

# **Cognitive Robotics**

## **07. Mapping with Known Poses**

AbdElMoniem Bayoumi, PhD

Fall 2022

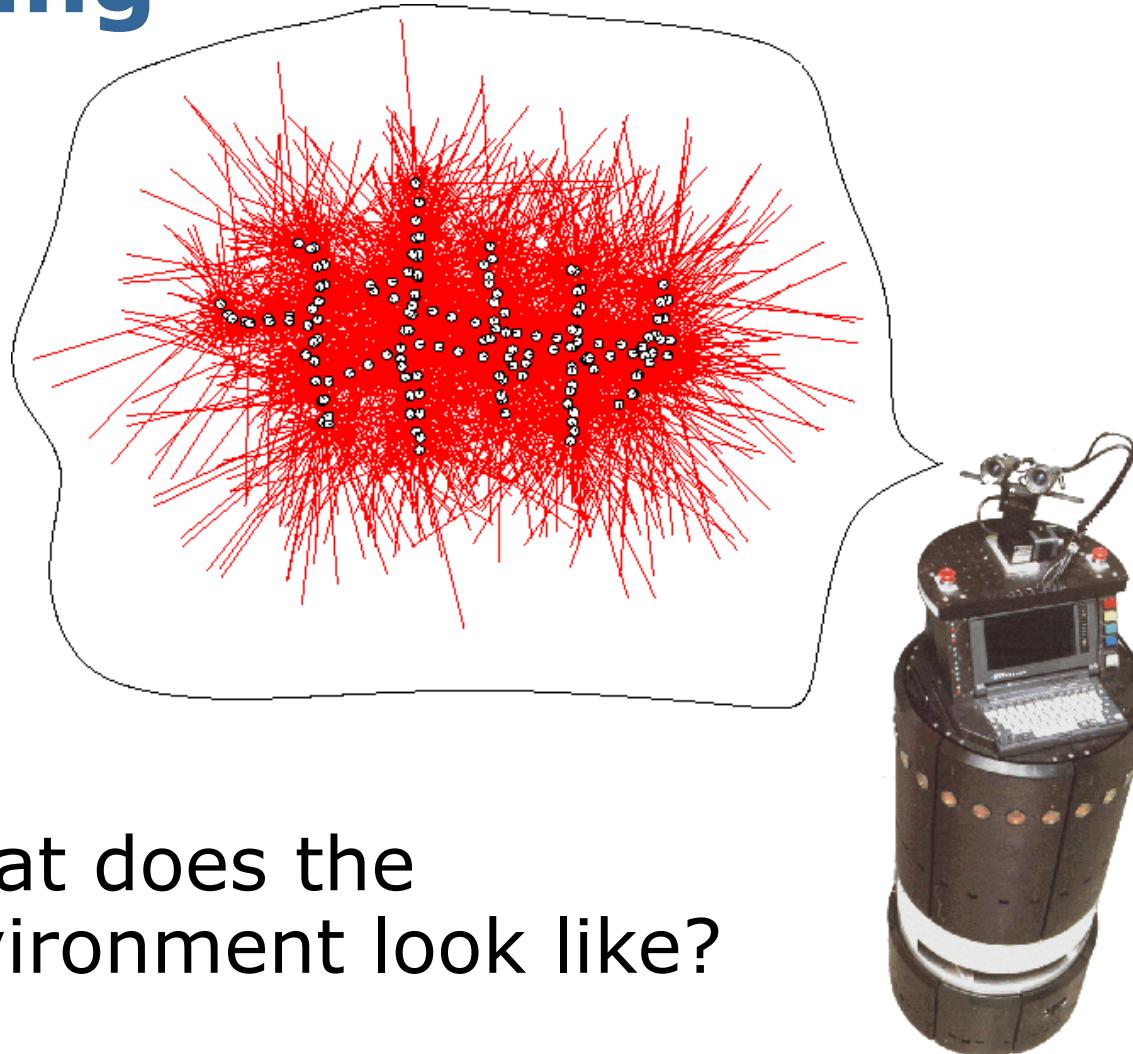
# Acknowledgment

- These slides have been created by Wolfram Burgard, Dieter Fox, Cyrill Stachniss and Maren Bennewitz

# 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

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

- to calculate the most likely map

$$m^* = \operatorname{argmax}_m P(m|d)$$

# 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

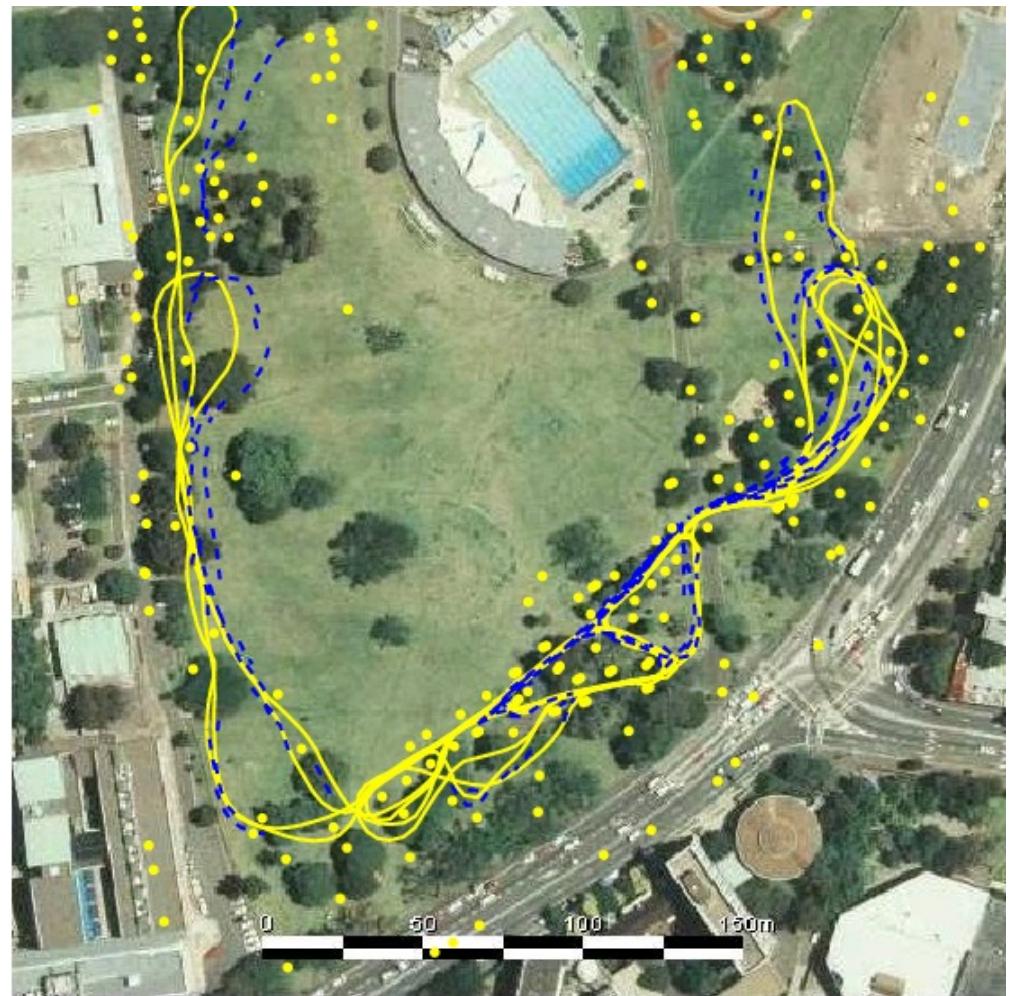
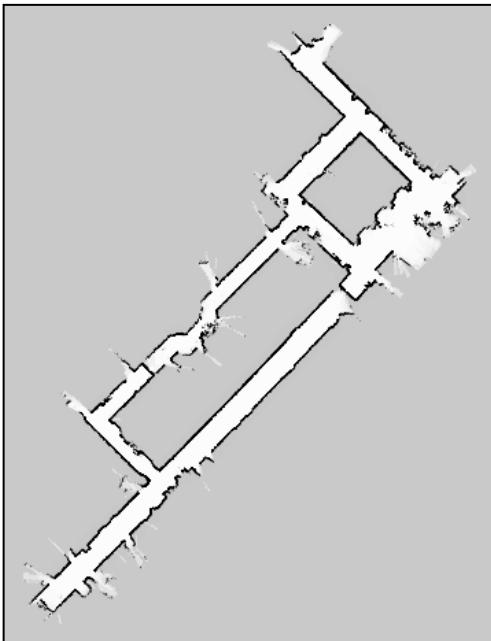
$$m^* = \operatorname{argmax}_m P(m|d)$$

