Introduction to Mobile Robotics

Grid Maps and Mapping with Known Poses

Lukas Luft

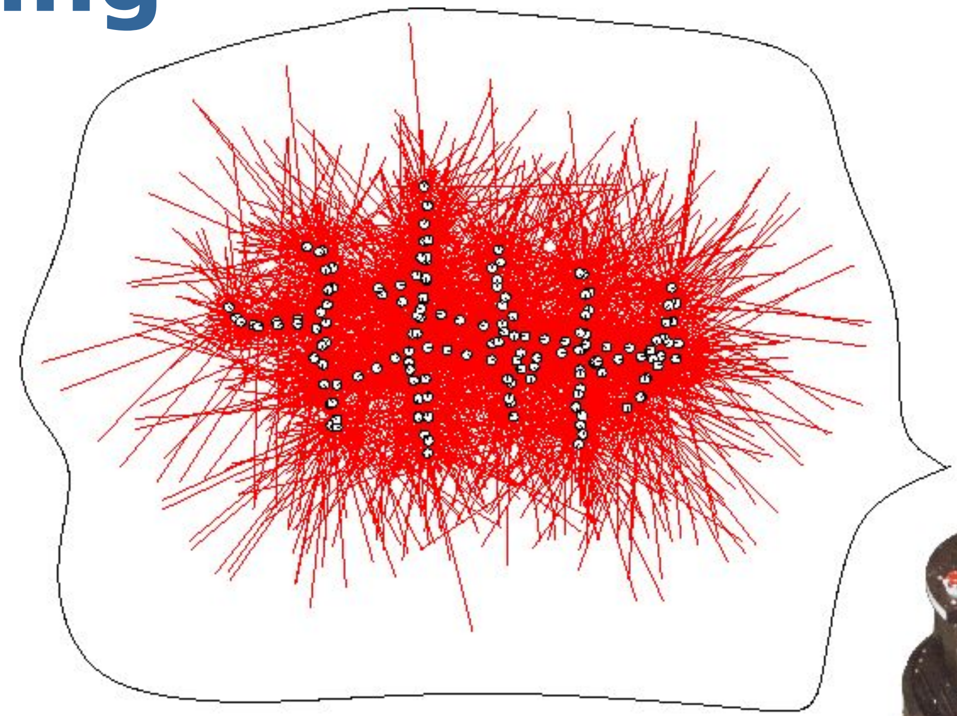
UNI FREIBURG

AiS Autonomous Intelligent Systems

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.

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

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

The General Problem of Mapping

- Formally, mapping involves, given the sensor data

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

- to calculate the most likely map

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

- **How to calculate map given robot's poses?**

The General Problem of Mapping with Known Poses

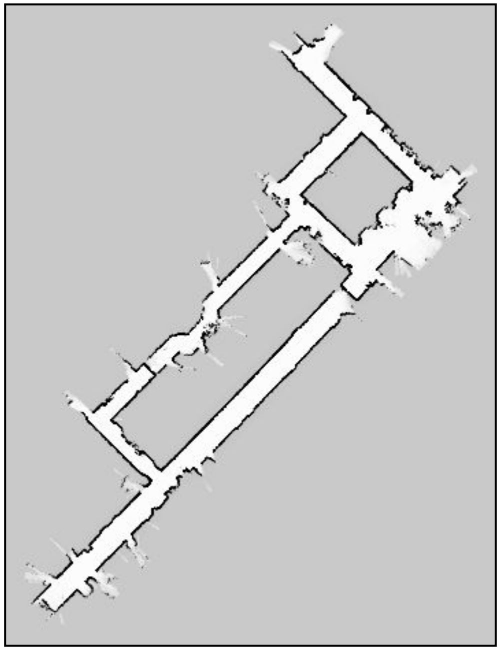
- Formally, mapping with known poses involves, given the measurements and the poses

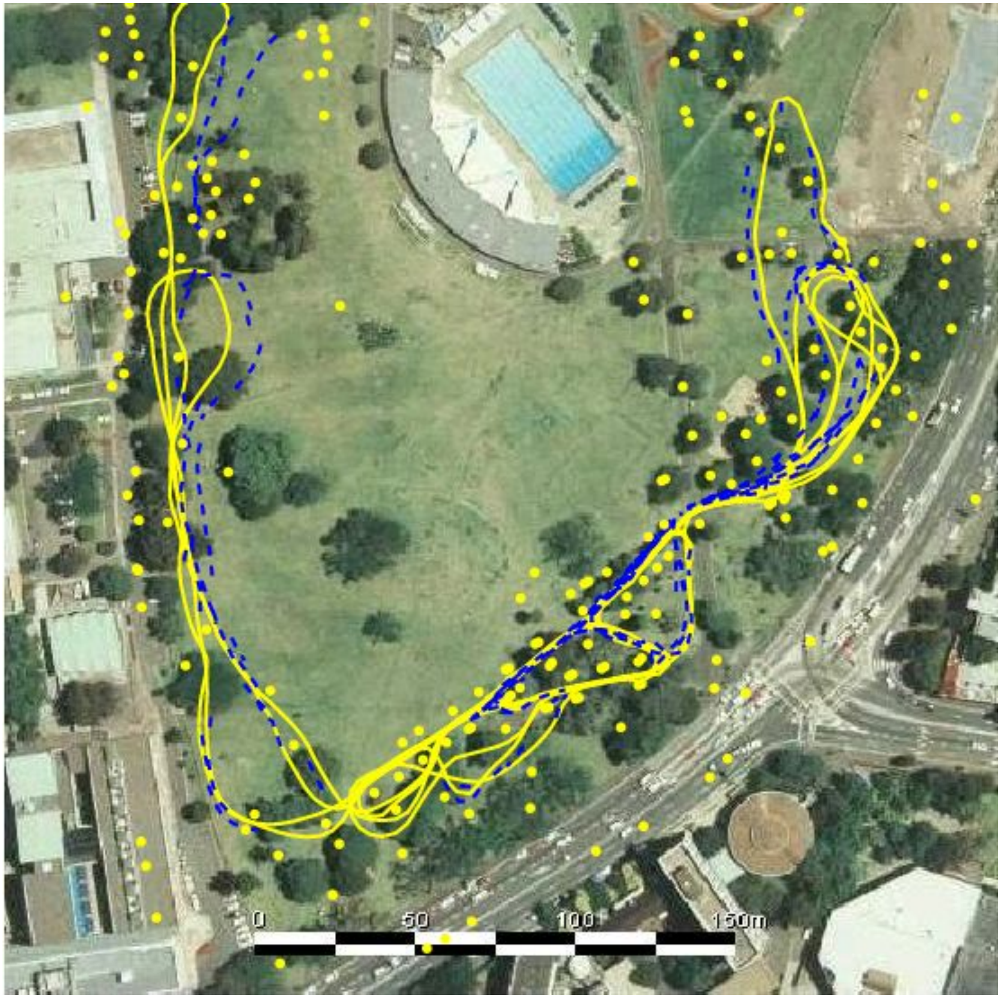
$$d = \{x_1, z_1, x_2, z_2, \ldots, x_t, z_t\}$$

