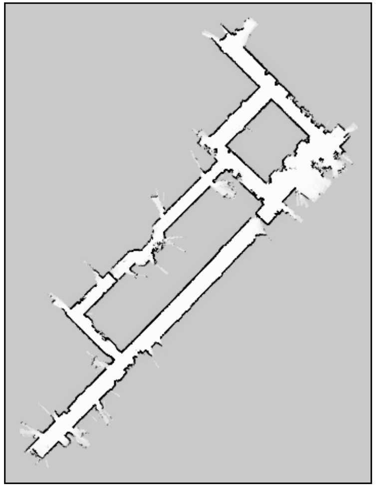
Formally, mapping with known poses involves, given the measurements $z_1, \ldots, z_t$ and the poses $x_1, \ldots, x_t$,

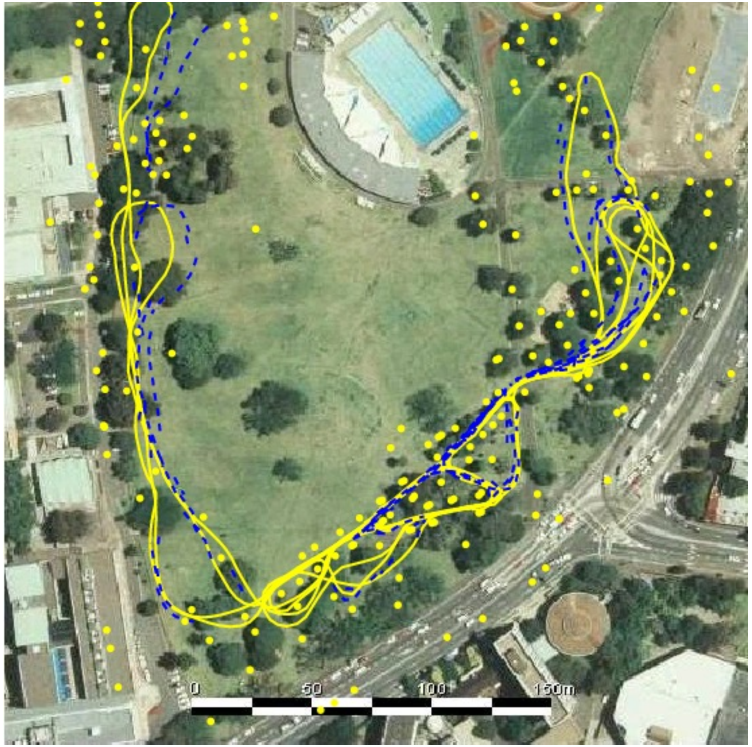
$$m^* = argmax_m P(m \mid z_1, \ldots, z_t, u_1, \ldots, u_t, x_1, \ldots, x_t)$$

$$m^* = argmax_m P(m \mid z_1, \ldots, z_t, x_1, \ldots, x_t)$$

to calculate the most likely map.

Non-parametric vs. Feature-based Maps





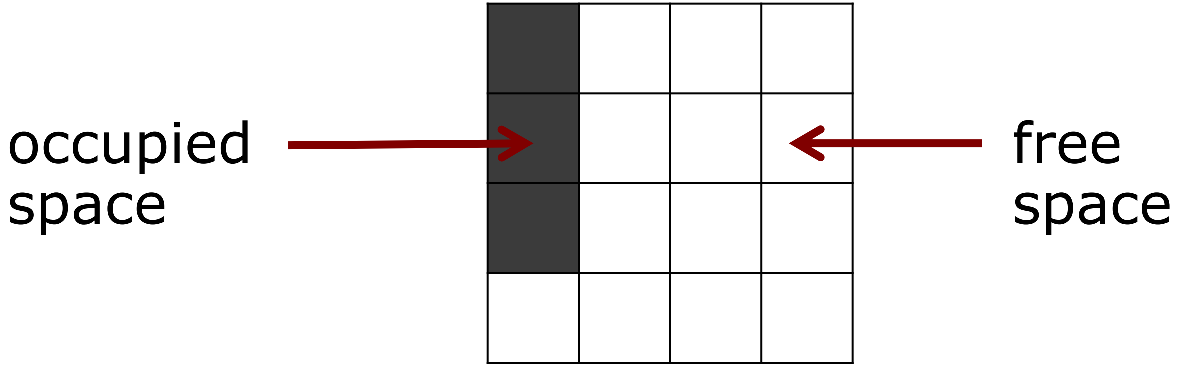
Courtesy by E. Nebot

Grid Maps

- We discretize the world into cells
- The grid structure is rigid
- Each cell is assumed to be occupied or free
- It is a non-parametric model
- It does not rely on a feature detector
- It requires substantial memory resources

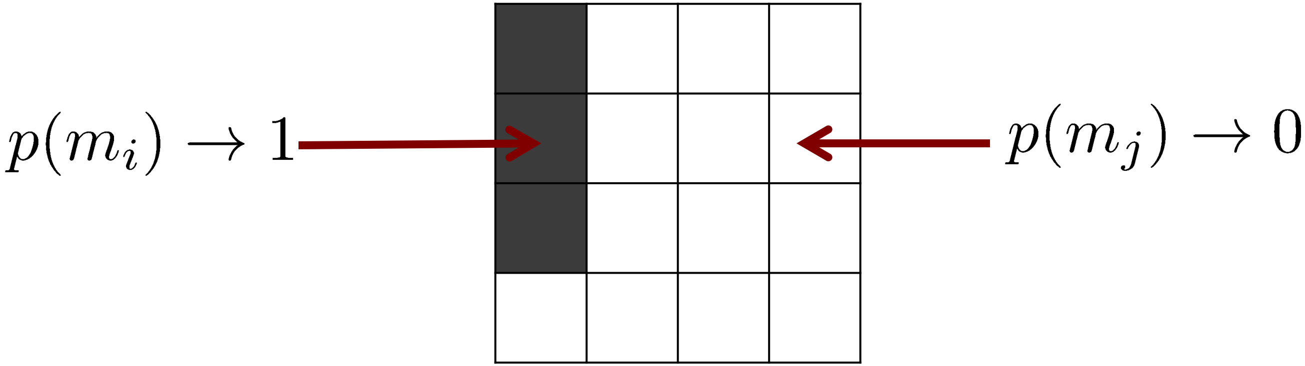
Assumption 1

The area that corresponds to each cell of the grid is either completely free or occupied



Representation

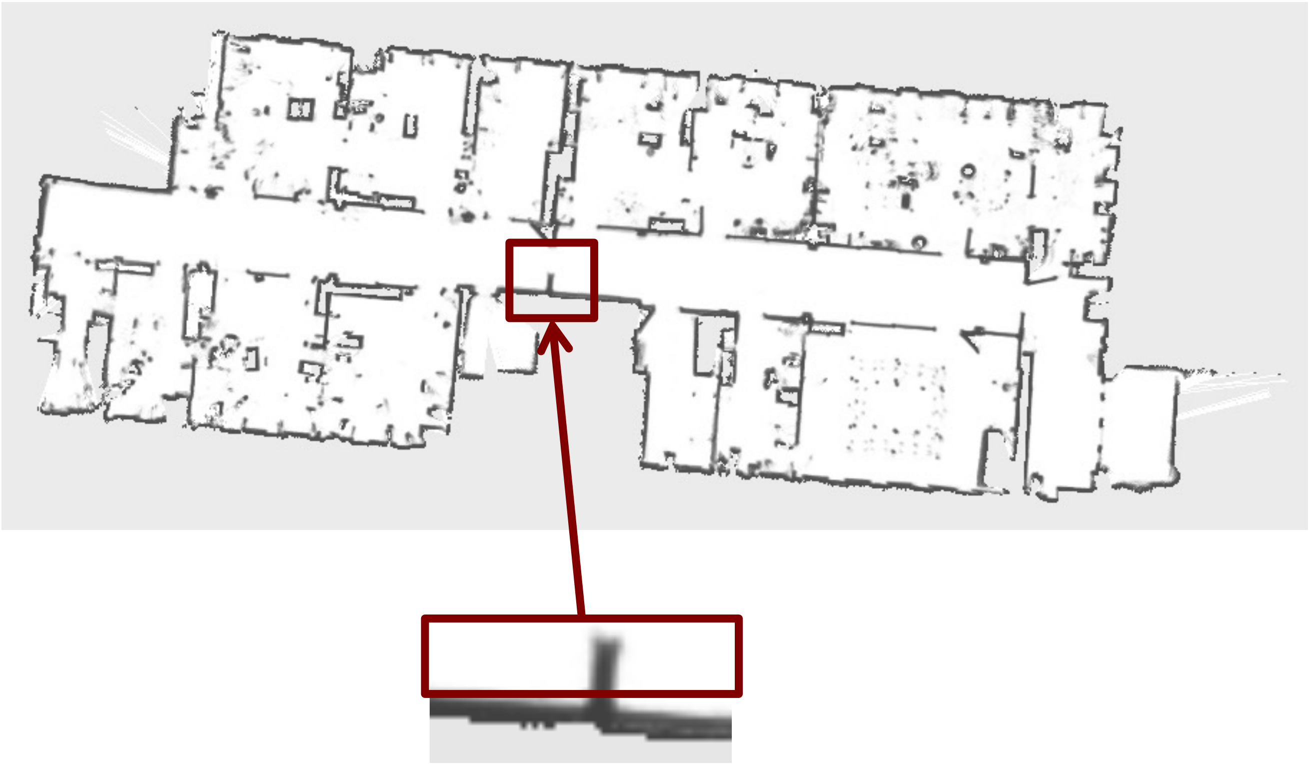
Each cell is a **binary random variable** that models the occupancy of the corresponding space in the environment.