- Today we describe **how to calculate a map given the robot's pose**

# Mapping as a Chicken and Egg Problem

- So far we learned how to estimate the pose of the vehicle given the data and the map
- Mapping, however, 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**

# Features vs. Volumetric Maps

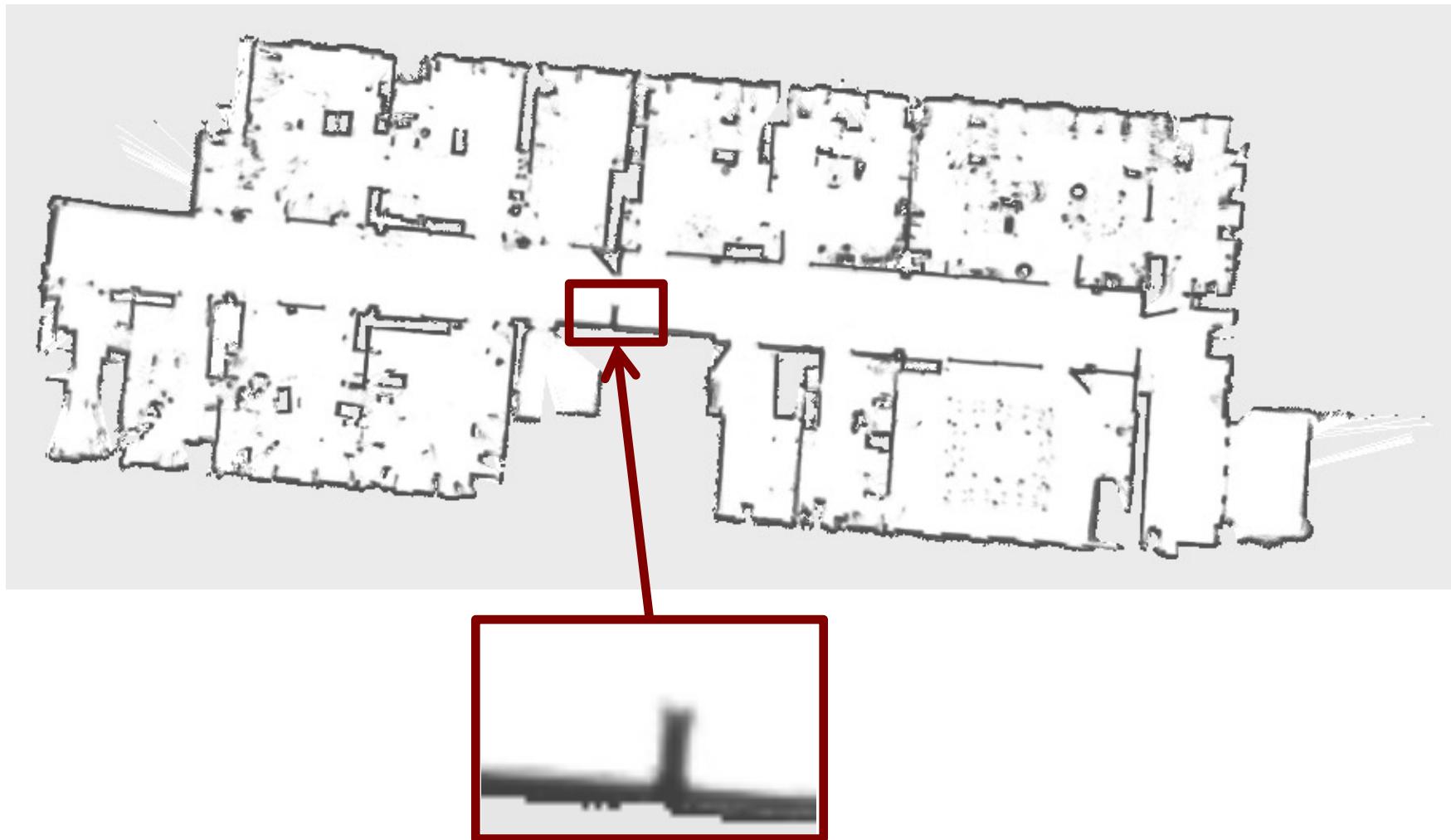


Courtesy by E. Nebot

# Grid Maps

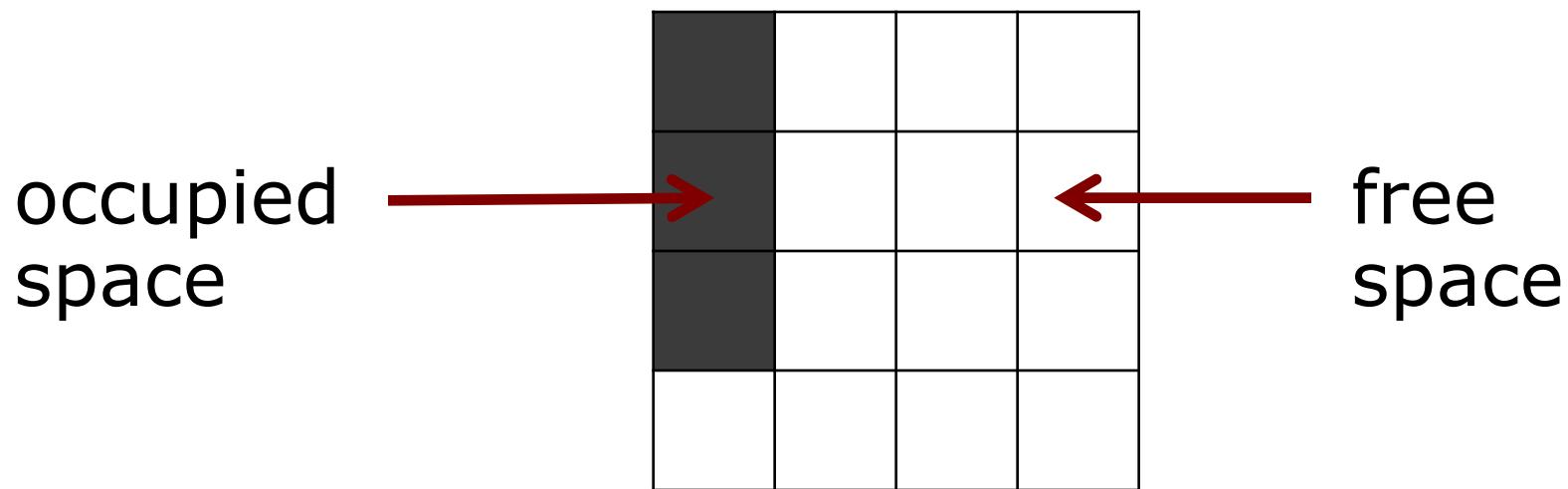
- Discretize the world into cells
- Grid structure is rigid
- Each cell is assumed to be occupied or free space
- Non-parametric model
- Large maps require substantial memory resources
- Do not rely on a feature detector

# Example



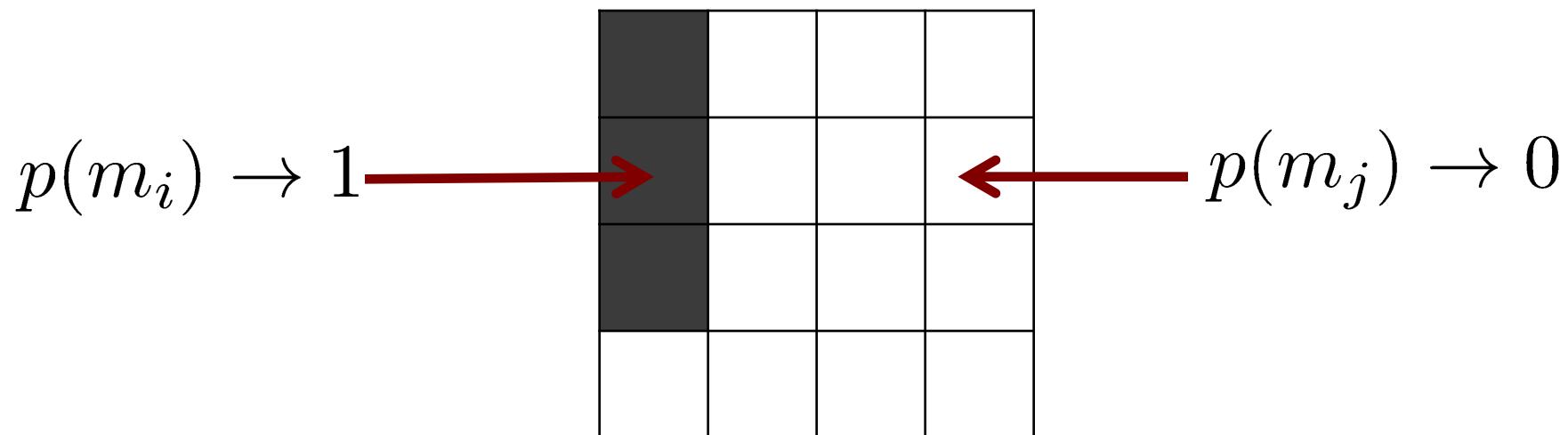
# Assumption 1

- 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
- Cell is occupied:  $p(m_i) = 1$
- Cell is not occupied:  $p(m_i) = 0$
- No knowledge:  $p(m_i) = 0.5$