- to calculate the most likely map

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

Features vs. Volumetric Maps



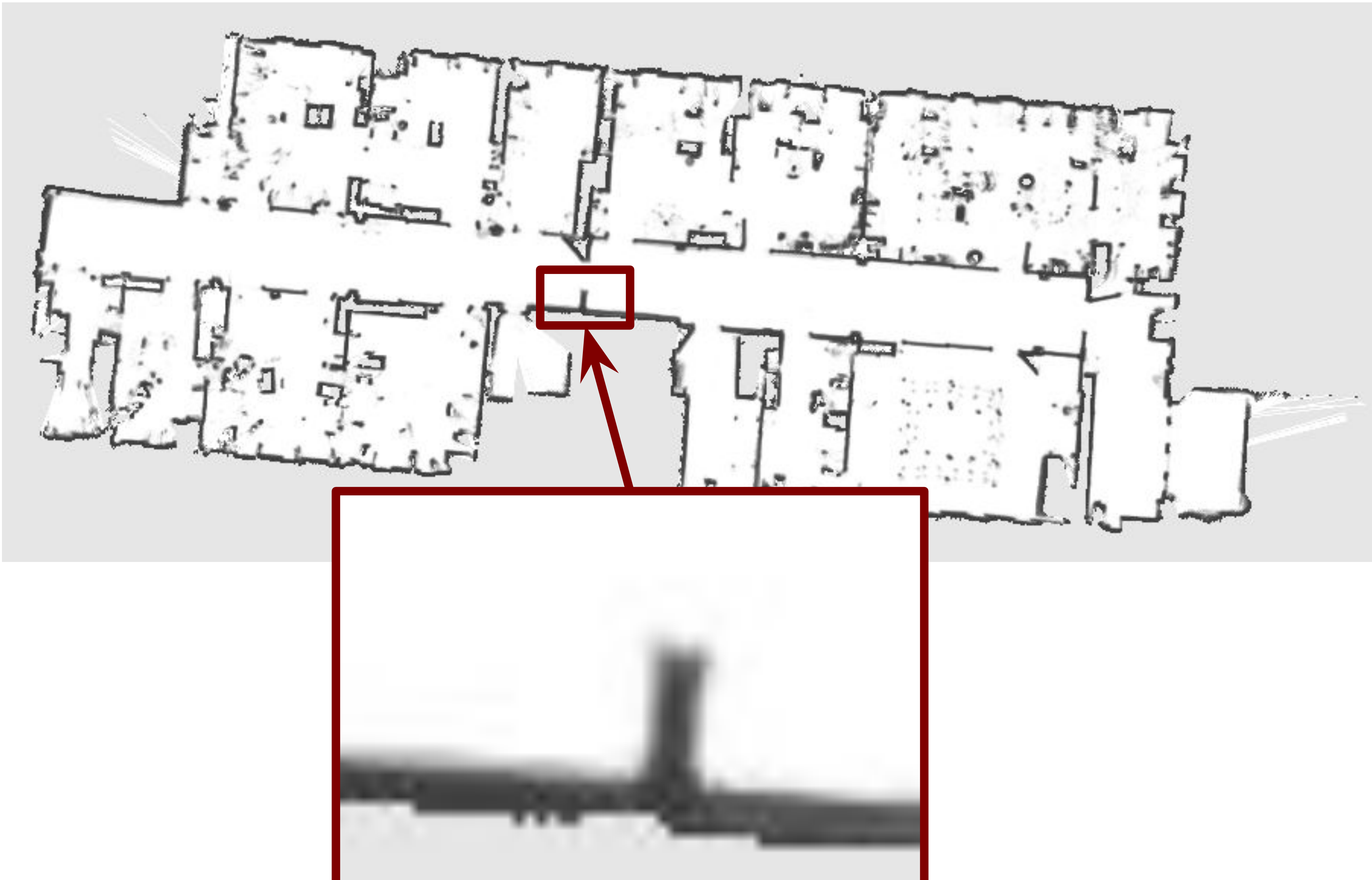


Courtesy by E. Nebot

Grid Maps

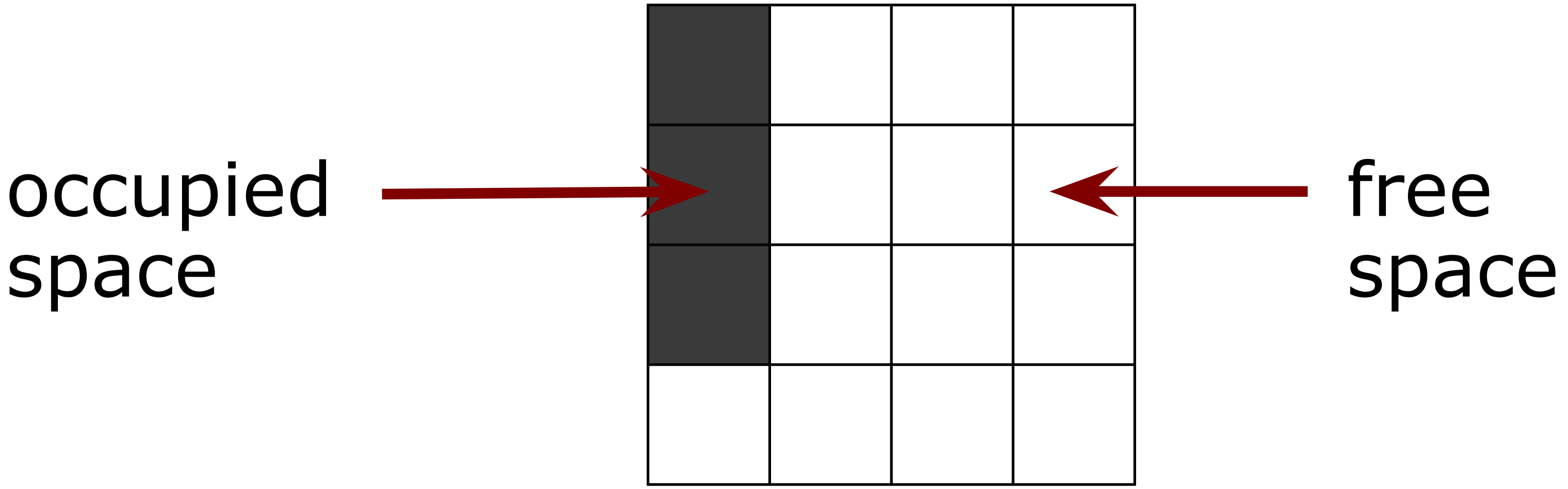
- We discretize the world into cells
- The grid structure is rigid
- Each cell is assumed to be occupied or free
- It is a non-parametric model
- It requires substantial memory resources
- It does not rely on a feature detector

Example



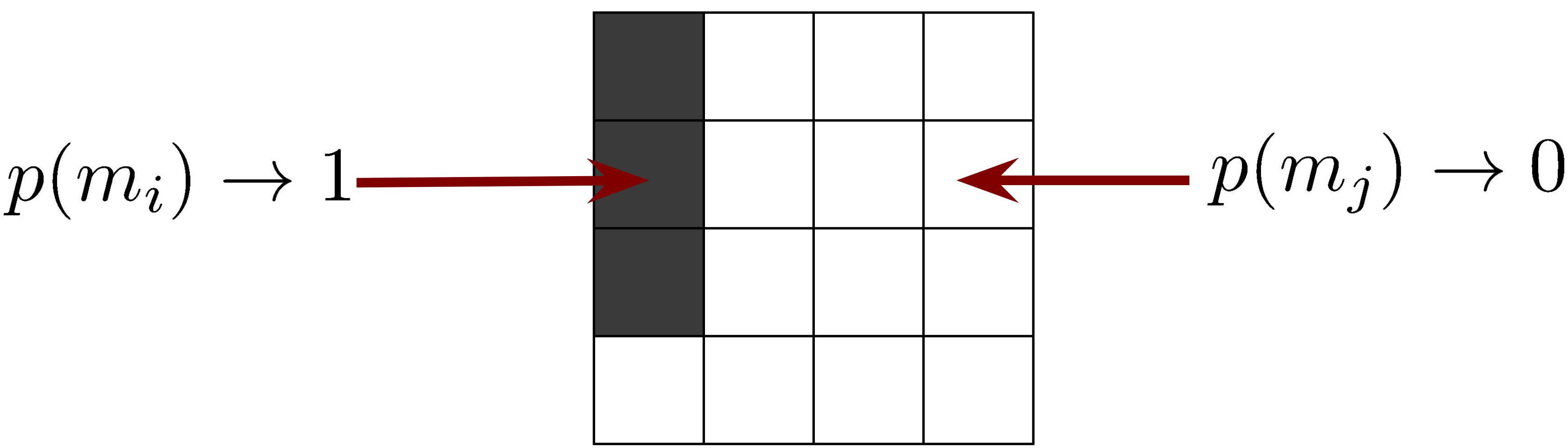
Assumption 1

- The area that corresponds to a cell is either completely free or occupied



Representation

- Each cell is a **binary random variable** that models the occupancy



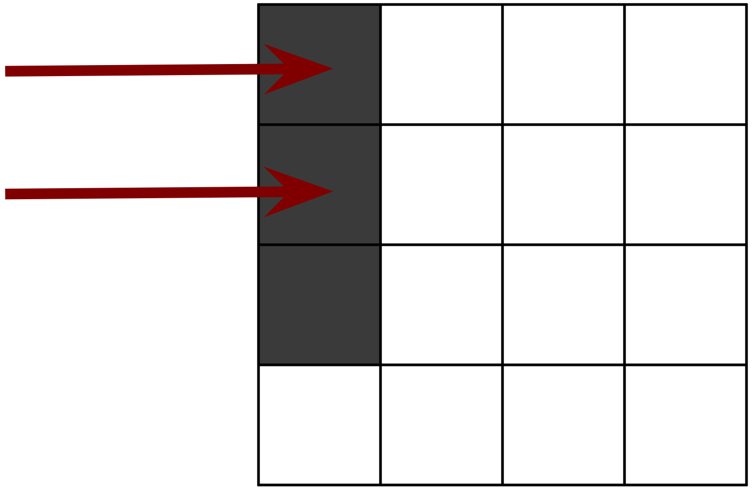
Occupancy Probability

- Each cell is a **binary random variable** that models the occupancy
- We know that the cell is occ $p(m_i) = 1$
- … is not occupie $p(m_i) = 0$
- No informatio $p(m_i) = 0.5$
- The environment is assumed to be **static**

Assumption 2

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

no dependency
between the cells



Representation

- The probability distribution of the map is given by the product of the probability distributions of the individual cells

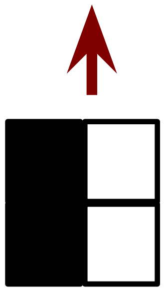
$$p(m) = \prod_i p(m_i)$$

map   cell

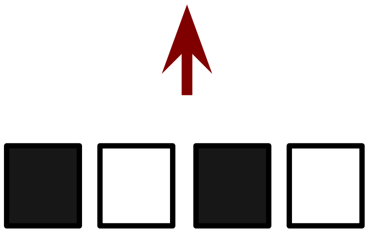
Representation

- The probability distribution of the map is given by the product of the probability distributions of the individual cells

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



four-dimensional vector



four independent cells

Estimating a Map From Data

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

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

binary random variable

➡ Binary Bayes filter (for a static state)

Estimating a Map From Data

Note that

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

is an even stronger assumption than

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

because measurements induce correlations between cells (especially for sonar). We have to use these (obviously false) assumptions for computational feasibility.

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

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

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

- The first assumption is actually only justified when conditioning on the full map not just m_i. People use it here, nevertheless.

Static State Binary Bayes Filter

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

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

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

- Defining a forward sensor model conditioned on only one cell is impossible, therefore, we use Bayes rule again to apply an (heuristic) inverse sensor model.