Occupancy Probability

- Each cell is a **binary random variable** that models its occupancy
- Cell is occupied $p(m_i) = 1$
- Cell is not occupied $p(m_i) = 0$
- No information $p(m_i) = 0.5$
- The environment is assumed to be **static**

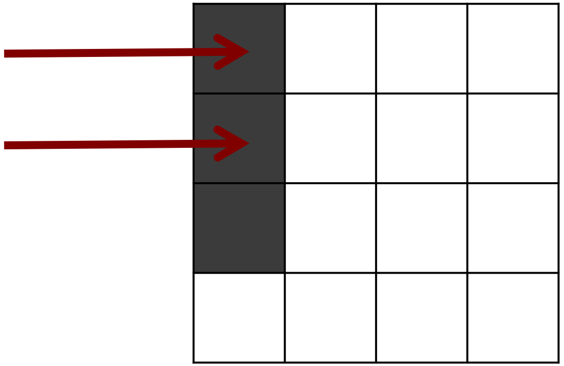
Example



Assumption 2

The cells (the random variables) are **independent** of each other

no dependency
between the cells



Representation

Given this independency assumption, the probability distribution of the map is given by the product of the probability distributions of the individual cells:

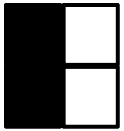
$$\underbrace{p(m)}_{\text{map}} = \prod_i \underbrace{p(m_i)}_{\text{cell}}$$

Complexity

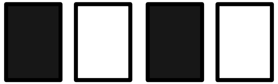
This independence assumption substantially reduces computational requirements.

$$p(m) = \prod_i p(m_i)$$

Exponential in the number of cells



n-dimensional vector



n independent cells

Linear in the number of cells

Estimating a Map From Data

Given sensor data $z_{1:t}$ and the poses $x_{1:t}$ of the sensor, estimate the map

$$p(m \mid z_{1:t}, x_{1:t}) \quad = \quad \prod_i p(m_i \mid z_{1:t}, x_{1:t})$$

binary random variable

Binary Bayes filter (for a static state)

Static State Binary Bayes Filter

$$p(m_i \mid z_{1:t}, x_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t})\, p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

Static State Binary Bayes Filter

$$p(m_i \mid z_{1:t}, x_{1:t}) \stackrel{\text{Bayes rule}}{=} \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t})\; p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

$$\stackrel{\text{Markov}}{=} \frac{p(z_t \mid m_i, x_t)\; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

Static State Binary Bayes Filter

$$p(m_i \mid z_{1:t}, x_{1:t}) \overset{\text{Bayes rule}}{=} \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t})\, p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

$$\overset{\text{Markov}}{=} \frac{p(z_t \mid m_i, x_t)\, p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})}$$

$$p(z_t \mid m_i, x_t) \overset{\text{Bayes rule}}{=} \frac{p(m_i \mid z_t, x_t)\, p(z_t \mid x_t)}{p(m_i \mid x_t)}$$

Static State Binary Bayes Filter

$$
\begin{aligned}
p(m_i \mid z_{1:t}, x_{1:t}) &\overset{\text{Bayes rule}}{=} \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t})\; p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Markov}}{=} \frac{p(z_t \mid m_i, x_t)\; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Bayes rule}}{=} \frac{p(m_i \mid z_t, x_t)\; p(z_t \mid x_t)\; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i \mid x_t)\; p(z_t \mid z_{1:t-1}, x_{1:t})}
\end{aligned}
$$

Static State Binary Bayes Filter

$$
\begin{aligned}
p(m_i \mid z_{1:t}, x_{1:t}) &\overset{\text{Bayes rule}}{=} \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t})\; p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Markov}}{=} \frac{p(z_t \mid m_i, x_t)\; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Bayes rule}}{=} \frac{p(m_i \mid z_t, x_t)\; p(z_t \mid x_t)\; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i \mid x_t)\; p(z_t \mid z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Markov}}{=} \frac{p(m_i \mid z_t, x_t)\; p(z_t \mid x_t)\; p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i)\; p(z_t \mid z_{1:t-1}, x_{1:t})}
\end{aligned}
$$

Static State Binary Bayes Filter

$$
\begin{aligned}
p(m_i \mid z_{1:t}, x_{1:t}) &\overset{\text{Bayes rule}}{=} \frac{p(z_t \mid m_i, z_{1:t-1}, x_{1:t})\, p(m_i \mid z_{1:t-1}, x_{1:t})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Markov}}{=} \frac{p(z_t \mid m_i, x_t)\, p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(z_t \mid z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Bayes rule}}{=} \frac{p(m_i \mid z_t, x_t)\, p(z_t \mid x_t)\, p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i \mid x_t)\, p(z_t \mid z_{1:t-1}, x_{1:t})} \\
&\overset{\text{Markov}}{=} \frac{p(m_i \mid z_t, x_t)\, p(z_t \mid x_t)\, p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i)\, p(z_t \mid z_{1:t-1}, x_{1:t})}
\end{aligned}
$$

Do exactly the same for the opposite:

$$
p(\neg m_i \mid z_{1:t}, x_{1:t}) \overset{\text{the same}}{=} \frac{p(\neg m_i \mid z_t, x_t)\, p(z_t \mid x_t)\, p(\neg m_i \mid z_{1:t-1}, x_{1:t-1})}{p(\neg m_i)\, p(z_t \mid z_{1:t-1}, x_{1:t})}
$$

Static State Binary Bayes Filter

- By computing the ratio of both probabilities, we obtain:

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

Static State Binary Bayes Filter

By computing the ratio of both probabilities, we obtain:

$$\frac{p(m_i \mid z_{1:t}, x_{1:t})}{p(\neg m_i \mid z_{1:t}, x_{1:t})}$$

$$= \frac{p(m_i \mid z_t, x_t)\; p(m_i \mid z_{1:t-1}, x_{1:t-1})\; p(\neg m_i)}{p(\neg m_i \mid z_t, x_t)\; p(\neg m_i \mid z_{1:t-1}, x_{1:t-1})\; p(m_i)}$$

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

Static State Binary Bayes Filter

By computing the ratio of both probabilities, we obtain:

$$\begin{aligned}
&\frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})} \\
&\quad = \frac{p(m_i \mid z_t, x_t)\, p(m_i \mid z_{1:t-1}, x_{1:t-1})\, p(\neg m_i)}{p(\neg m_i \mid z_t, x_t)\, p(\neg m_i \mid z_{1:t-1}, x_{1:t-1})\, p(m_i)} \\
&\quad = \underbrace{\frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)}}_{\text{uses } z_t} \underbrace{\frac{p(m_i \mid z_{1:t-1}, x_{1:t-1})}{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})}}_{\text{recursive term}} \underbrace{\frac{1 - p(m_i)}{p(m_i)}}_{\text{prior}}
\end{aligned}$$

Occupancy Update Rule

- Recursive rule

$$\frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})}$$

$$= \underbrace{\frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)}}_{\text{uses } z_t} \underbrace{\frac{p(m_i \mid z_{1:t-1}, x_{1:t-1})}{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})}}_{\text{recursive term}} \underbrace{\frac{1 - p(m_i)}{p(m_i)}}_{\text{prior}}$$

Occupancy Update Rule

- Recursive rule

$$\frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})} = \underbrace{\frac{p(m_i \mid z_t, x_t)}{1 - p(m_i \mid z_t, x_t)}}_{\text{uses } z_t} \underbrace{\frac{p(m_i \mid z_{1:t-1}, x_{1:t-1})}{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})}}_{\text{recursive term}} \underbrace{\frac{1 - p(m_i)}{p(m_i)}}_{\text{prior}}$$