# Occupancy Probability Example

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



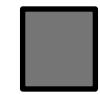
$$P(M_i = \text{occ}) = p(m_i) = 1$$

$$P(M_i = \text{free}) = p(\neg m_i) = 1 - p(m_i) = 0$$



$$P(M_i = \text{occ}) = p(m_i) = 0$$

$$P(M_i = \text{free}) = p(\neg m_i) = 1 - p(m_i) = 1$$

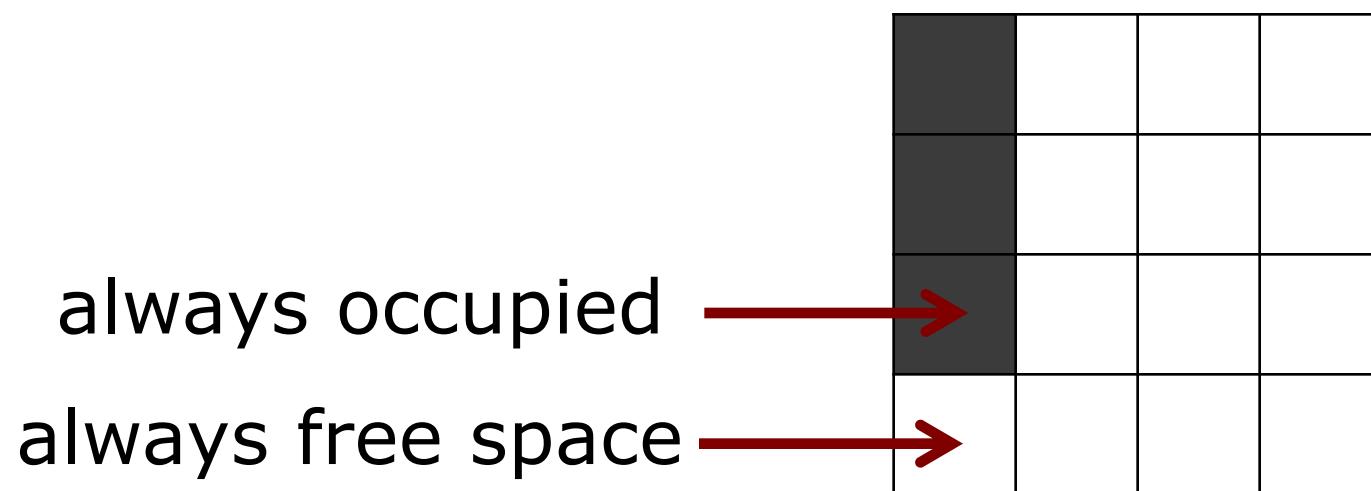


$$P(M_i = \text{occ}) = p(m_i) = 0.75$$

$$P(M_i = \text{free}) = p(\neg m_i) = 0.25$$

## Assumption 2

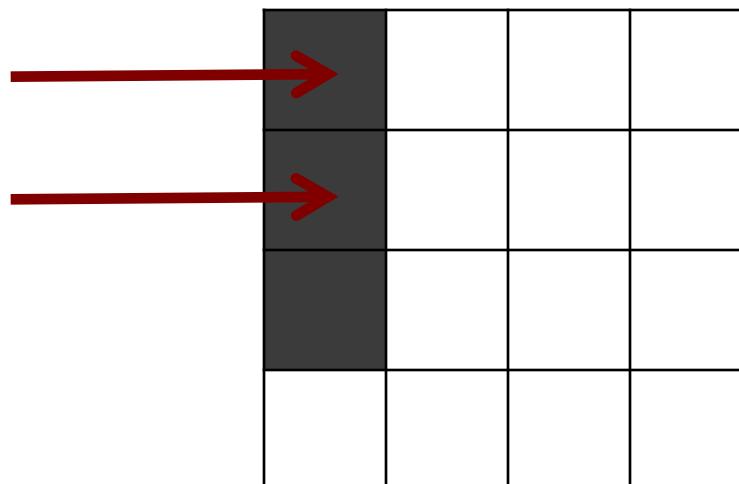
- The world is **static** (most mapping systems make this assumption)



# Assumption 3

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

no dependency  
between the cells



# Joint Distribution

$$p(m) = p(m_1, m_2, \dots, m_N)$$

The diagram illustrates the joint distribution of multiple variables. It features four vertical red arrows pointing upwards, each associated with a label below it: 'map' on the left, 'cell 1' in the middle-left, 'cell N' in the middle-right, and a double-headed arrow between 'cell 1' and 'cell N'. The double-headed arrow is positioned such that its top point is aligned with the top of the arrow for 'cell 1' and its bottom point is aligned with the bottom of the arrow for 'cell N'.

# Representation

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

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

A mathematical equation  $p(m) = \prod_i p(m_i)$  is displayed. Two red arrows point upwards from the words "map" and "cell" to the terms  $p(m)$  and  $i$  respectively. The word "map" is positioned below the first term, and the word "cell" is positioned below the index  $i$ .

# Example A

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

$$p(M = \begin{array}{|c|c|}\hline & \blacksquare \\ \hline \blacksquare & \blacksquare \\ \hline\end{array}) = p(M_1 = \blacksquare)p(M_2 = \blacksquare) \\ p(M_3 = \blacksquare)p(M_4 = \blacksquare)$$

M vs. m to distinguish a configuration  
and the random variable for the map

## Example B

$$\begin{aligned} p(M = \begin{array}{|c|c|}\hline \blacksquare & \square \\ \hline \blacksquare & \square \\ \hline \end{array}) &= p(M_1 = \blacksquare)p(M_2 = \square) \\ &\quad p(M_3 = \blacksquare)p(M_4 = \square) \\ &= p(M_1 = \blacksquare)(1 - p(M_2 = \blacksquare)) \\ &\quad (1 - p(M_3 = \blacksquare))p(M_4 = \blacksquare) \end{aligned}$$

M vs. m to distinguish a configuration  
and the random variable for the map

# Estimating a Map From Data

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

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



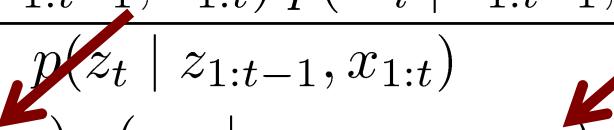
binary random variable

→ Binary Bayes filter  
(for a static state)

# Static State Binary Bayes Filter

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

# Static State Binary Bayes Filter

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


# Static State Binary Bayes Filter

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

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

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



# Static State Binary Bayes Filter

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

# Static State Binary Bayes Filter

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

# Static State Binary Bayes Filter

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

Do exactly the same for the opposite event:

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

# Static State Binary Bayes Filter

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

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

# Static State Binary Bayes Filter

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

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

# Static State Binary Bayes Filter

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

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

# From Ratio to Probability

- We can easily turn the ratio into the probability

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

$$p(x) = Y - Y p(x)$$

$$p(x) (1 + Y) = Y$$

$$p(x) = \frac{Y}{1 + Y}$$

$$p(x) = \frac{1}{1 + \frac{1}{Y}}$$

# From Ratio to Probability

- Using  $p(x) = [1 + Y^{-1}]^{-1}$  directly leads to

$$\begin{aligned} p(m_i \mid z_{1:t}, x_{1:t}) \\ = \left[ 1 + \frac{1 - p(m_i \mid z_t, x_t)}{p(m_i \mid z_t, x_t)} \frac{1 - p(m_i \mid z_{1:t-1}, x_{1:t-1})}{p(m_i \mid z_{1:t-1}, x_{1:t-1})} \frac{p(m_i)}{1 - p(m_i)} \right]^{-1} \end{aligned}$$