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

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

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

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

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

Static State Binary Bayes Filter

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

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

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

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

Do exactly the same for the opposite event:

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

Static State Binary Bayes Filter

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

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

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:

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

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

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

Occupancy Update Rule

- Recursive rule

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

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

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

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

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

Occupancy Mapping in Log Odds Form

- The product turns into a sum

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

- 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

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

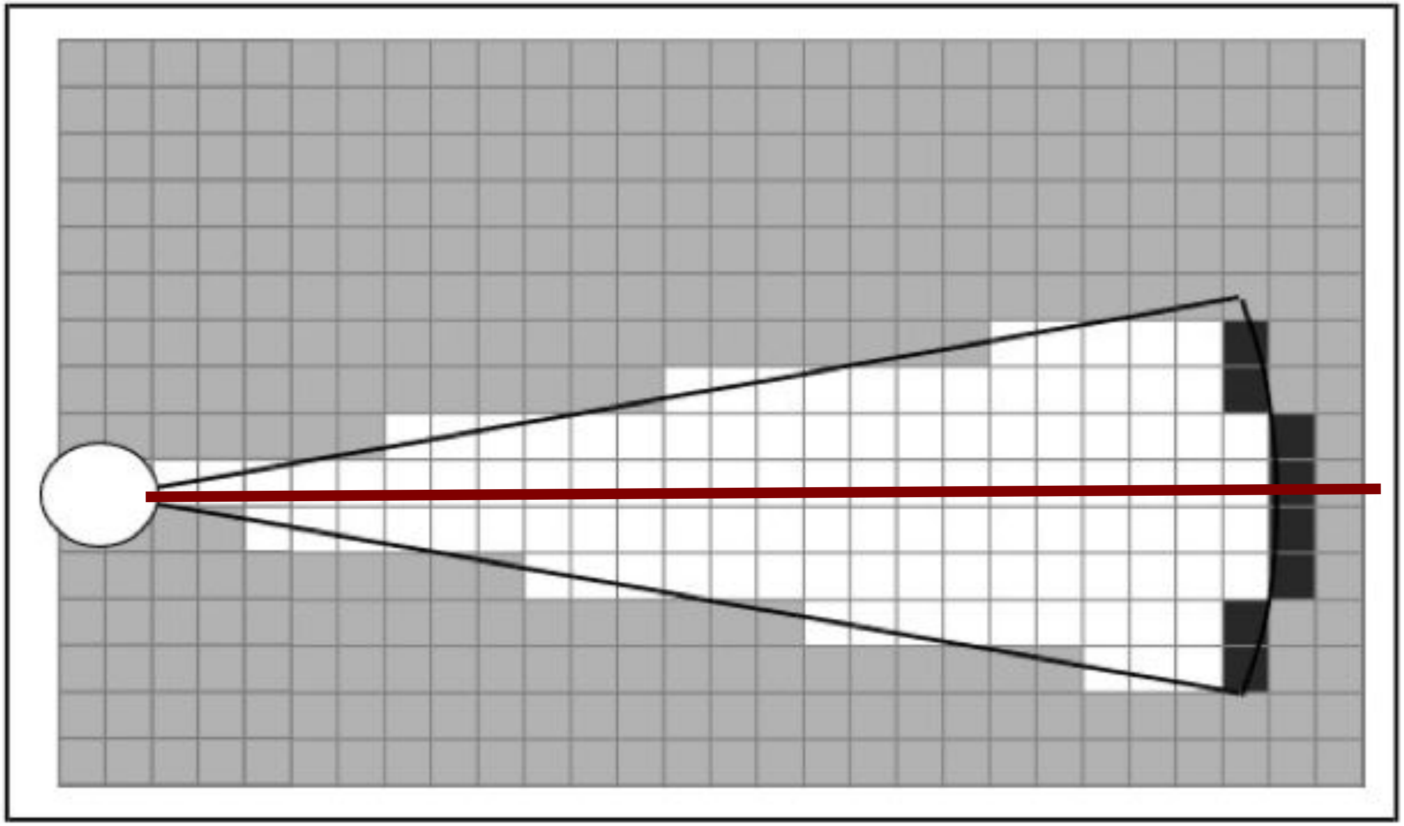
for all cells m_i *do*
    if m_i *in perceptual field of* z_t *then*
        $l_{t,i} = l_{t-1,i} + \text{inv\_sensor\_model}(m_i, x_t, z_t) - l_0$
    else
        $l_{t,i} = l_{t-1,i}$
    endif
endfor
return $\{l_{t,i}\}$

highly efficient, only requires to compute sums

Occupancy Grid Mapping

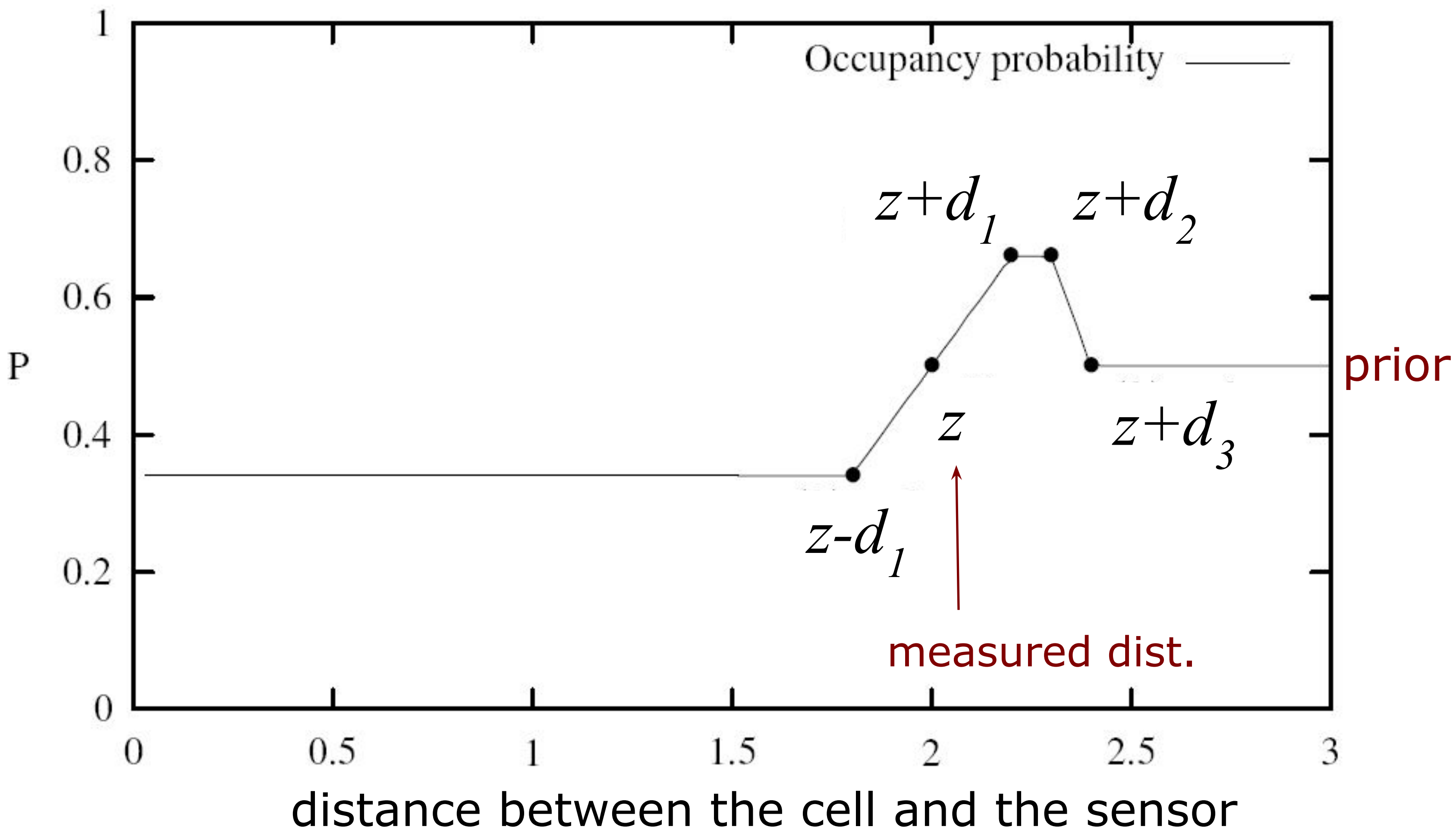
- Developed in the mid 80’s by Moravec and Elfes
- Originally developed for noisy sonars
- Also called “mapping with know poses”