- Often written as

$$Bel(m_t^i) = \left[1 + \frac{1 - p(m_t^i \mid z_t, x_t)}{p(m_t^i \mid z_t, x_t)} \frac{p(m_t^i)}{1 - p(m_t^i)} \frac{1 - Bel(m_{t-1}^i)}{Bel(m_{t-1}^i)}\right]^{-1}$$

Log Odds Notation

- Log odds ratio is defined as

$$l(x) = \log \frac{p(x)}{1 - p(x)}$$

- and with the ability to retrieve $p(x)$

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

Occupancy Mapping in Log Odds Form

- The product turns into a sum

$$
\begin{aligned}
& l(m_i \mid z_{1:t}, x_{1:t}) \\
& = \underbrace{l(m_i \mid z_t, x_t)}_{\text{inverse sensor model}} + \underbrace{l(m_i \mid z_{1:t-1}, x_{1:t-1})}_{\text{recursive term}} - \underbrace{l(m_i)}_{\text{prior}}
\end{aligned}
$$

- or in short

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

Occupancy Mapping Algorithm

$$\textbf{occupancy\_grid\_mapping}(\{l_{t-1,i}\}, x_t, z_t)\textbf{:}$$

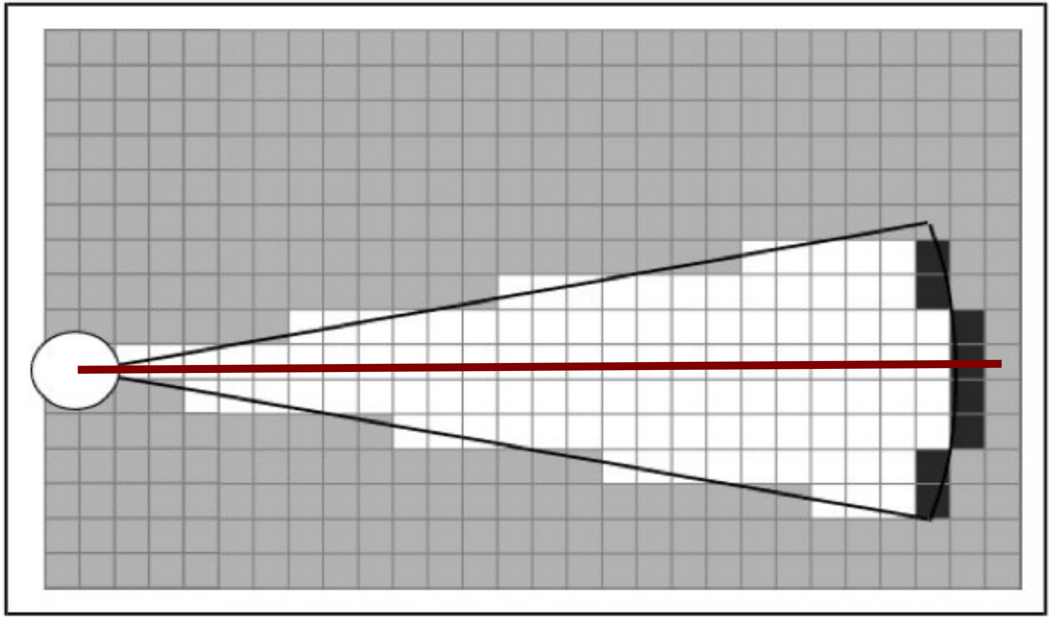
1: *for all cells* m_i *do*
2: *if* m_i *in perceptual field of* z_t *then*
3: $l_{t,i} = l_{t-1,i} + \text{inv\_sensor\_model}(m_i, x_t, z_t) - l_0$
4: *else*
5: $l_{t,i} = l_{t-1,i}$
6: *endif*
7: *endfor*
8: *return* $\{l_{t,i}\}$

highly efficient, only requires to compute sums

Occupancy Grid Mapping

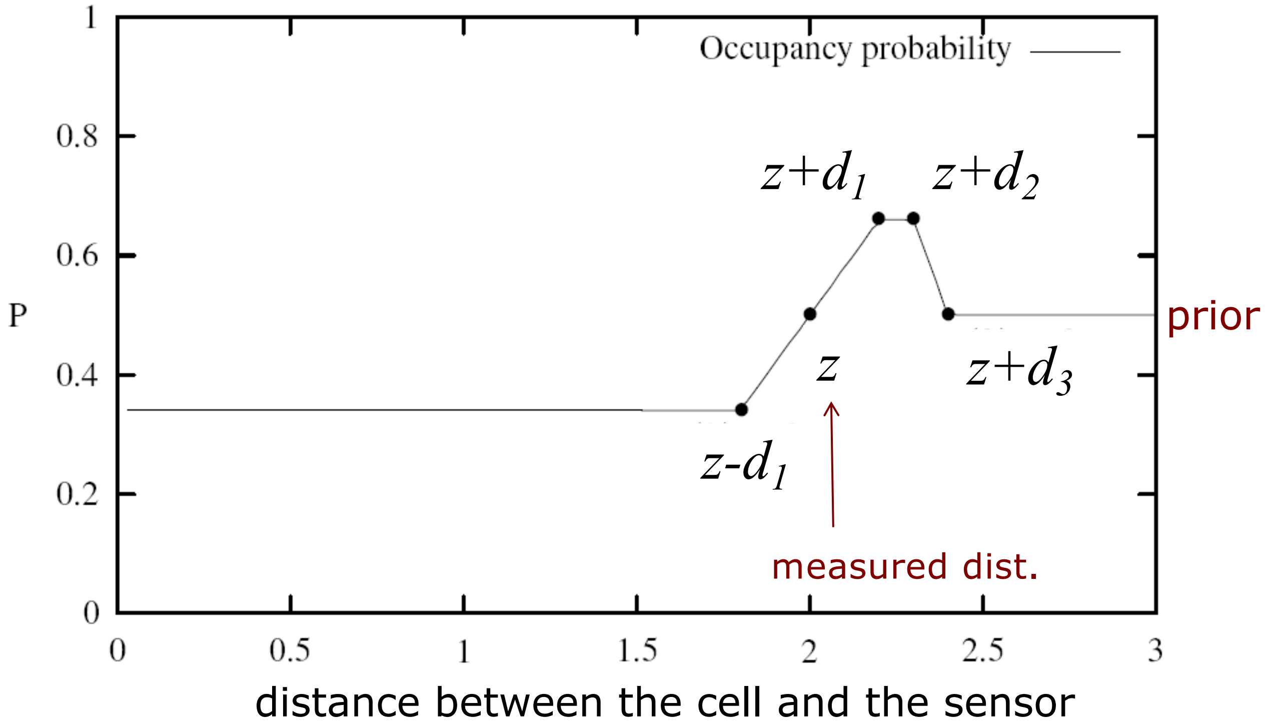
- Developed in the mid 80's by Moravec and Elfes
- Originally developed for noisy sonars
- Also called "mapping with known poses"