**For reasons of efficiency, one performs the calculations in the log odds notation**

# Log Odds Notation

- The log odds notation computes the logarithm of the ratio of probabilities

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

$$\rightarrow l(m_i \mid z_{1:t}, x_{1:t}) = \log \left( \frac{p(m_i \mid z_{1:t}, x_{1:t})}{1 - p(m_i \mid z_{1:t}, x_{1:t})} \right)$$

# Log Odds Notation

- Log odds ratio is defined as

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

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

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

# Occupancy Mapping in Log Odds Form

- The product turns into a sum

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

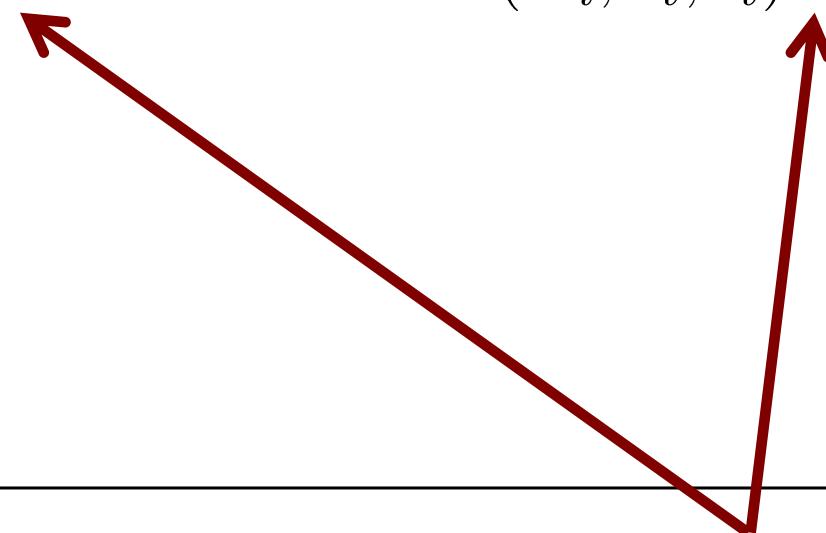
- or in short

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

# Occupancy Mapping Algorithm

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

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

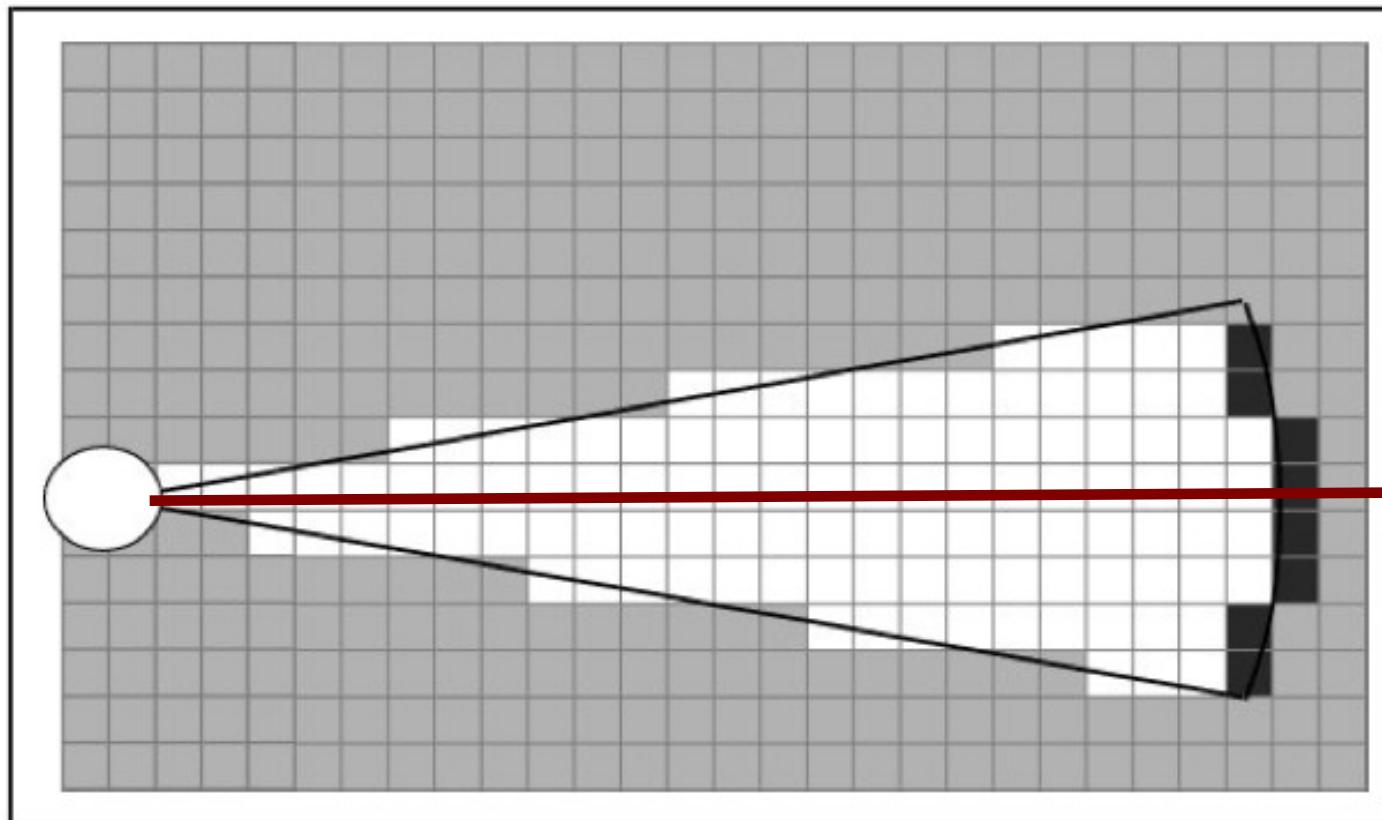


**highly efficient, we only have to compute sums**

# Occupancy Grid Mapping

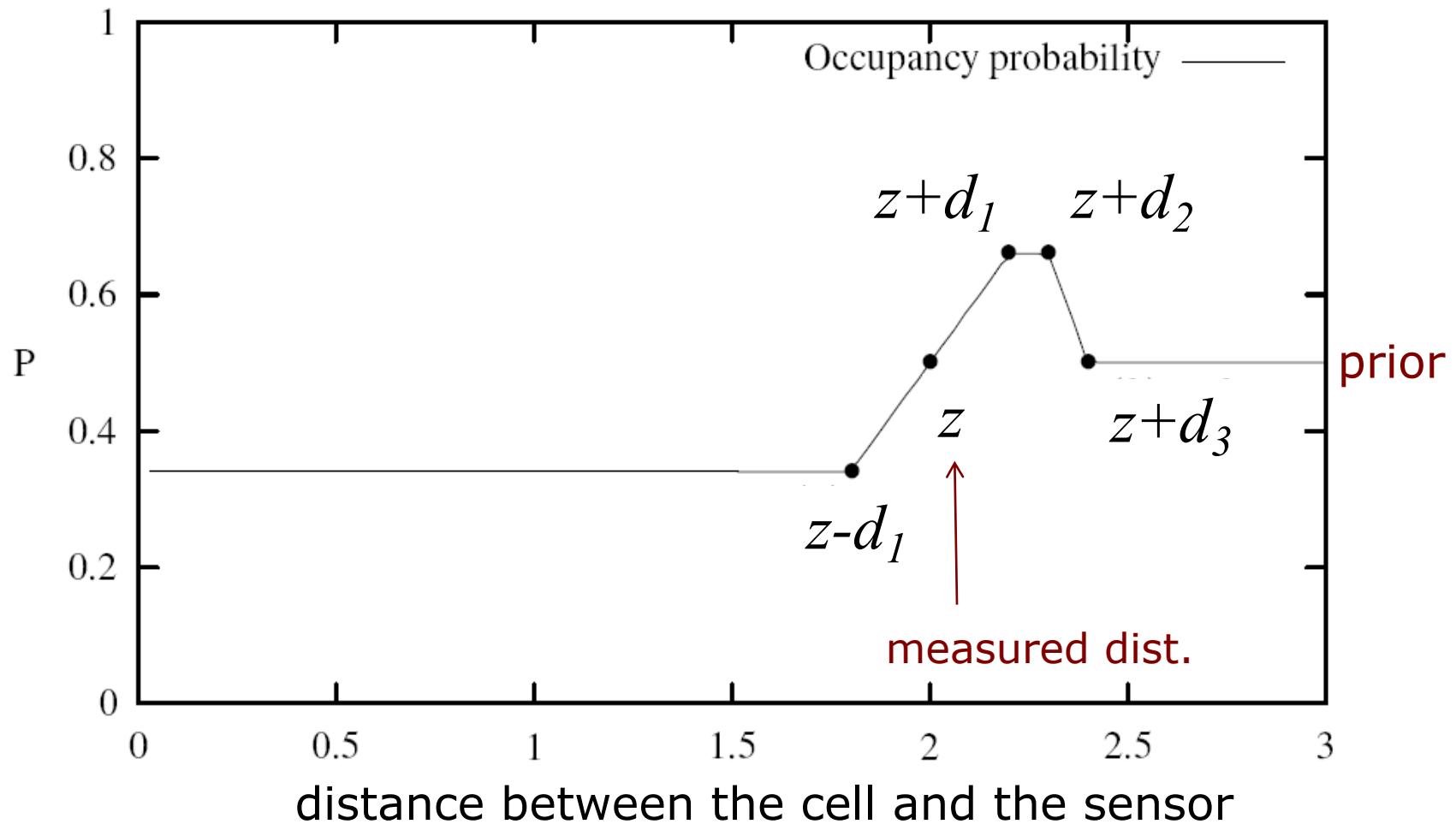
- Moravec and Elfes proposed occupancy grid mapping in the mid 80'ies
- Developed for noisy sonar sensors