Inverse Sensor Model for Sonars Range Sensors

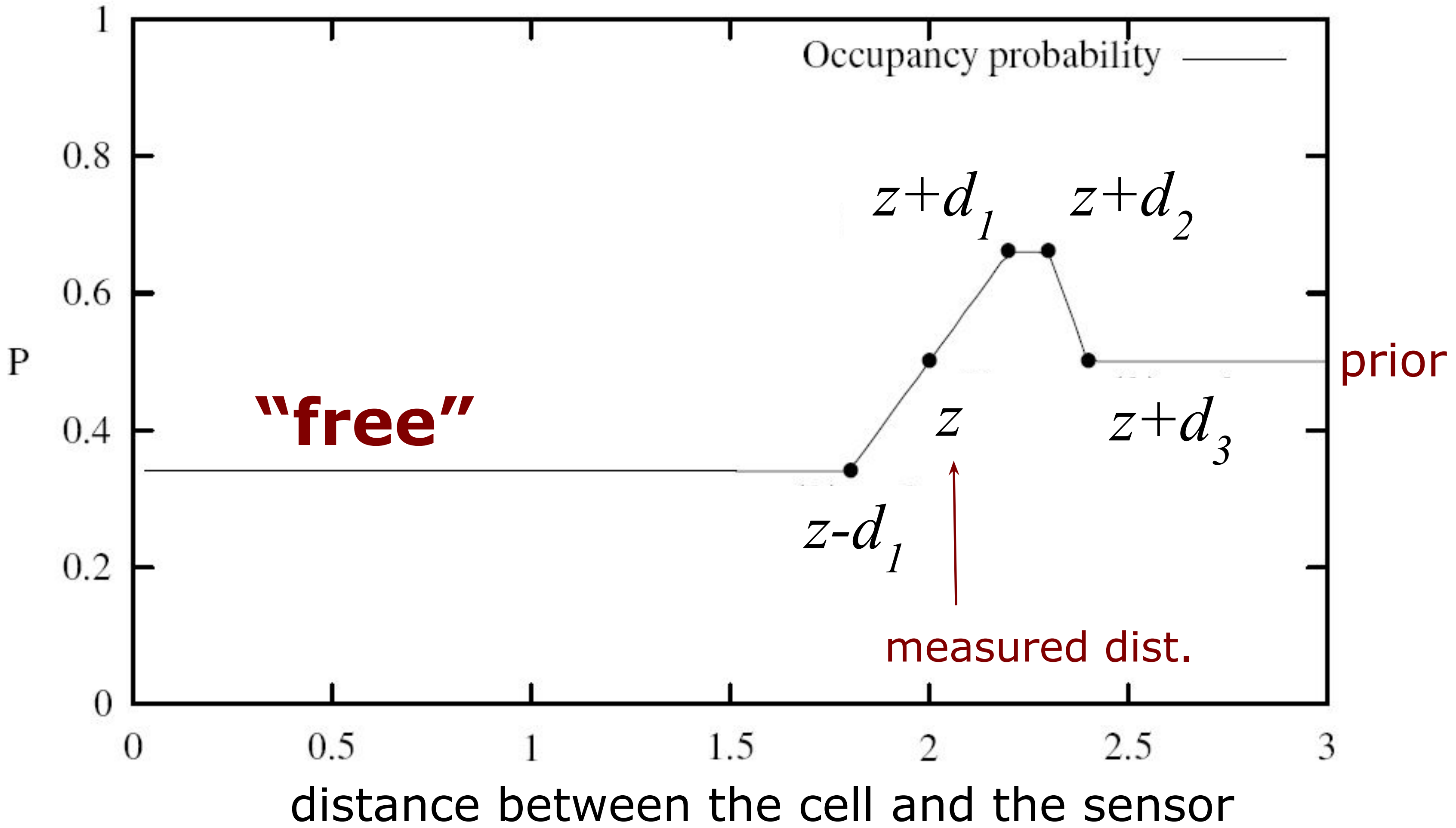


In the following, consider the cells along the optical axis (red line)

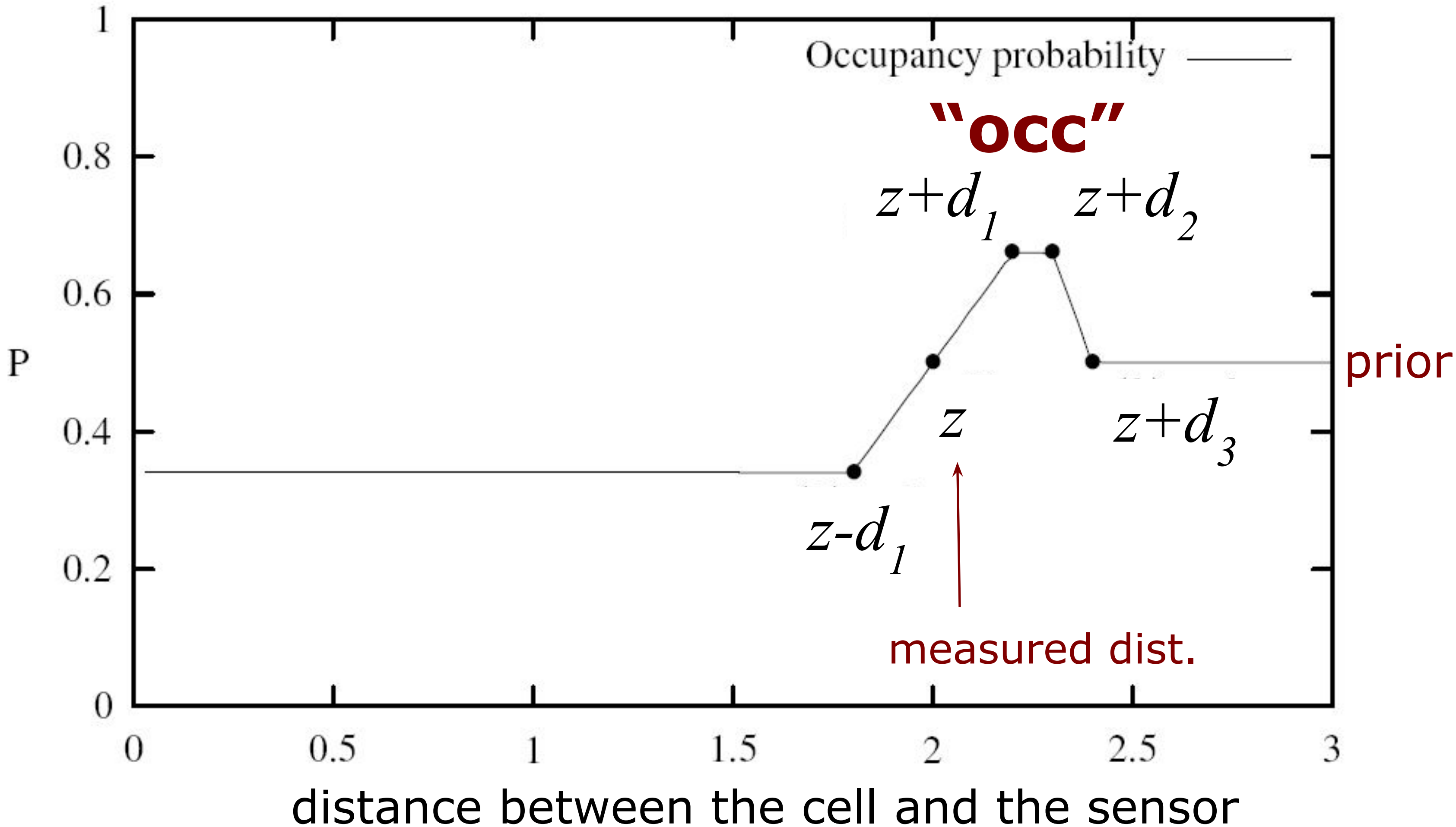
Occupancy Value Depending on the Measured Distance



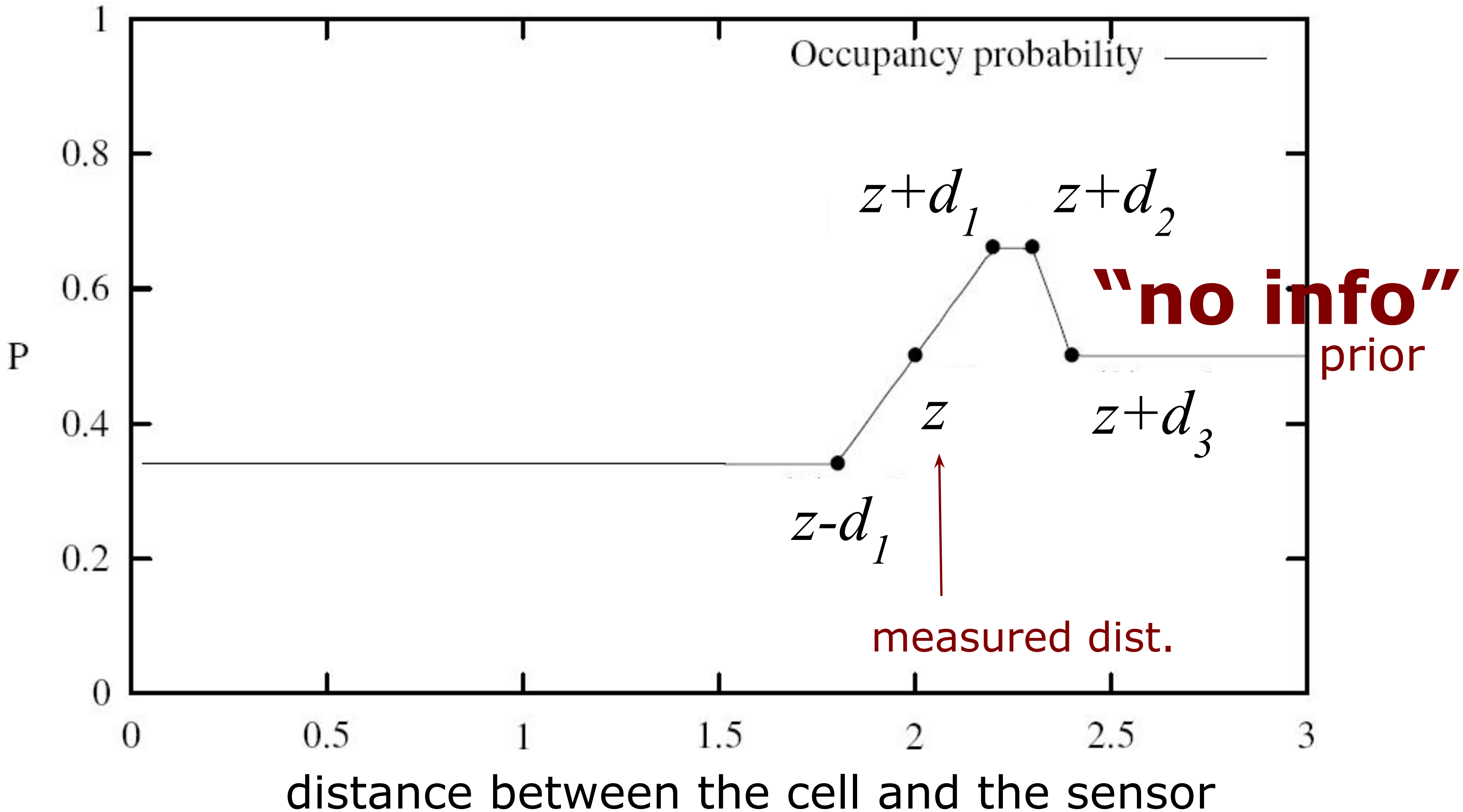
Occupancy Value Depending on the Measured Distance