Inverse Sensor Model $p(m_i \mid z_t, x_t)$ for Sonars Range Sensors

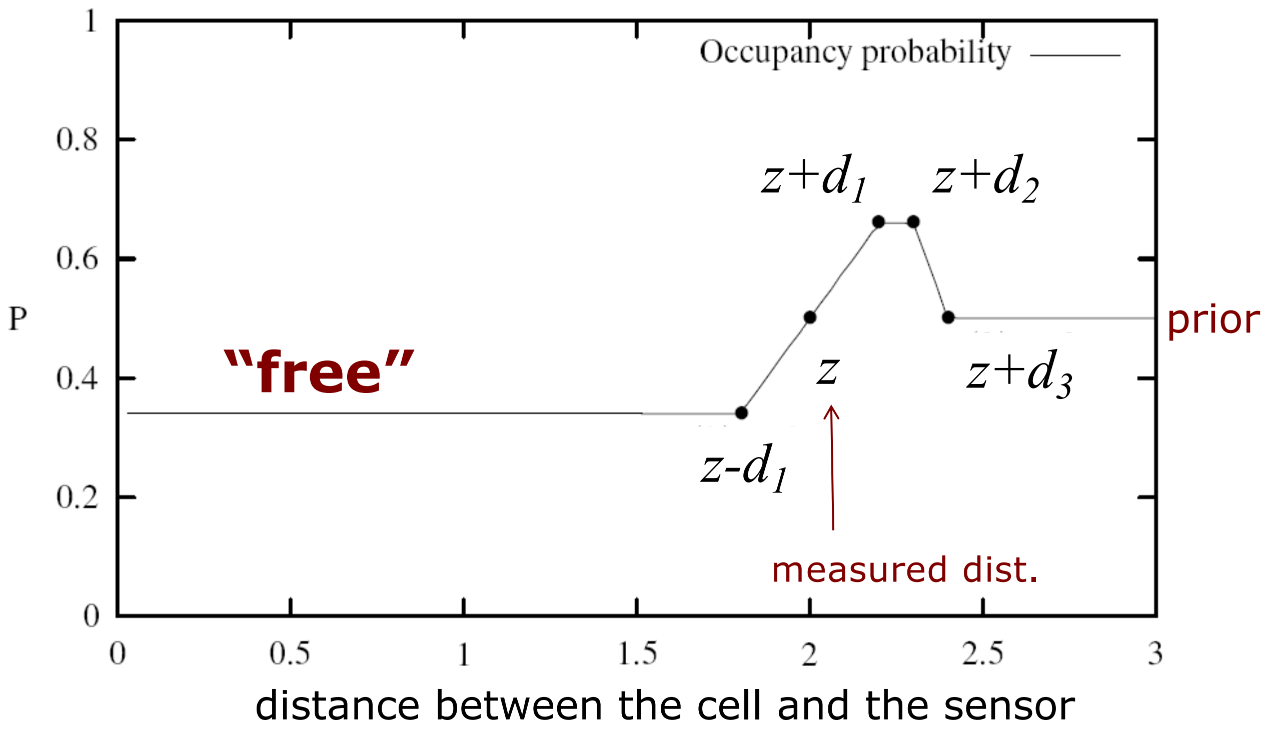


In the following, consider the cells along the optical axis (red line)

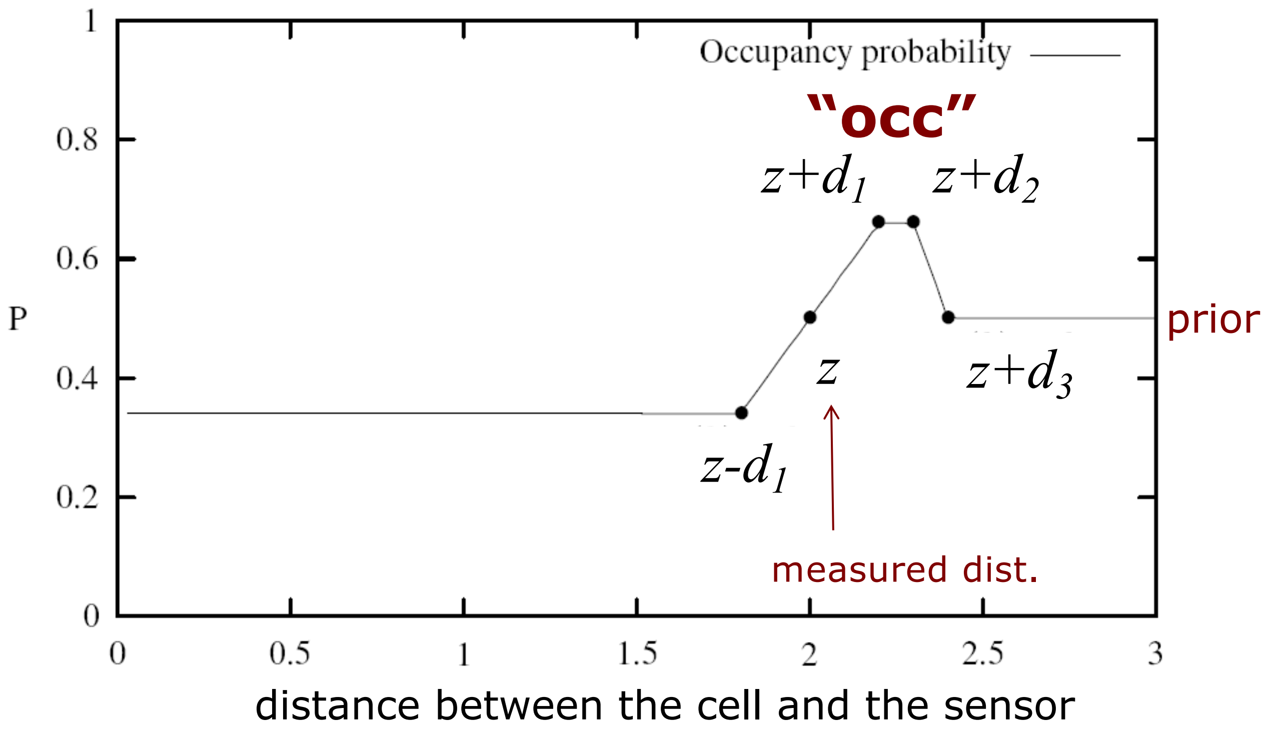
Occupancy Value Depending on the Measured Distance



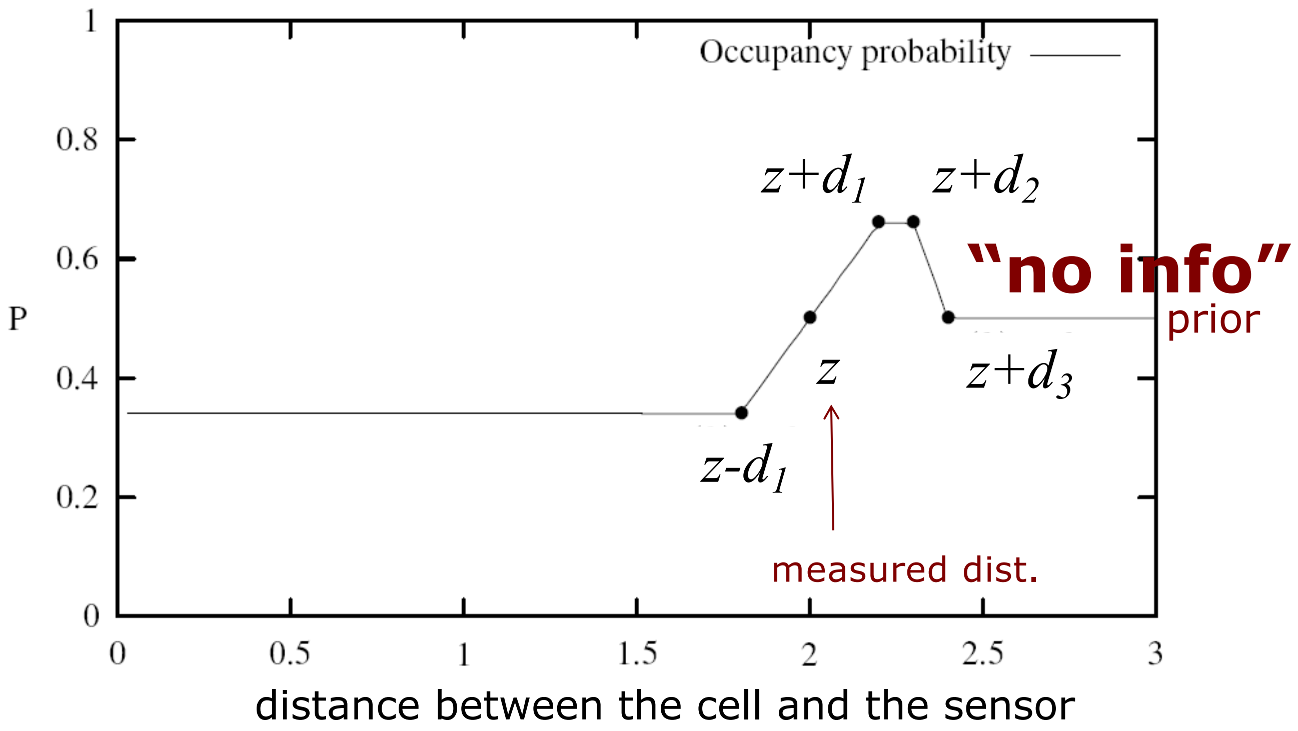
Occupancy Value Depending on the Measured Distance