# Inverse Sensor Model for Sonar Range Sensors

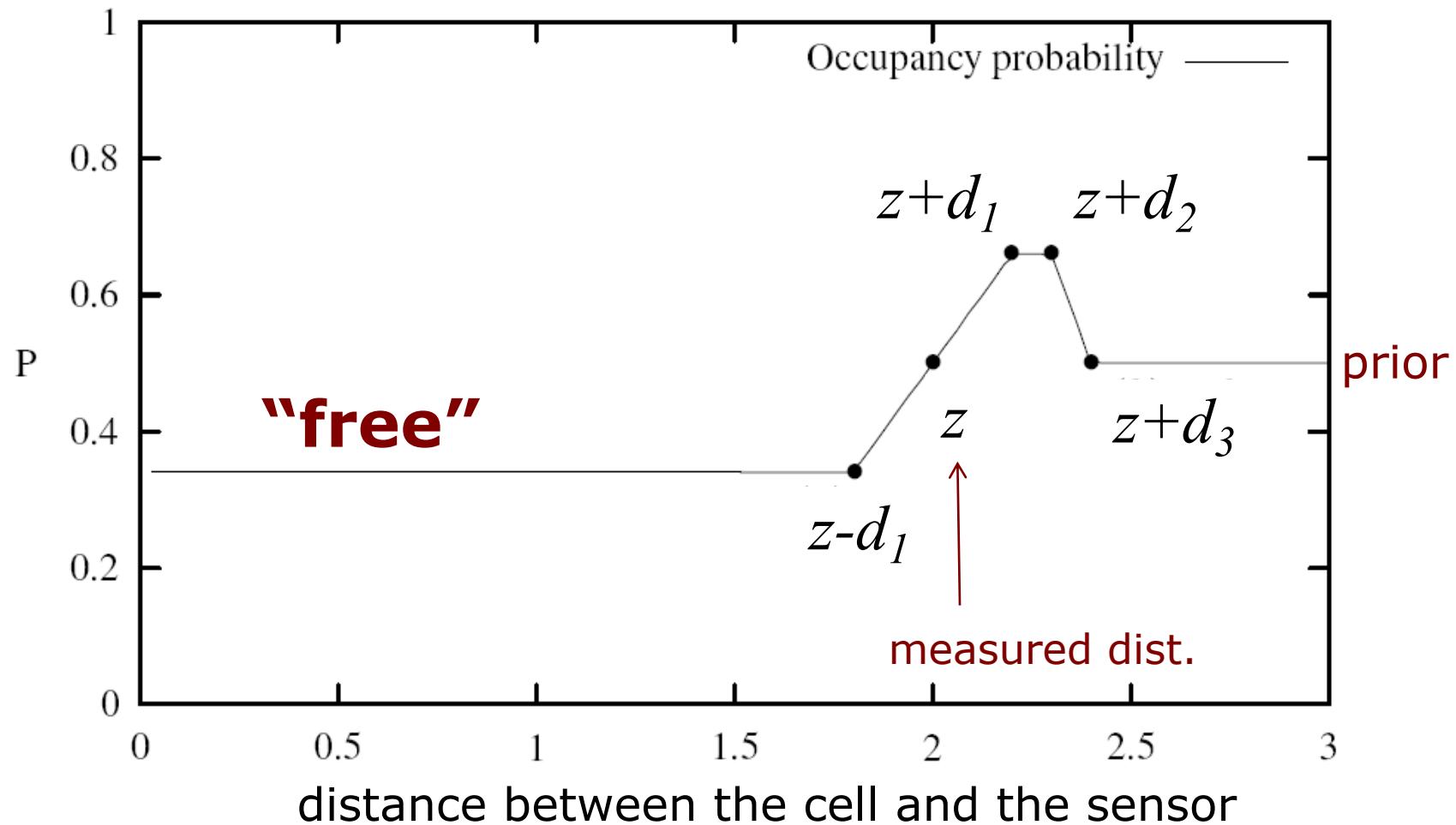


In the following, consider the cells along the optical axis (red line)

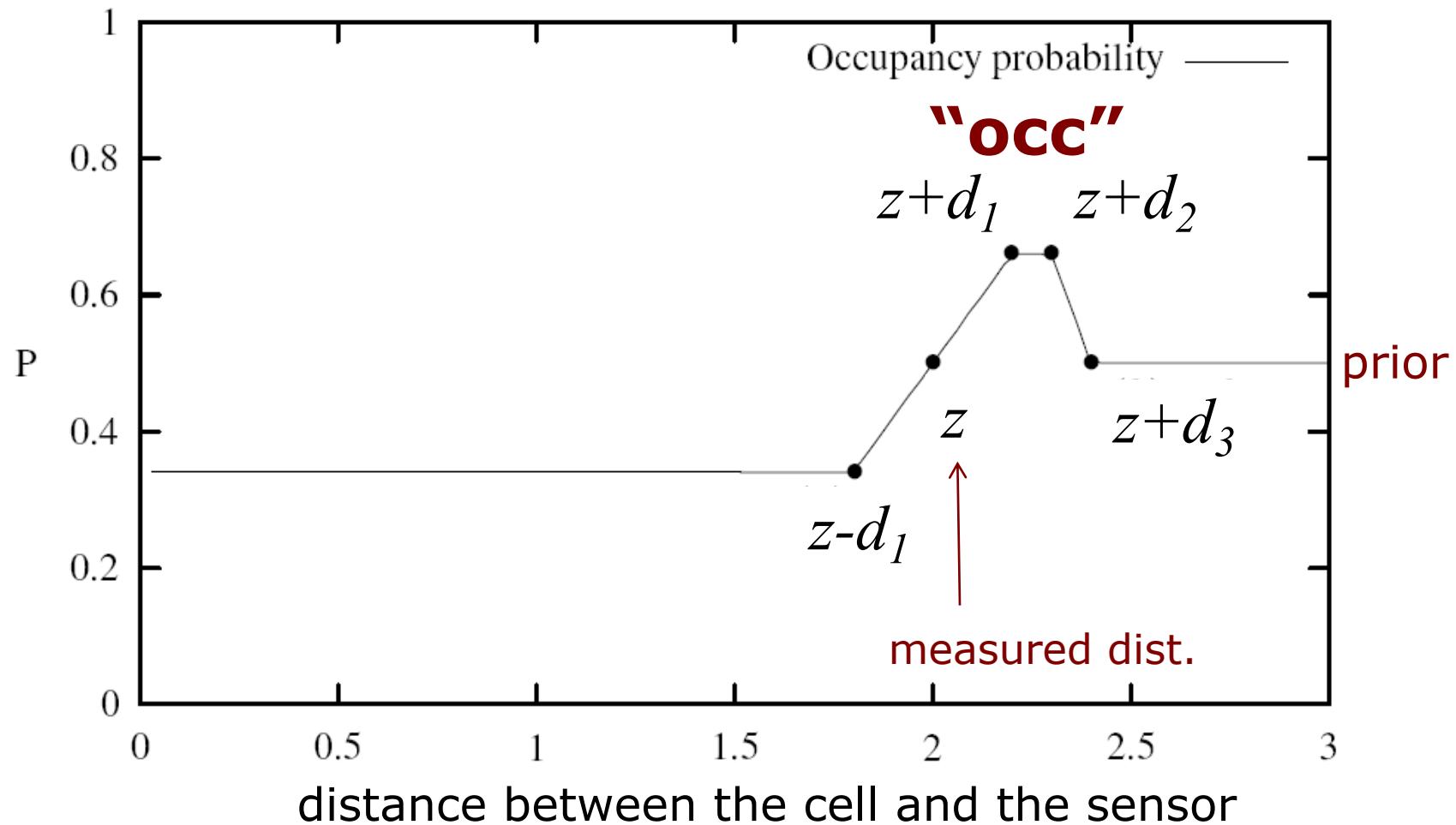
# Occupancy Value Depending on the Measured Distance