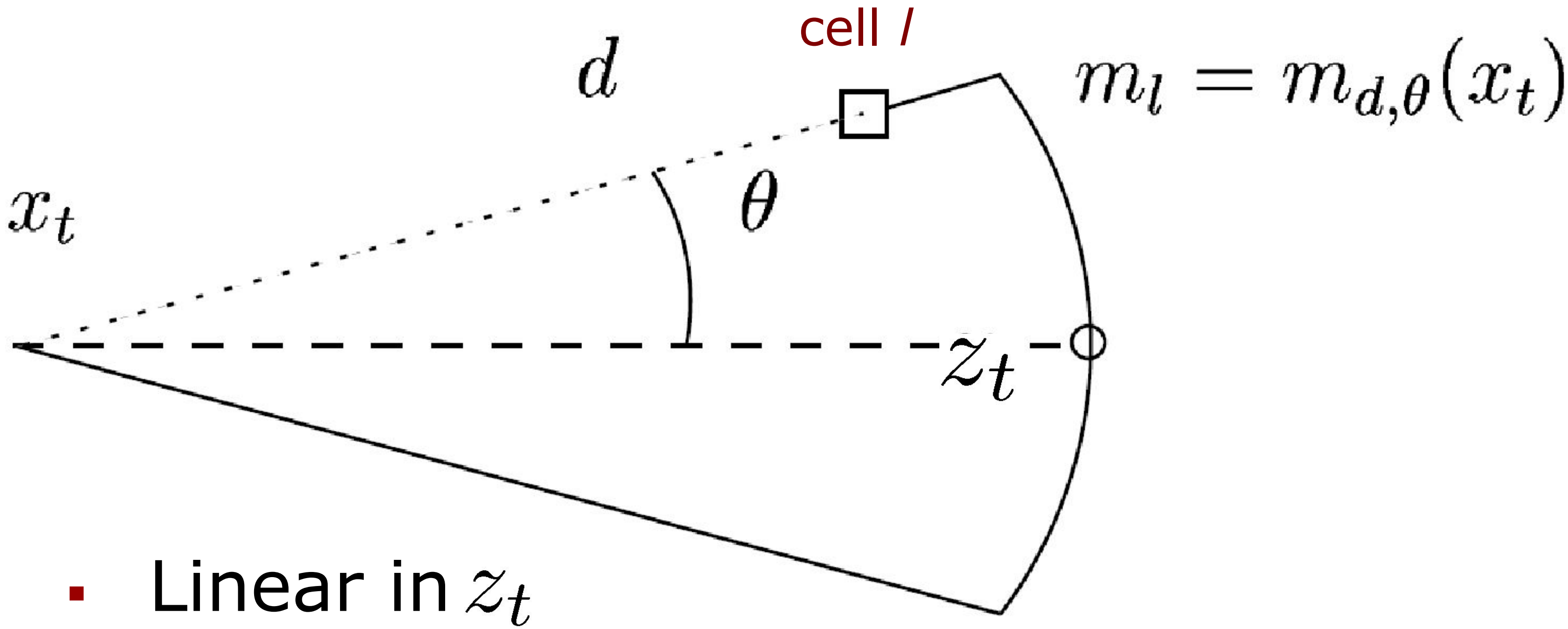
Occupancy Value Depending on the Measured Distance