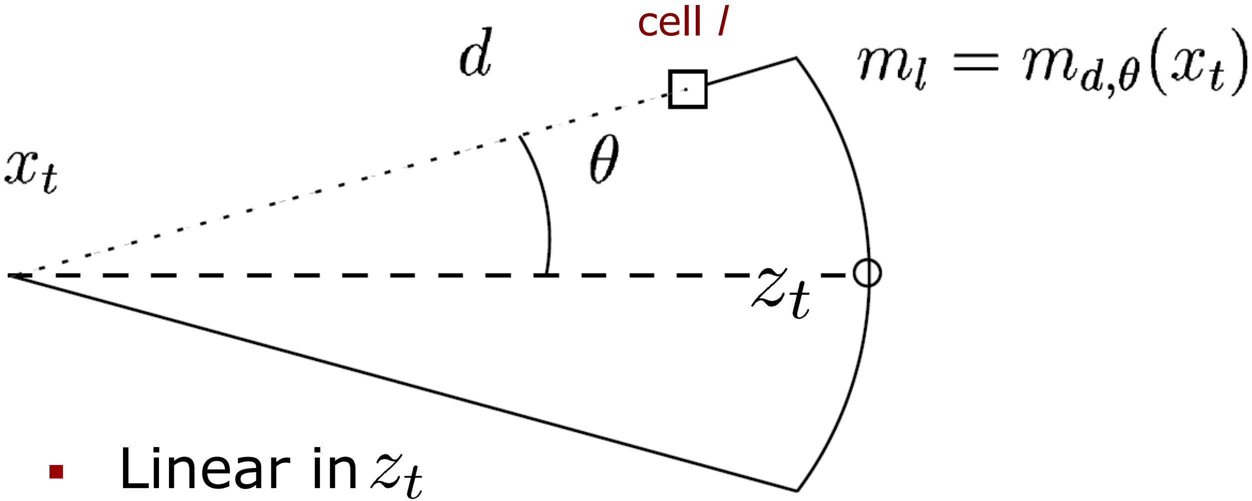
Occupancy Value Depending on the Measured Distance