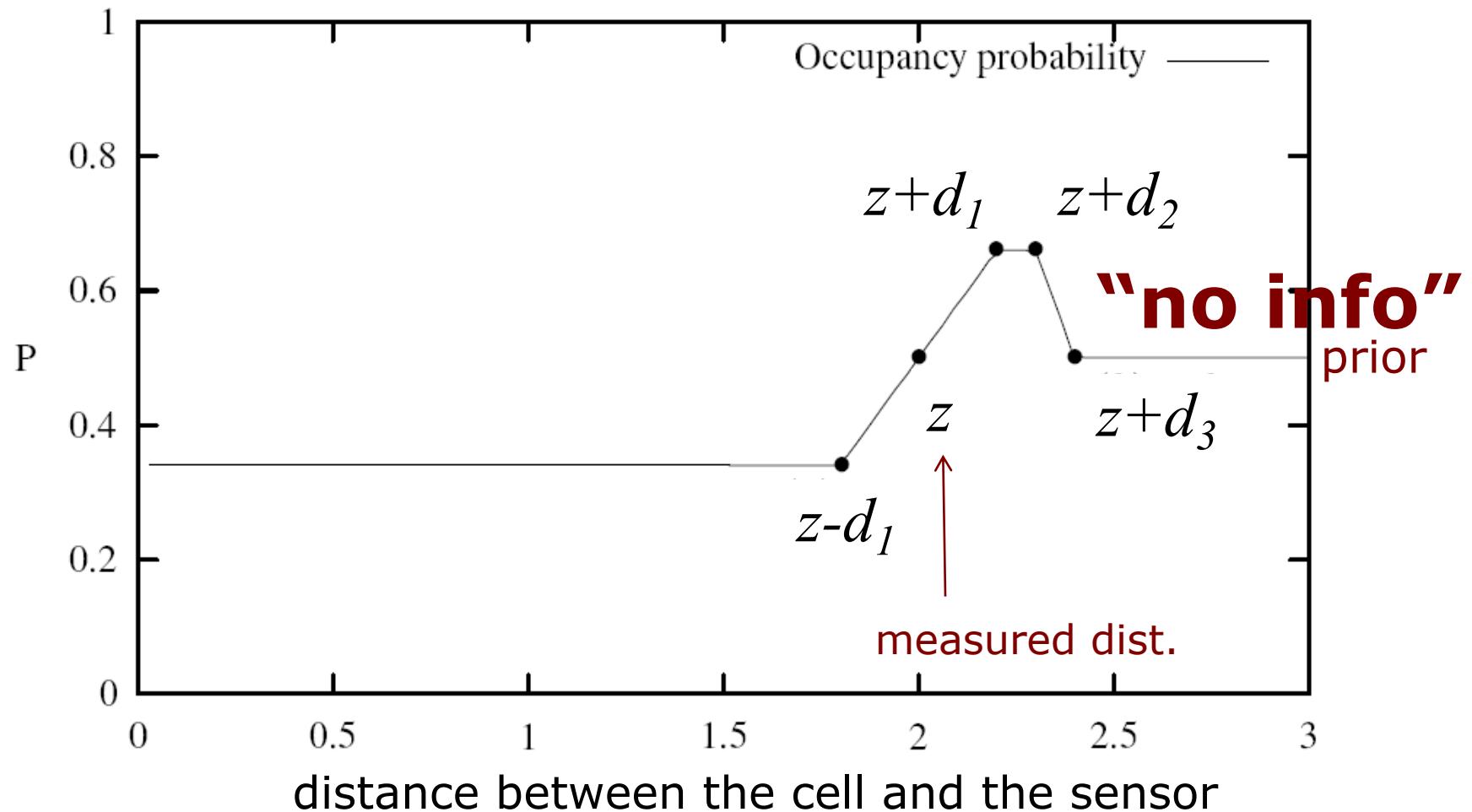
# Occupancy Value Depending on the Measured Distance



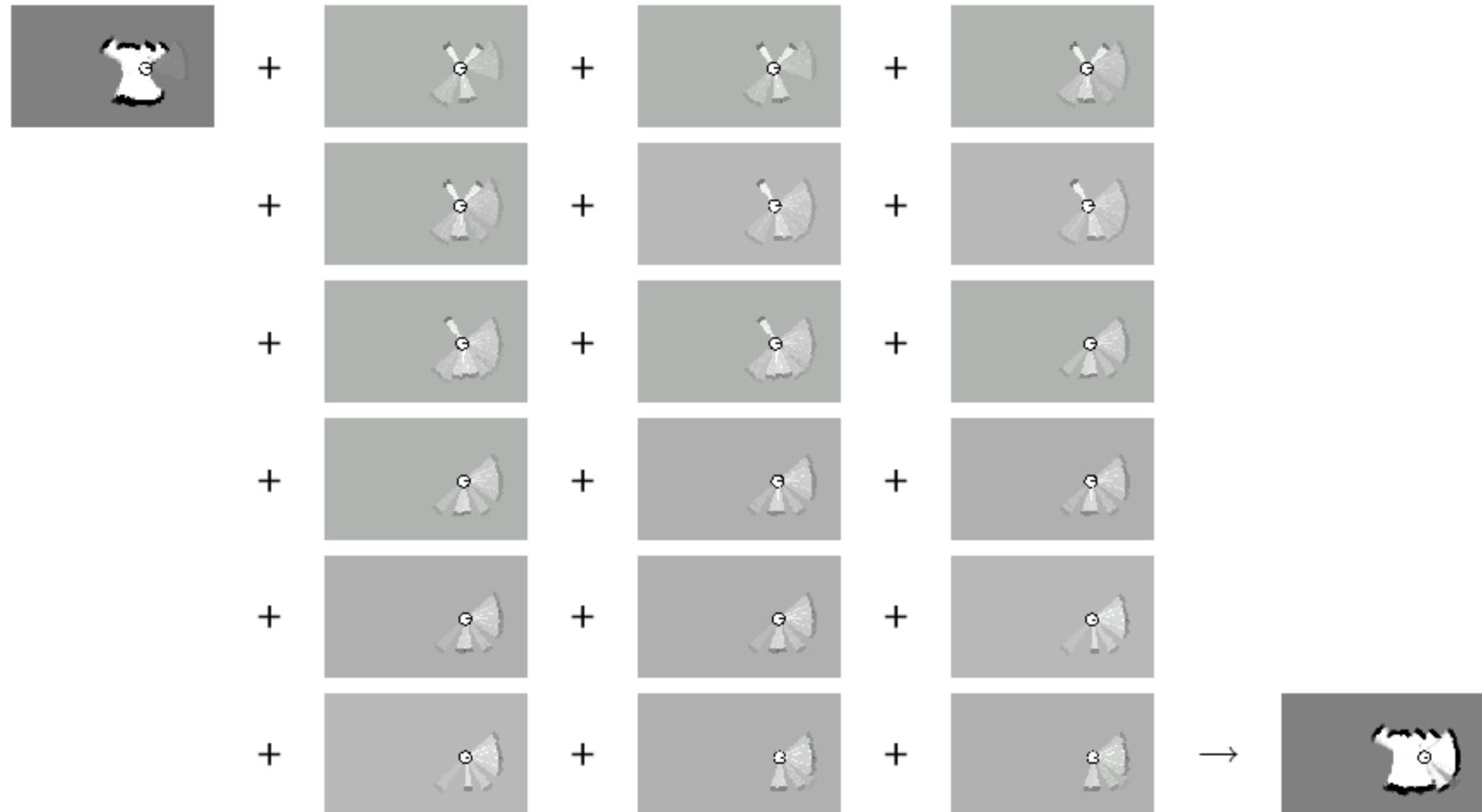
# Occupancy Value Depending on the Measured Distance