Occupancy Value Depending on the Measured Distance

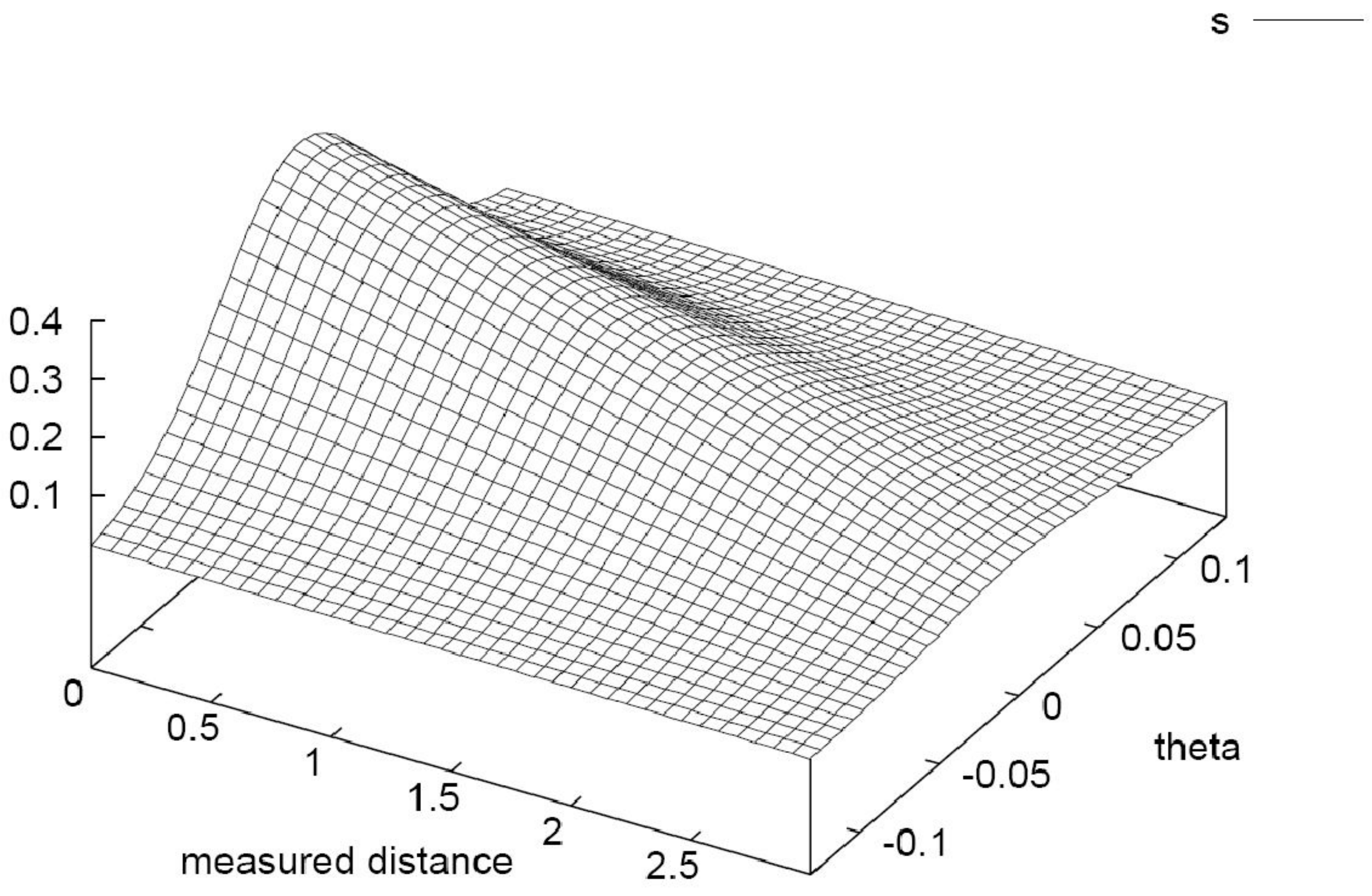


Update depends on the Measured Distance and Deviation from the Optical Axis

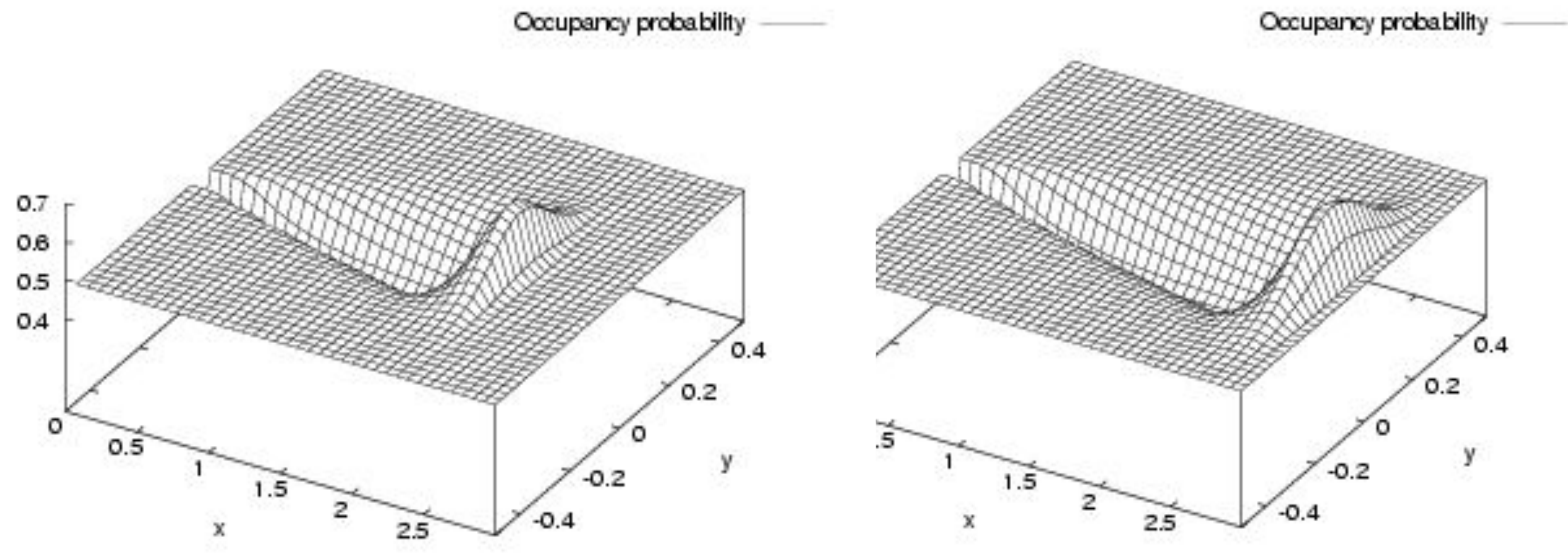


- Linear in z_t
- Gaussian in θ

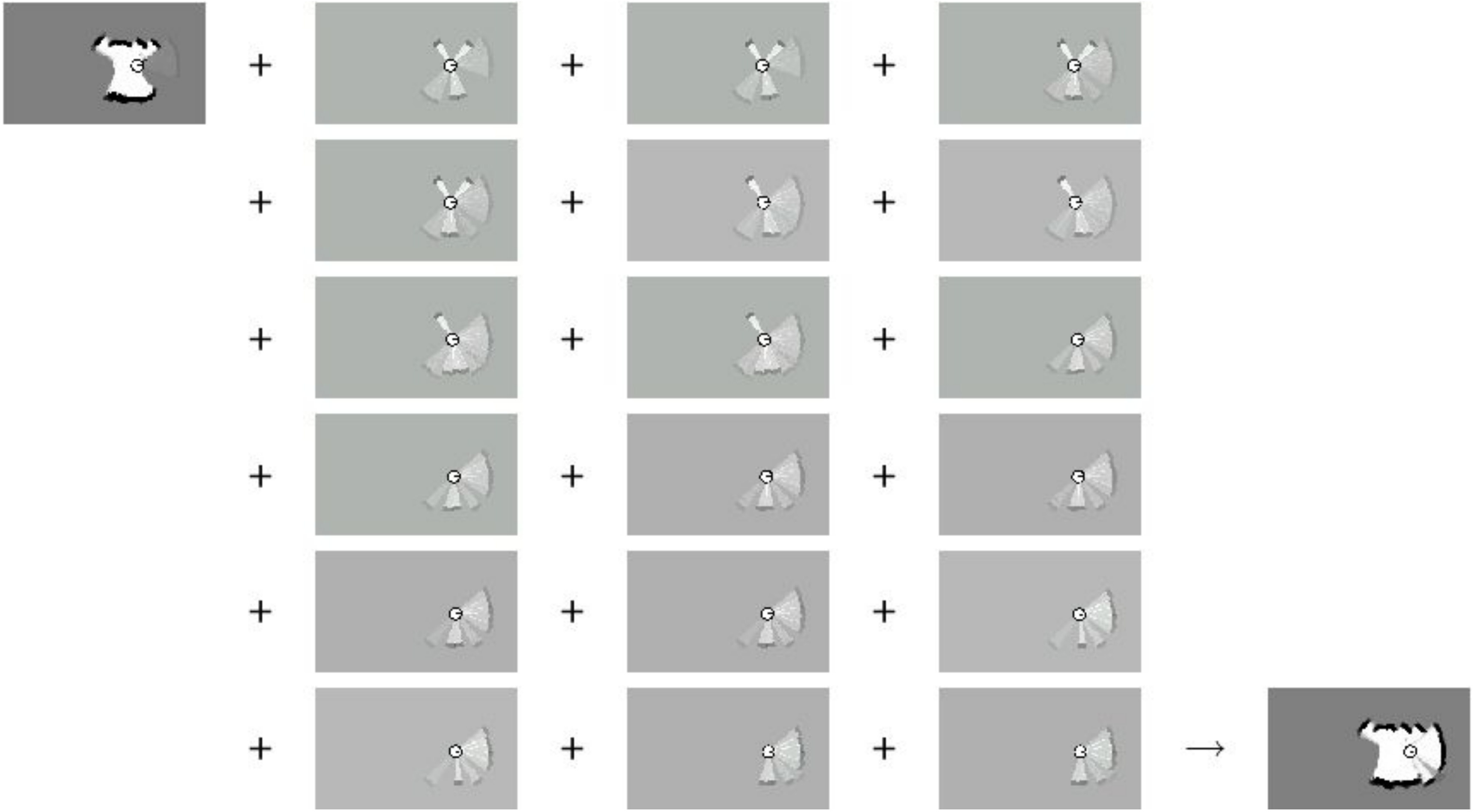
Intensity of the Update



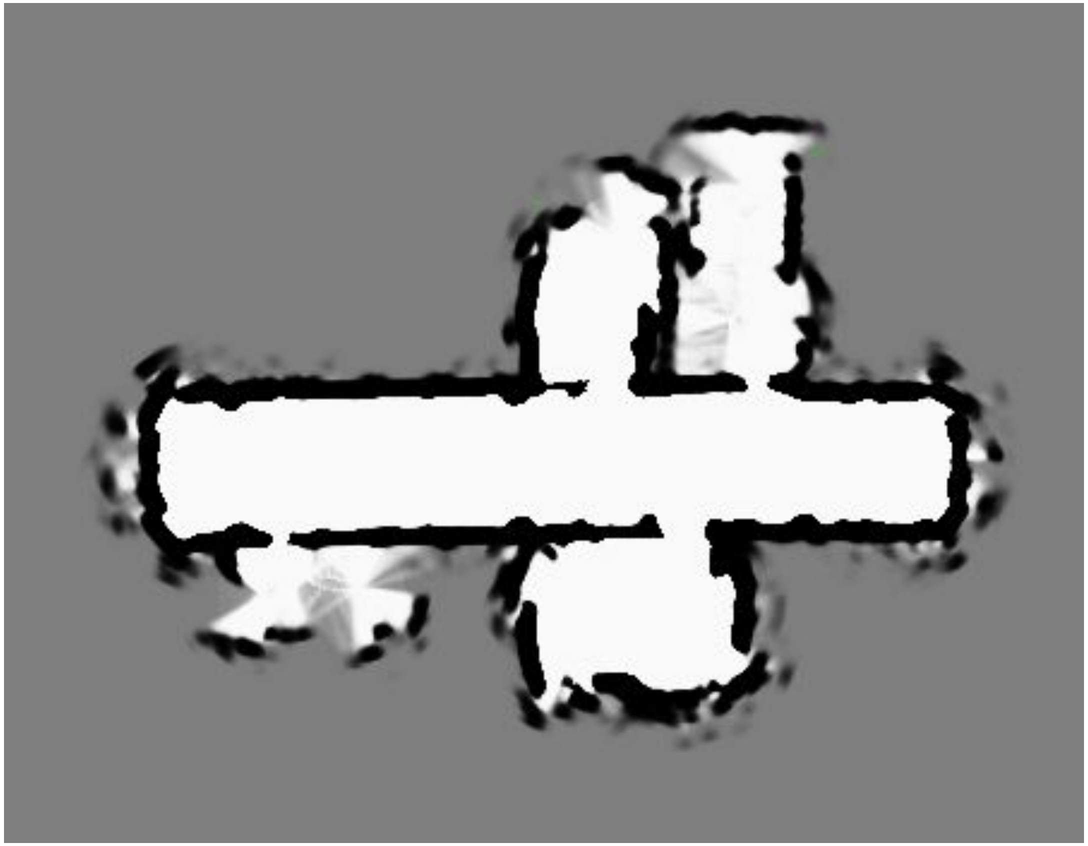
Resulting Model $p(m_i \mid z_t, x_t)$



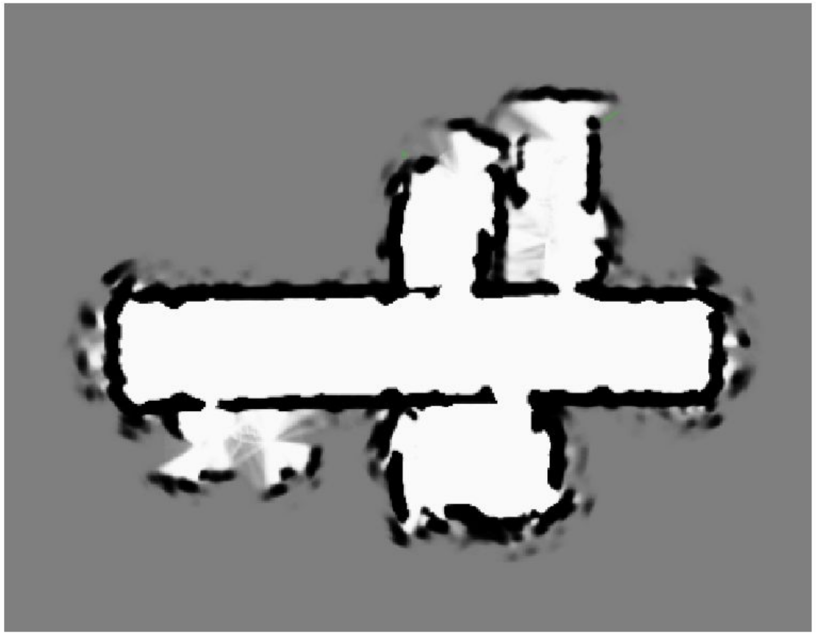
Example: Incremental Updating of Occupancy Grids