Occupancy Value Depending on the Measured Distance

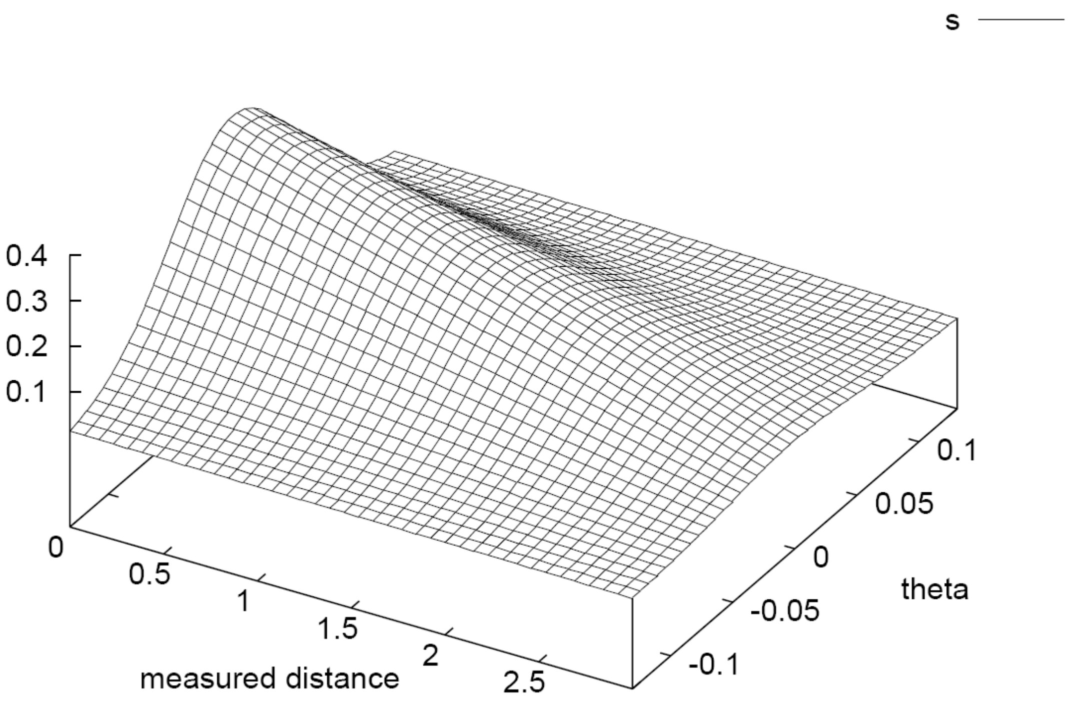


Update depends on the Measured Distance and Deviation from the Optical Axis

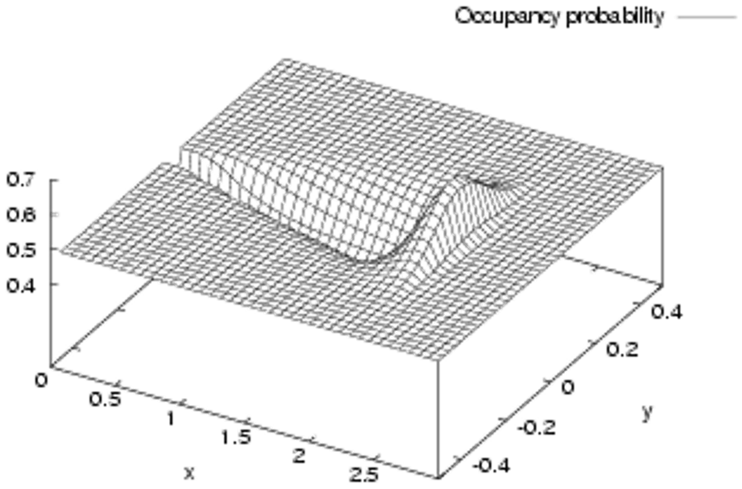


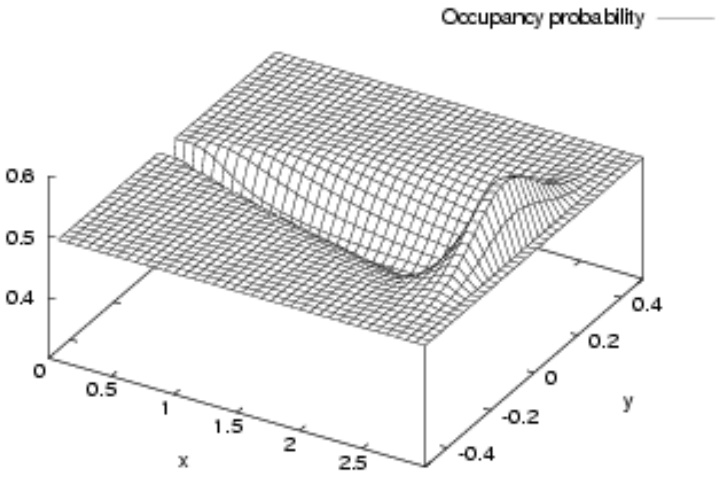
- Linear in z_t
- Gaussian in θ

Intensity of the Update

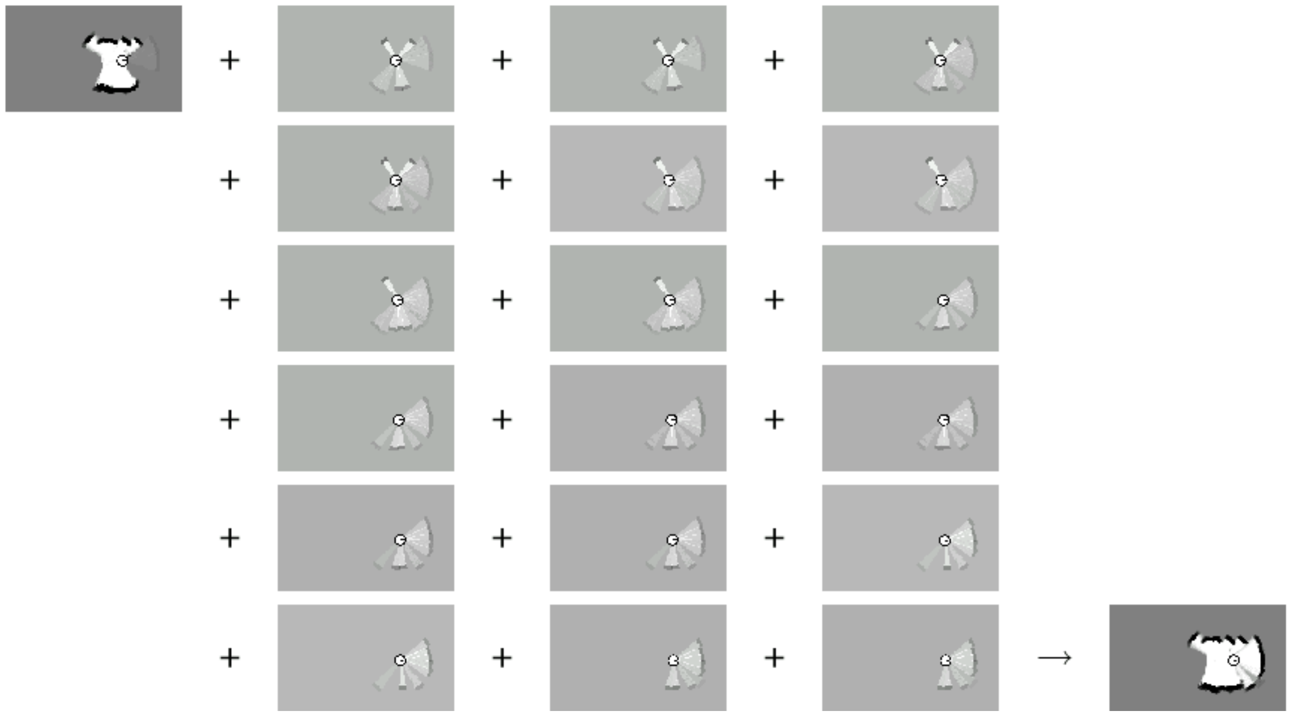


Resulting Model

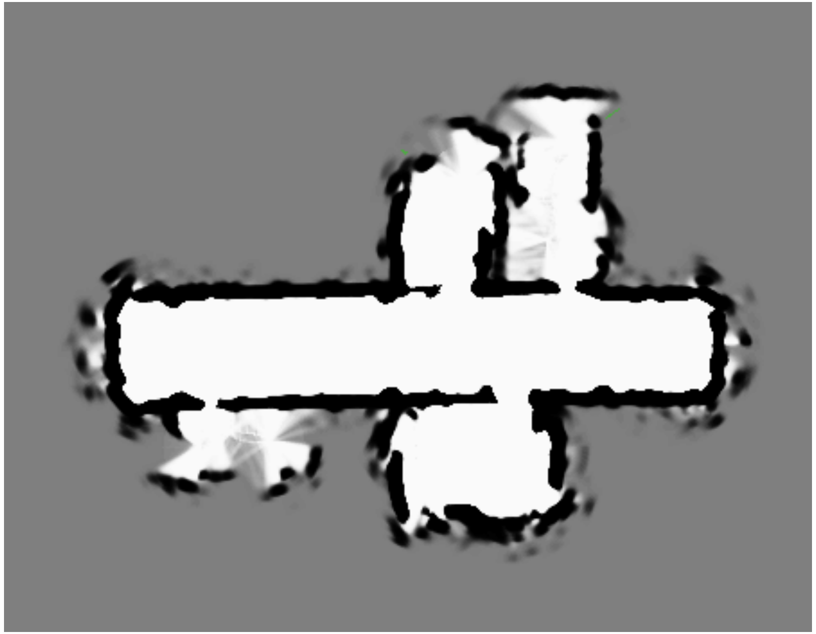




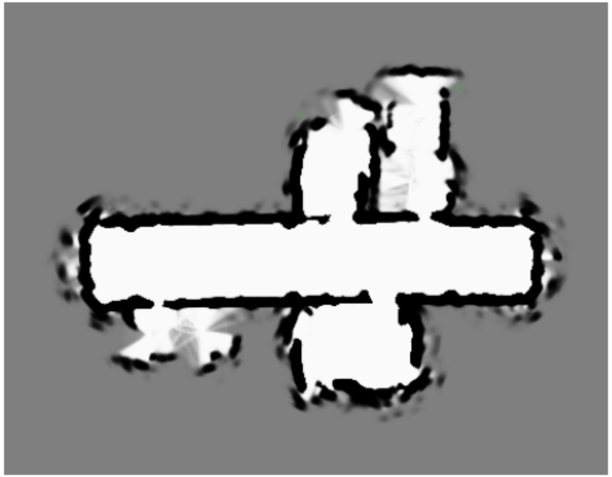
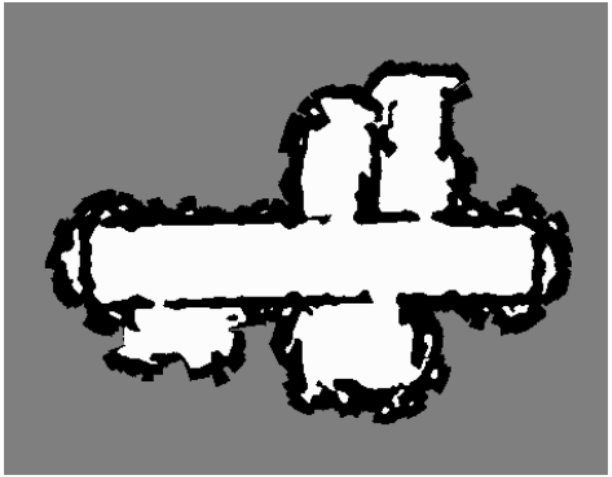
Example: Incremental Updating of Occupancy Grids