# Occupancy Value Depending on the Measured Distance



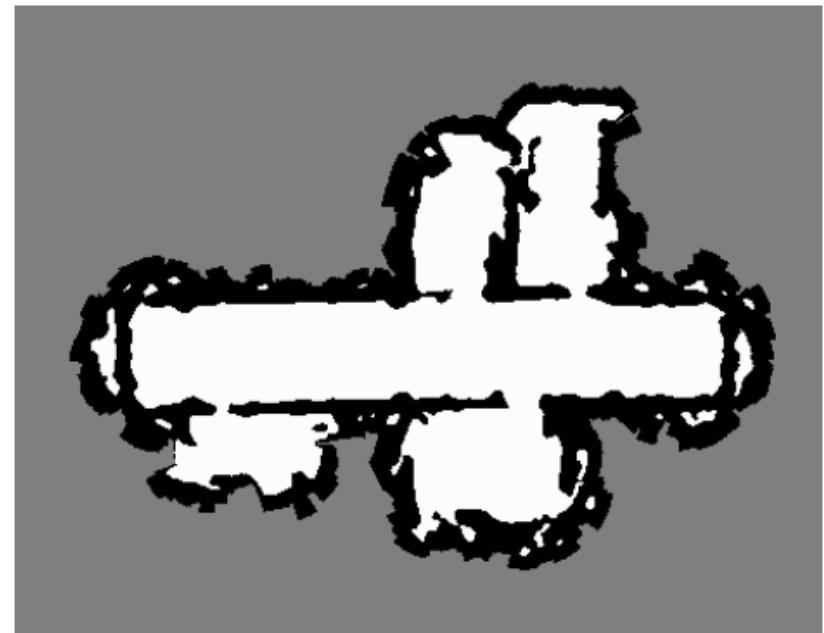
# Example: Incremental Updating of Occupancy Grids



# Resulting Map Obtained with 24 Sonar Range Sensors

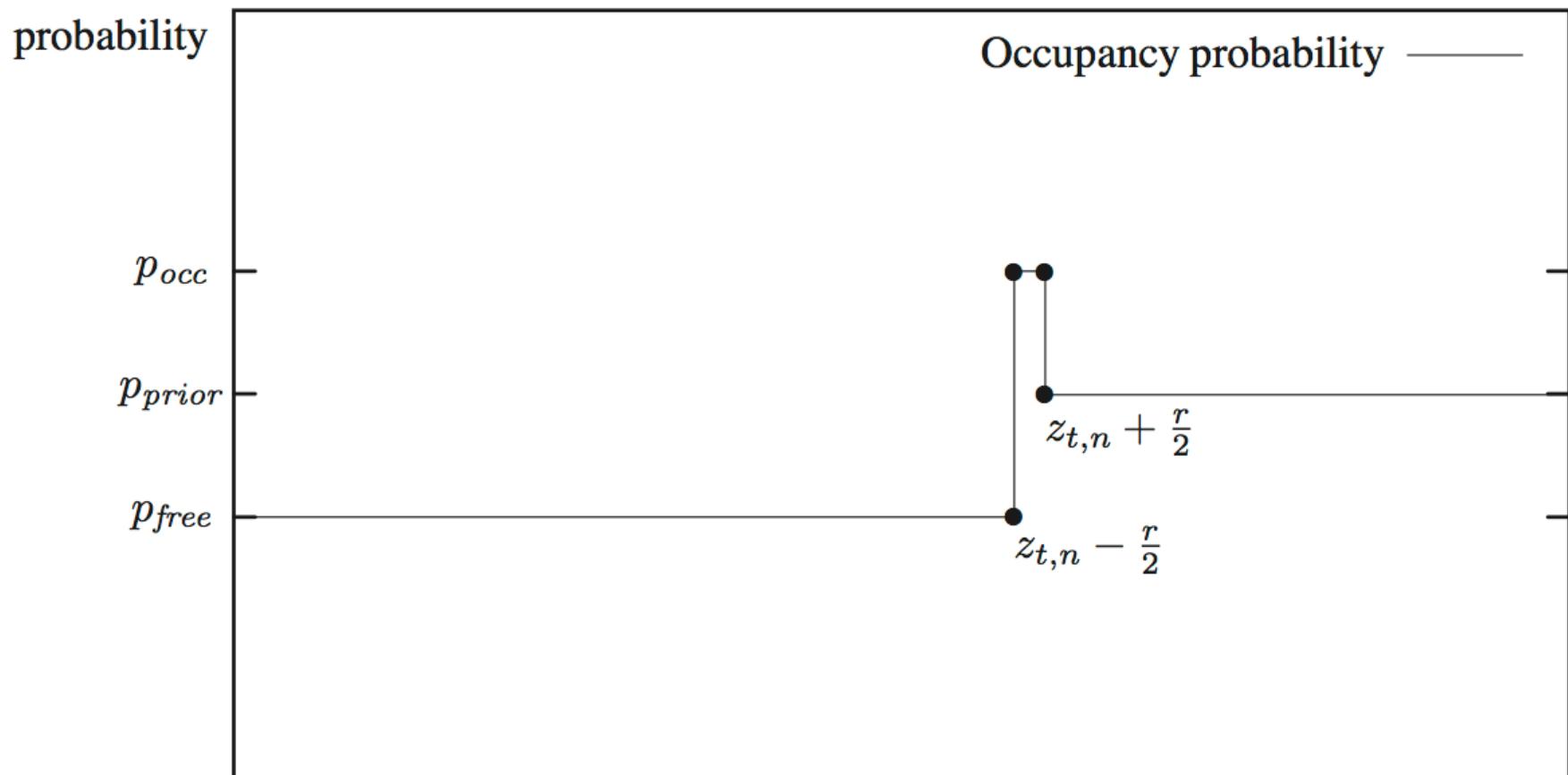


# Resulting Occupancy and Maximum Likelihood Map



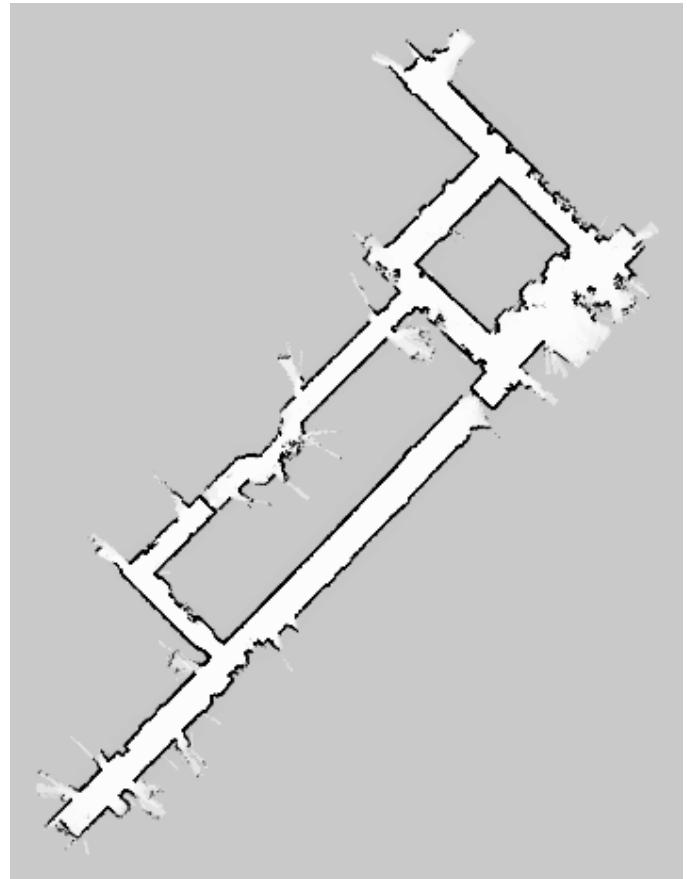
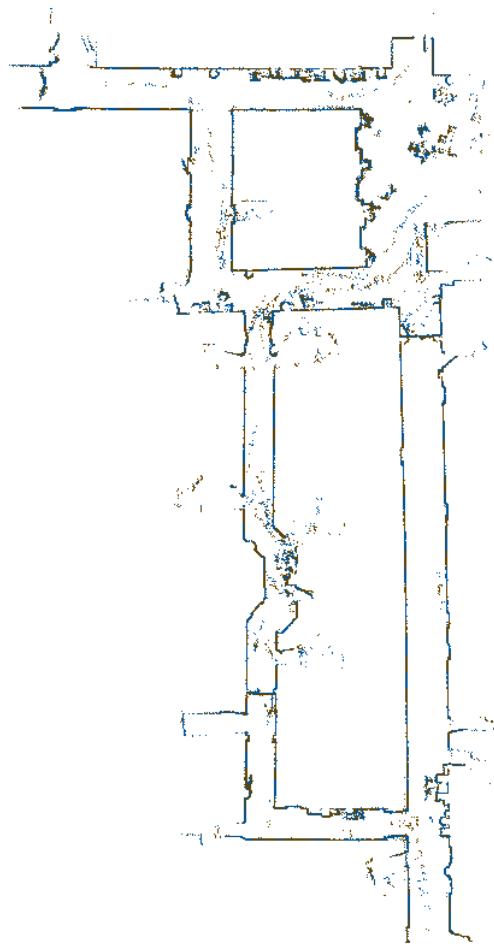
The maximum likelihood map is obtained by rounding the probability for each cell to 0 or 1