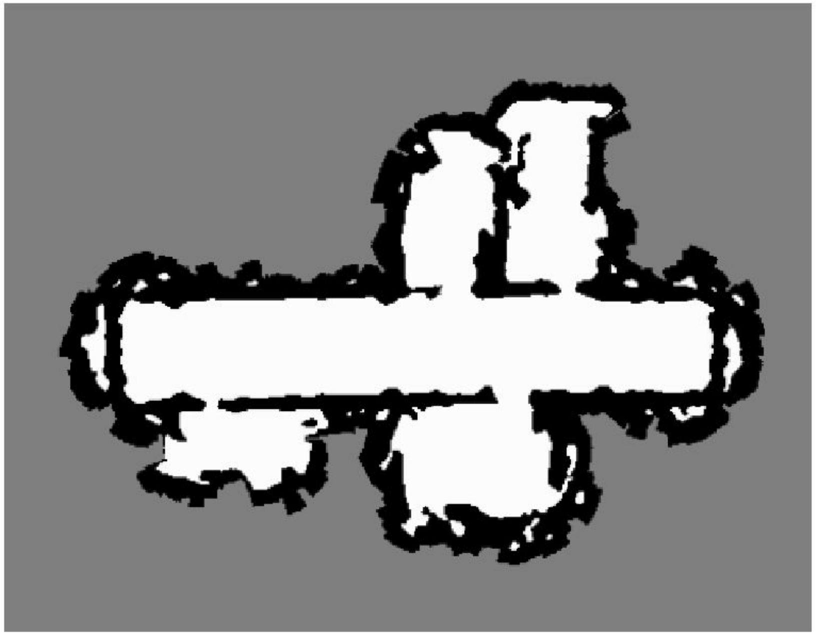


Resulting Map Obtained with Ultrasound Sensors



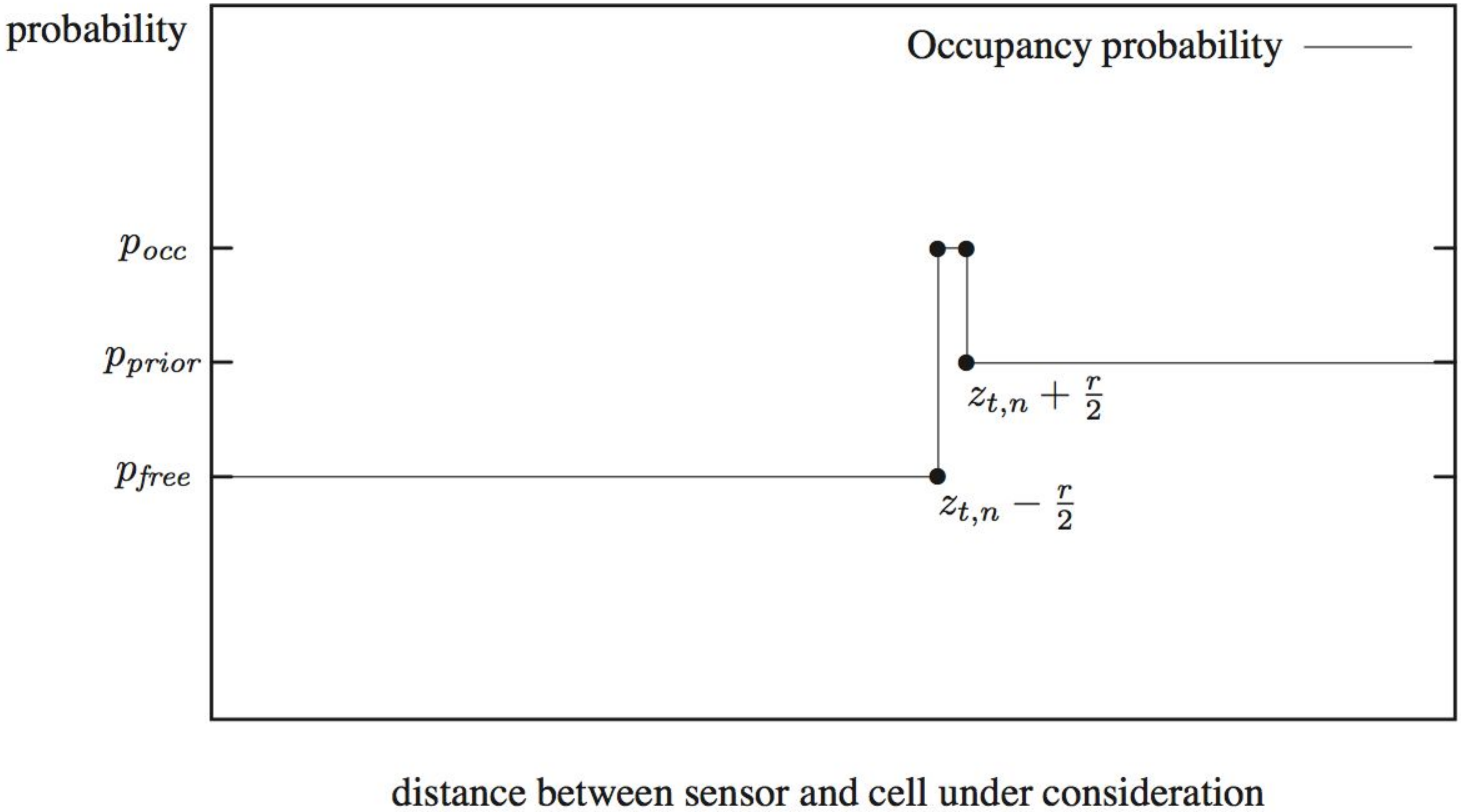
Resulting Occupancy and Maximum Likelihood Map



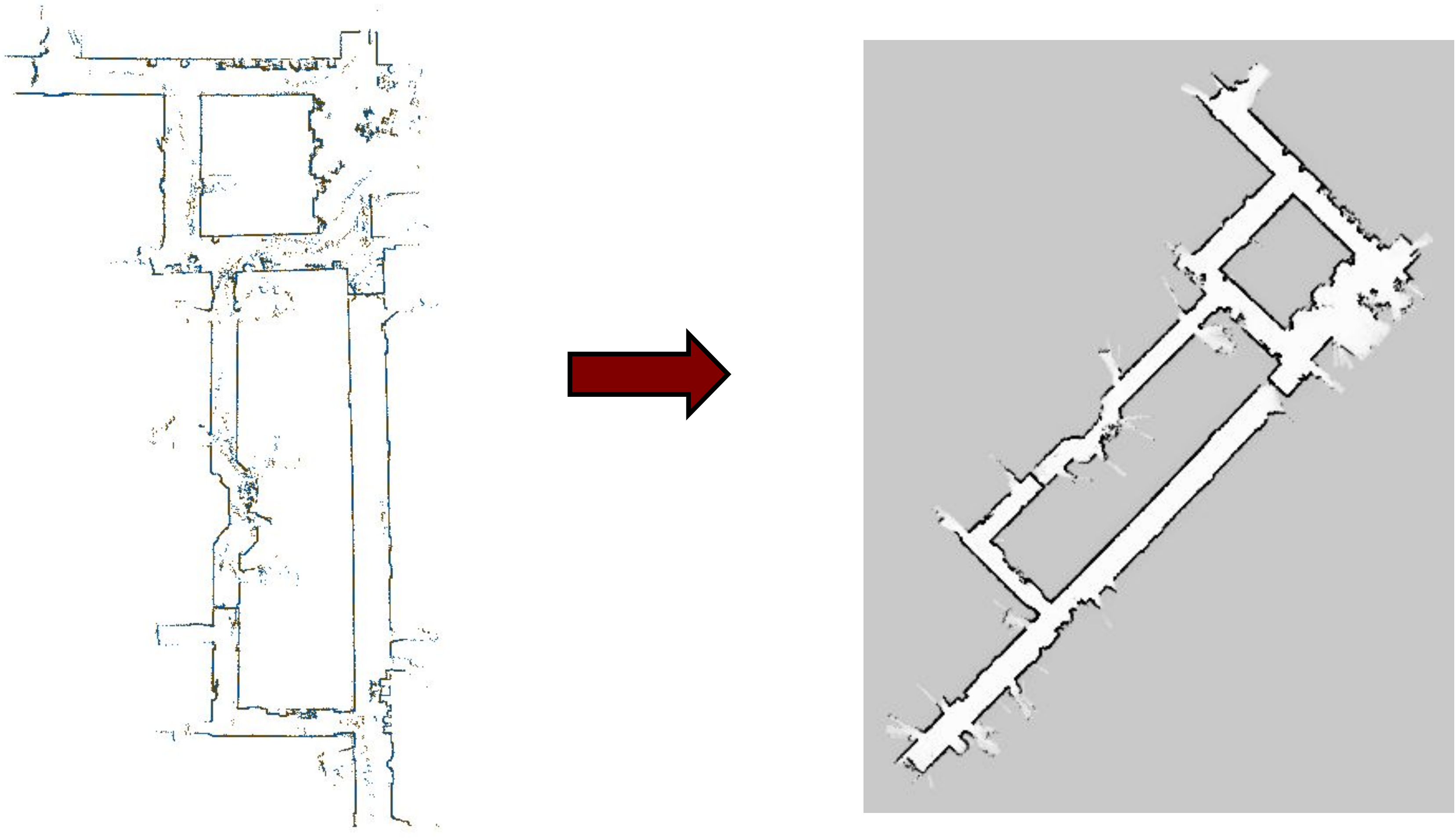


The maximum likelihood map is obtained by rounding the probability for each cell to 0 or 1.