Resulting Map Obtained with Ultrasound Sensors

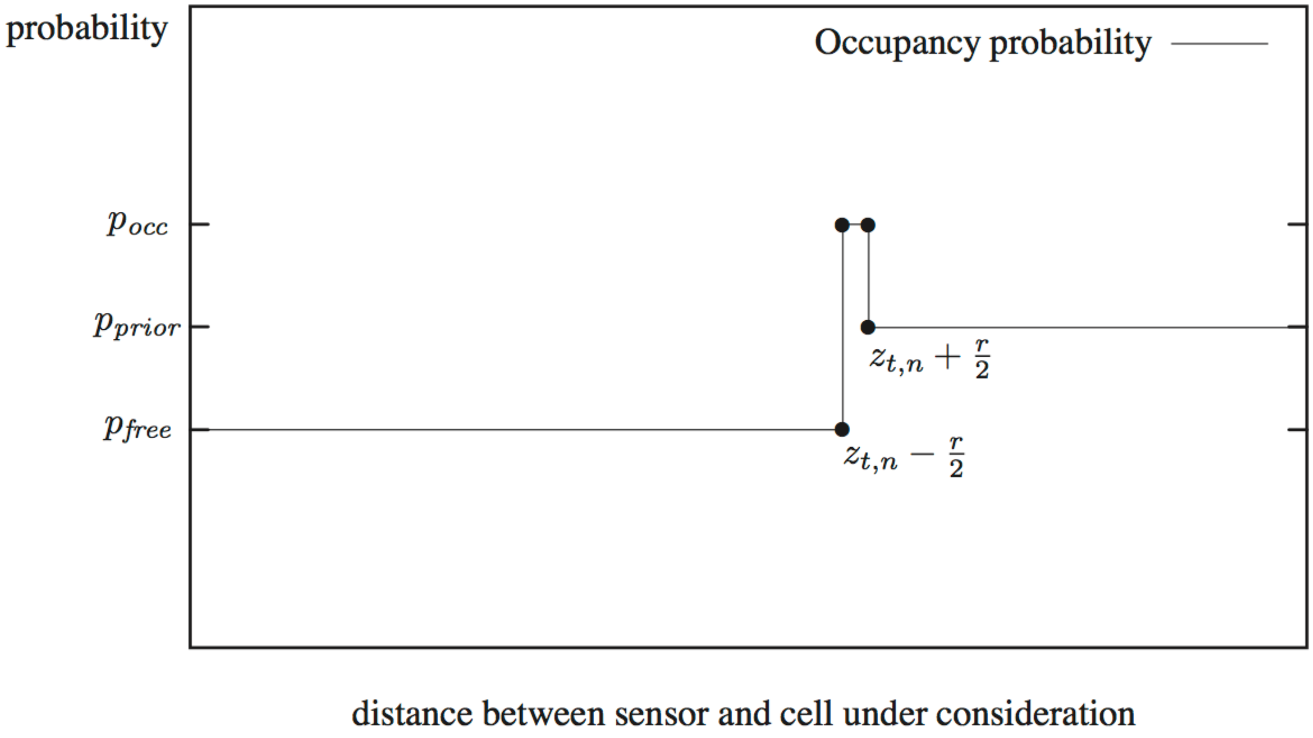


Resulting Occupancy and Maximum Likelihood Map

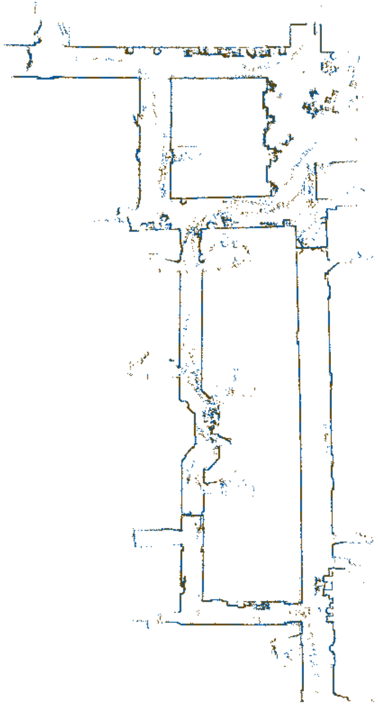
 

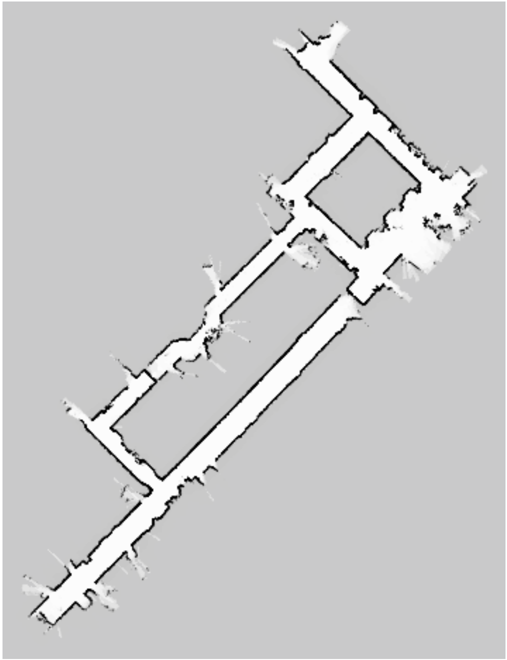
The maximum likelihood map is obtained by rounding the probability for each cell to 0 or 1.

Inverse Sensor Model $p(m_i \mid z_t, x_t)$ for Laser Range Finders

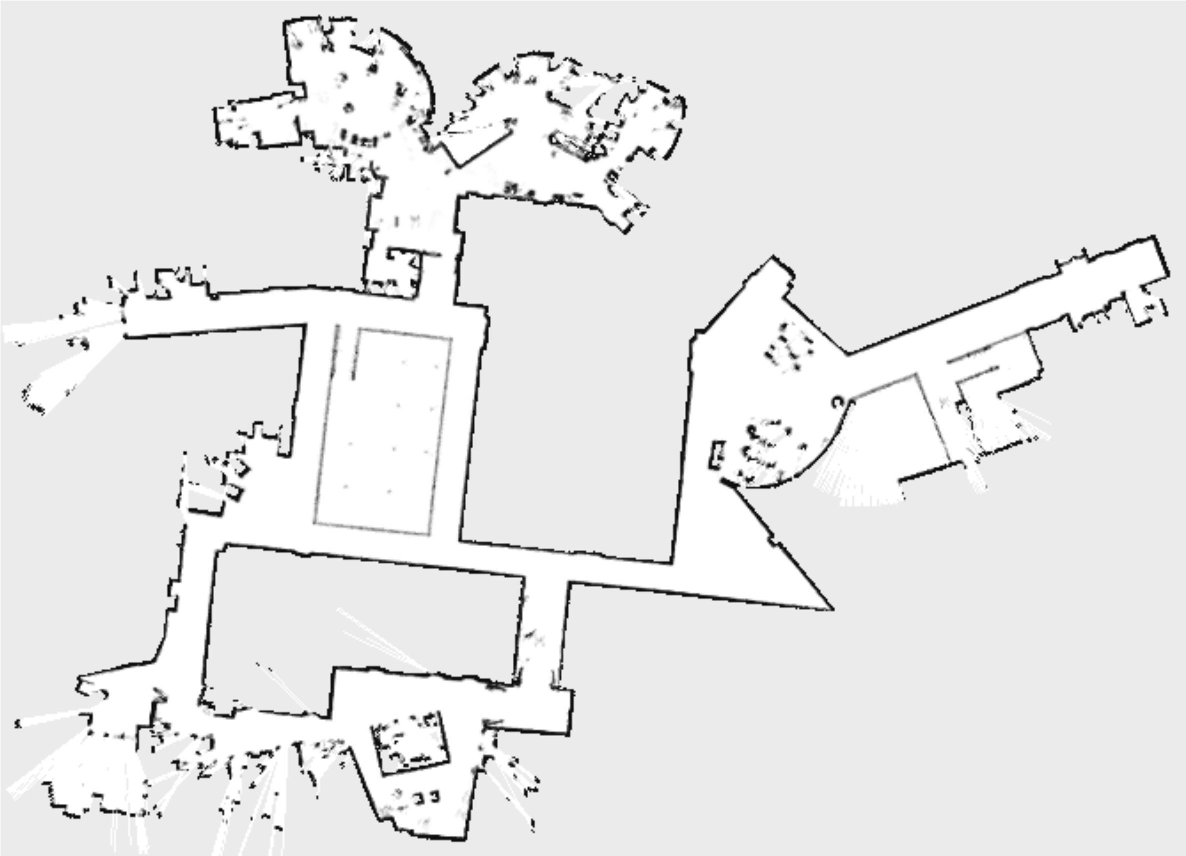


Occupancy Grids From Laser Scans

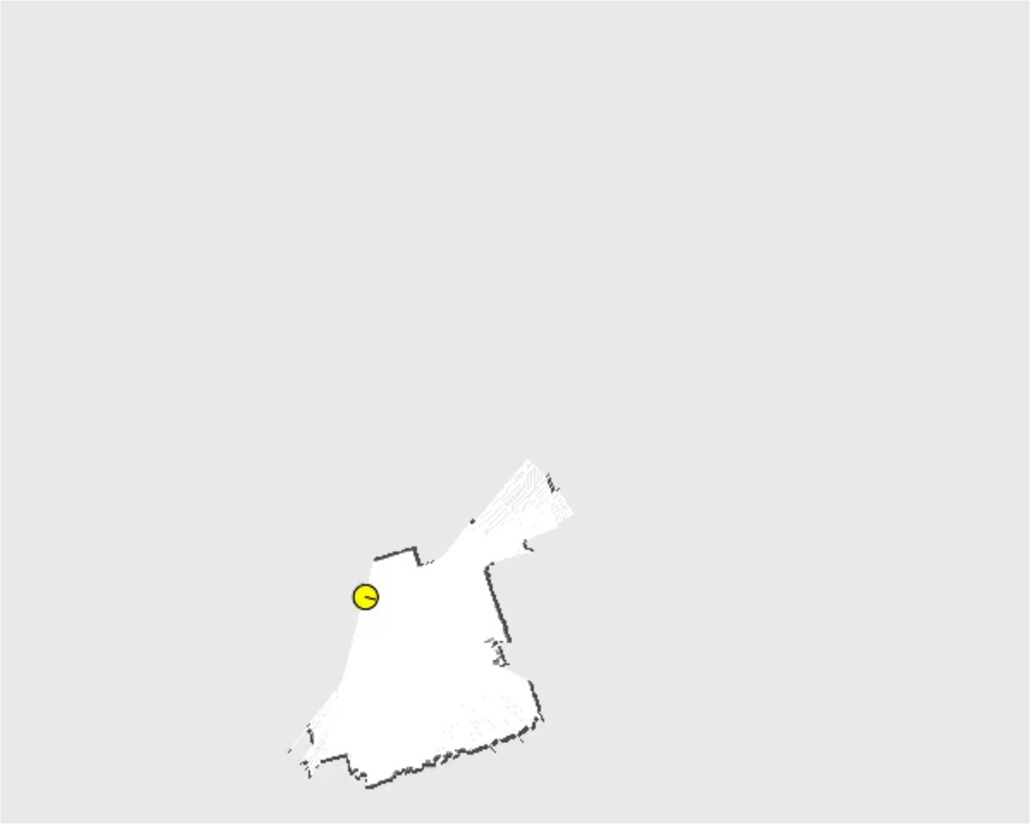




Example: MIT CSAIL 3rd Floor



Uni Freiburg Building 106



Alternative Approach: The Counting Model

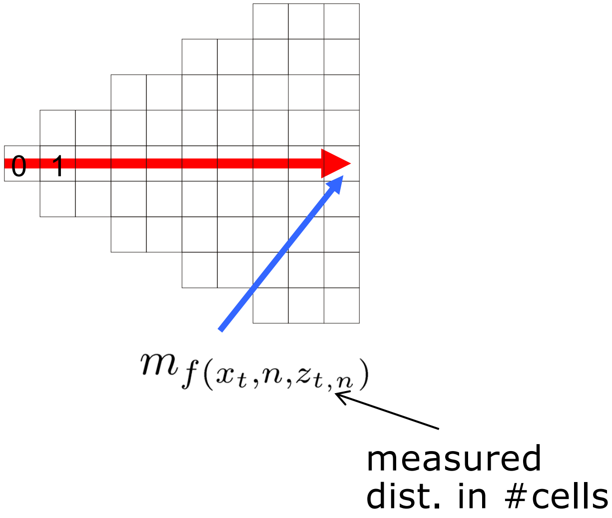
- For every cell count
  - hits(x,y): number of cases where a beam ended at <x,y>
  - misses(x,y): number of cases where a beam passed through <x,y>

$$Bel(m^{[xy]}) = \frac{\text{hits}(x,y)}{\text{hits}(x,y)+\text{misses}(x,y)}$$

- Value of interest: P(reflects(x,y))