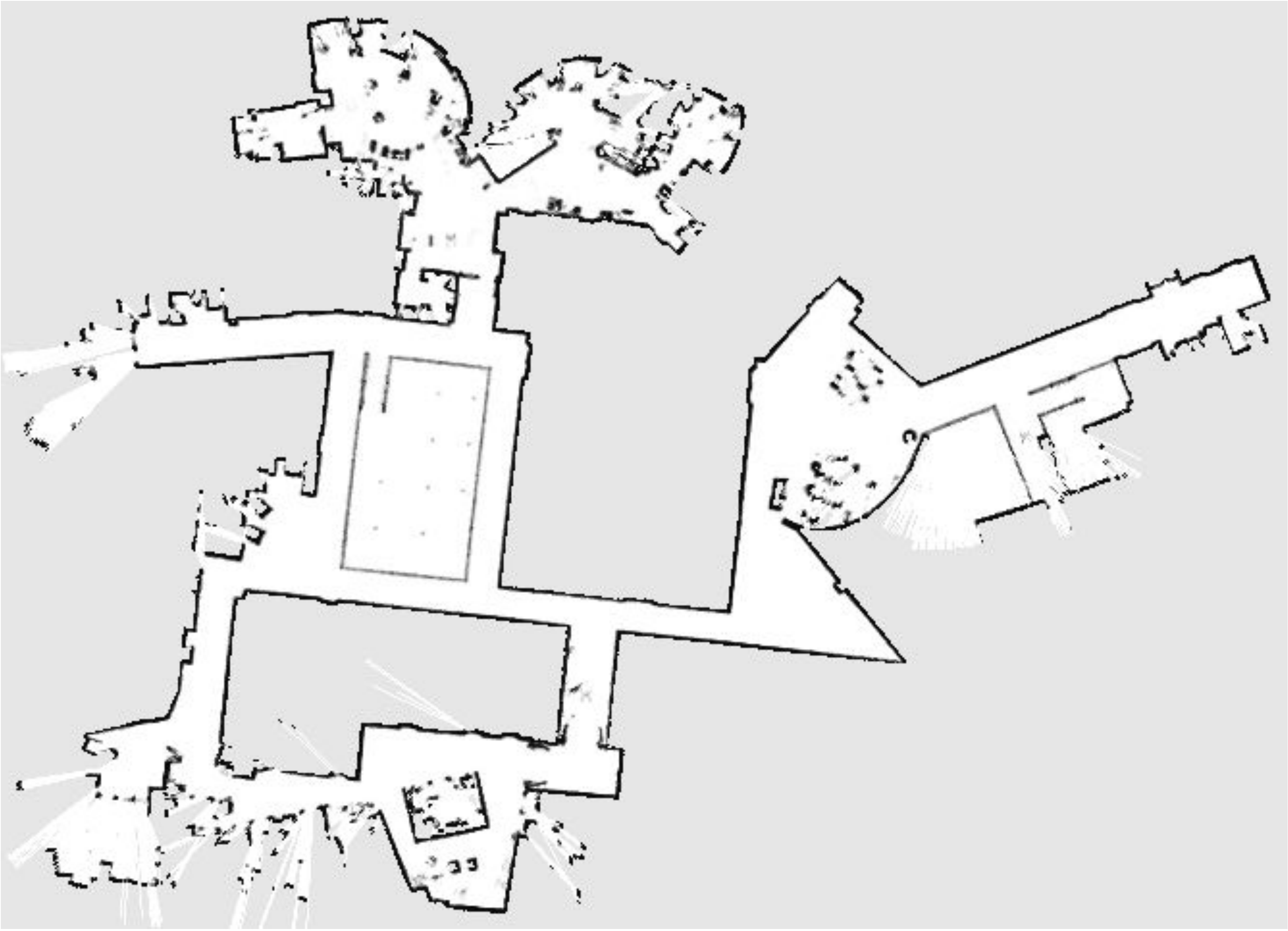
Inverse Sensor Model for Laser Range Finders



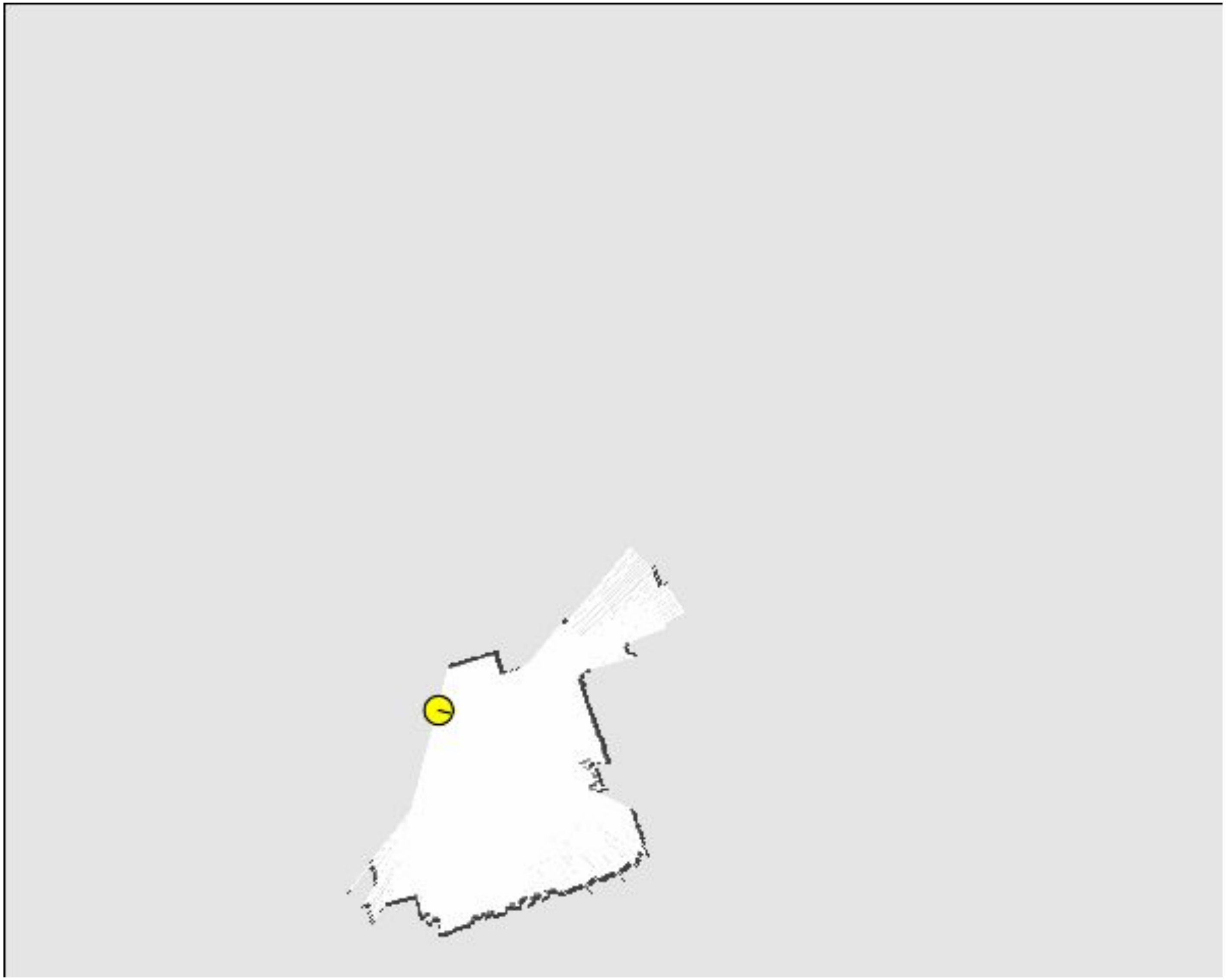
Occupancy Grids
From Laser Scans to Maps



Example: MIT CSAIL 3rd Floor



Uni Freiburg Building 106



Summary OGM

- Grid maps are a popular model for representing the environment
- Occupancy grid maps discretize the space into independent cells, where each cell is a binary random variable estimating if the cell is occupied.
- We efficiently estimate the state of every cell using a binary Bayes filter
- The log odds model is fast to compute
- False independence assumptions are used. More consistent approaches are ongoing research.