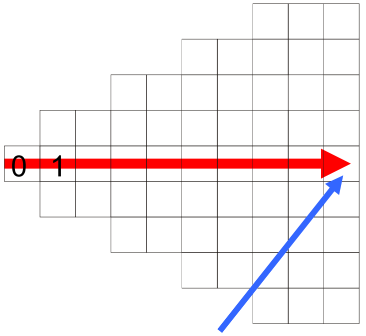
The Measurement Model

- Pose at time t: x_t
- Beam n of scan at time t: $z_{t,n}$
- Maximum range reading: $\zeta_{t,n} = 1$
- Beam reflected by an object: $\zeta_{t,n} = 0$



The Measurement Model

- Pose at time t: x_t
- Beam n of scan at time t: $z_{t,n}$
- Maximum range reading: $\zeta_{t,n} = 1$
- Beam reflected by an object: $\zeta_{t,n} = 0$

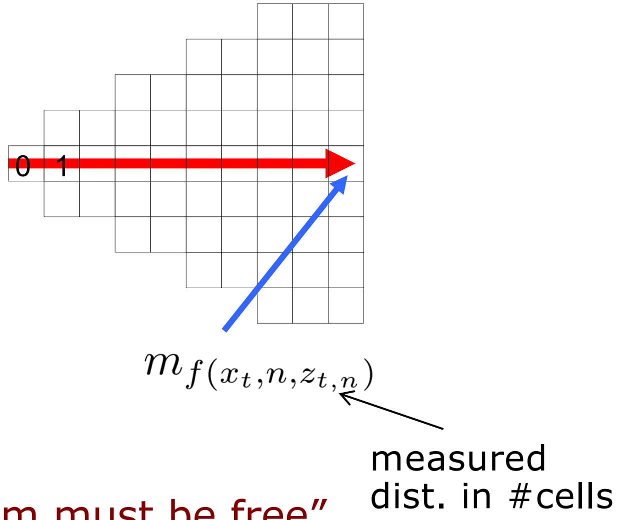


max range: "first $z_{t,n} - 1$ cells covered by the beam must be free"

$$p(z_{t,n}|x_t, m) = \begin{cases} \prod_{k=0}^{z_{t,n}-1} \left(1 - m_{f(x_t,n,k)}\right) & \text{if } \zeta_{t,n} = 1 \\ \end{cases}$$

The Measurement Model

- Pose at time t: x_t
- Beam n of scan at time t: $z_{t,n}$
- Maximum range reading: $\zeta_{t,n} = 1$
- Beam reflected by an object: $\zeta_{t,n} = 0$



max range: "first $z_{t,n} - 1$ cells covered by the beam must be free"

$$p(z_{t,n}|x_t, m) = \begin{cases} \prod_{k=0}^{z_{t,n}-1} \left(1 - m_{f(x_t,n,k)}\right) & \text{if } \zeta_{t,n} = 1 \\ m_{f(x_t,n,z_{t,n})} \prod_{k=0}^{z_{t,n}-1} \left(1 - m_{f(x_t,n,k)}\right) & \zeta_{t,n} = 0 \end{cases}$$

otherwise: "last cell reflected beam, all others free"

Computing the Most Likely Map

- Compute values for m that maximize

$$m^\star = \operatorname{argmax}_m P(m \mid z_1, \cdots, z_t, x_1, \cdots, x_t)$$

- Assuming a uniform prior probability for P(m), this is equivalent to maximizing:

$$\begin{aligned} m^\star &= \operatorname{argmax}_m P(z_1, \cdots, z_t \mid m, x_1, \cdots, x_t) \\ &= \operatorname{argmax}_m \prod_{t=1}^{T} P(z_t \mid m, x_t) \quad \text{since } z_t \text{ independent given } x_t \text{ and only depend on } x_t \\ &= \operatorname{argmax}_m \sum_{t=1}^{T} \ln P(z_t \mid m, x_t) \end{aligned}$$

Computing the Most Likely Map

$$
\begin{aligned}
m^\star \quad = \quad & \operatorname{argmax}_m \overset{\text{cells}}{\sum_{j=1}^{J}} \sum_{t=1}^{T} \overset{\text{beams}}{\sum_{n=1}^{N}} \Big(I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j \\
& + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \Big)
\end{aligned}
$$

Computing the Most Likely Map

"beam n ends in cell j"

$$m^\star = \operatorname{argmax}_m \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \Big(I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j$$

$$+ \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \Big)$$

Computing the Most Likely Map

"beam n ends in cell j"

$$m^\star = \operatorname{argmax}_m \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \Big(I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j$$

$$+ \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \Big)$$

"beam n traversed cell j"

Computing the Most Likely Map

"beam n ends in cell j"

$$m^\star = \operatorname{argmax}_m \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \Big(I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \Big)$$

"beam n traversed cell j"

Define

$$\alpha_j = \sum_{t=1}^{T} \sum_{n=1}^{N} I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n})$$

$$\beta_j = \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j)$$

Meaning of α_j and β_j

$$\alpha_j = \sum_{t=1}^{T} \sum_{n=1}^{N} I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n})$$

Corresponds to the number of times a beam that is not a maximum range beam ended in cell j ($hits(j)$)

$$\beta_j = \sum_{t=1}^{T} \sum_{n=1}^{N} \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j)$$

Corresponds to the number of times a beam traversed cell j without ending in it ($misses(j)$)

Computing the Most Likely Map

Accordingly, we get

$$m^\star = \operatorname{argmax}_m \sum_{j=1}^{J} \left(\alpha_j \ln m_j + \beta_j \ln(1 - m_j) \right)$$

As the m_j's are independent of each other we can maximize this sum by maximizing it for every j

If we set $\frac{\partial}{\partial m_j} = \frac{\alpha_j}{m_j} - \frac{\beta_j}{1-m_j} = 0$ we obtain $m_j = \frac{\alpha_j}{\alpha_j+\beta_j}$

Computing the most likely map reduces to counting how often a cell has reflected a measurement and how often the cell was traversed by a beam.

Difference between Occupancy Grid Maps and Counting

- The counting model determines how often a cell reflects a beam.
- The occupancy model represents whether or not a cell is occupied by an object.
- Although a cell might be occupied by an object, the reflection probability of this object might be very small.

Example Occupancy Map

Example Reflection Map

glass panes

Example

- Out of n beams only 60% are reflected from a cell and 40% intercept it without ending in it.
- Accordingly, the reflection probability will be 0.6.
- Suppose *p(occ | z)* = 0.55 when a beam ends in a cell and *p(occ | z)* = 0.45 when a beam traverses a cell without ending in it.
- Accordingly, after n measurements we will have

$$\left(\frac{0.55}{0.45}\right)^{n*0.6} * \left(\frac{0.45}{0.55}\right)^{n*0.4} = \left(\frac{11}{9}\right)^{n*0.6} * \left(\frac{11}{9}\right)^{-n*0.4} = \left(\frac{11}{9}\right)^{n*0.2}$$

- The reflection map yields a value of 0.6, while the occupancy grid value converges to 1 as n increases.

Summary (1)

- Grid maps are a popular model for representing the environment of a (mobile) robot
- Occupancy grid maps discretize the space into independent cells
- Each cell is a binary random variable representing the occupancy in the environment
- Binary Bayes Filters are an effective way to estimate the occupancy of the individual cells
- This leads to an efficient algorithm for mapping with known poses
- The log odds model is fast to compute

Summary (2)

- Reflection probability maps estimate for each cell the probability that it reflects a sensor beam
- Counting the number of times how often a measurement intercepts or ends in a cell yields the maximum likelihood mapping model.
- In contrast to previously described sensor and inverse sensor models, the counting approach is consistent with the reflection model