# Inverse Sensor Model for Laser Range Finders

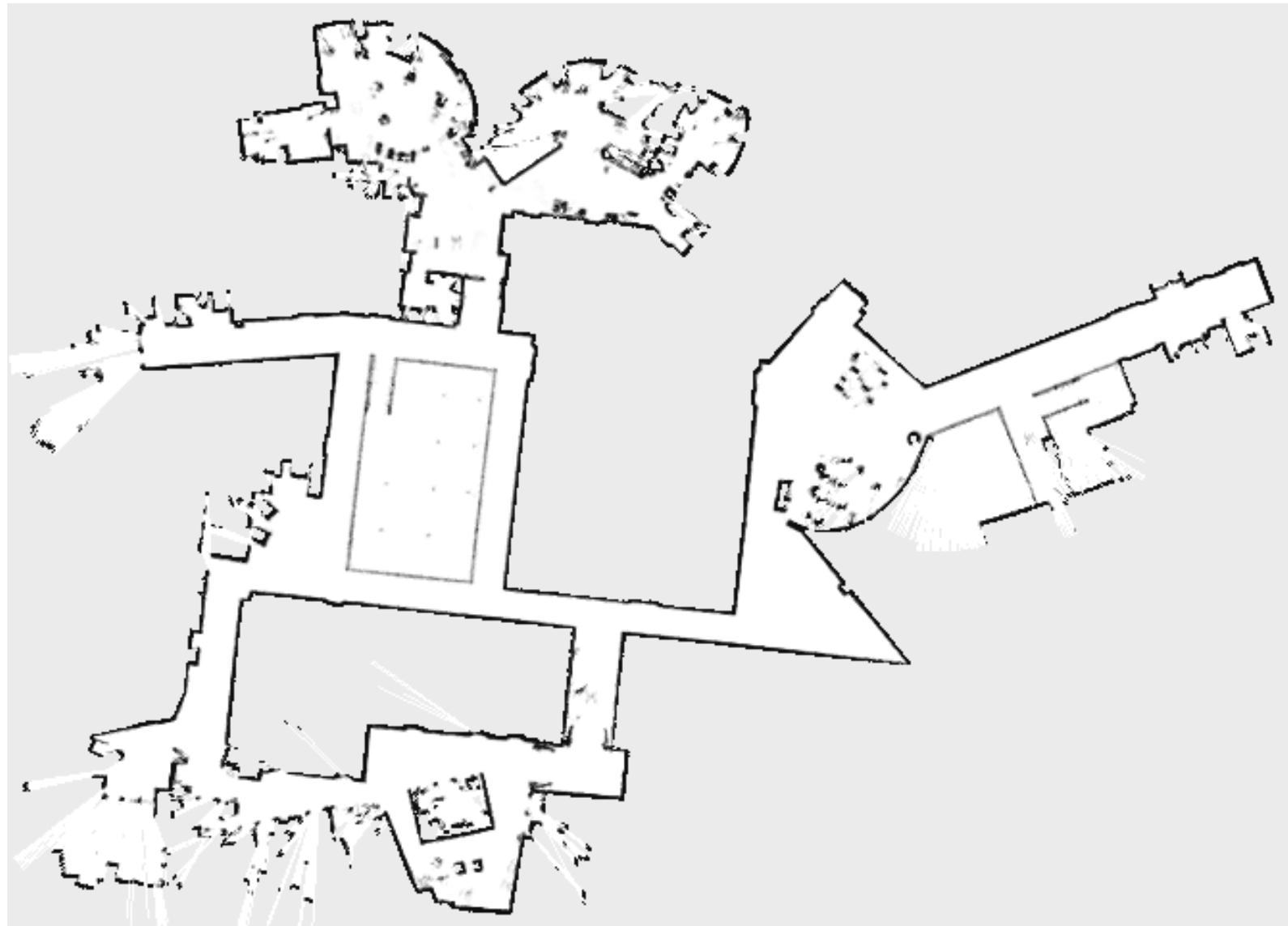


distance between sensor and cell under consideration

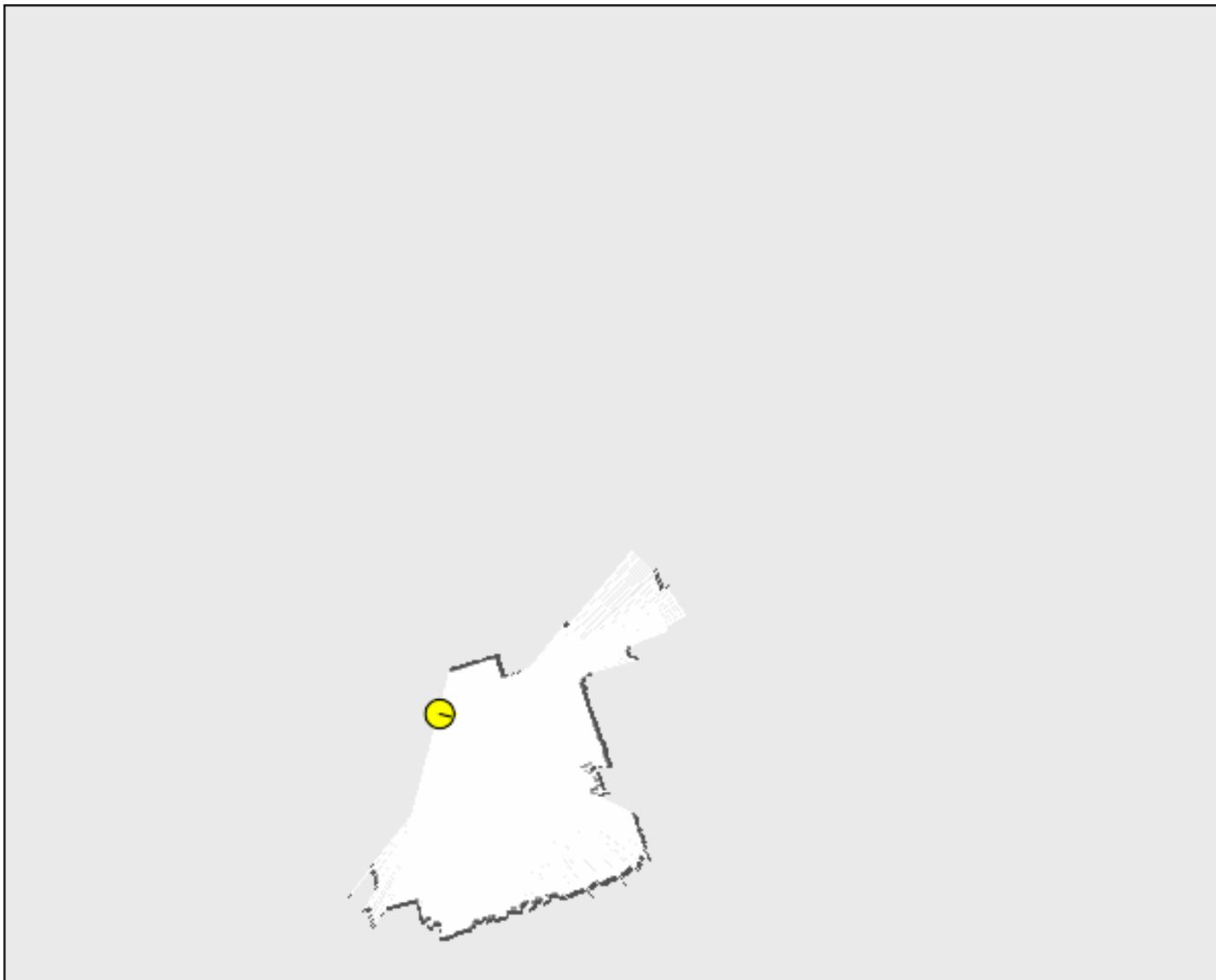
# Occupancy Grids From Laser Scans to Maps



# Example: MIT CSAIL 3<sup>rd</sup> Floor



# Uni Freiburg Building 106



# Summary: Occupancy Grid Maps

- Occupancy grid maps discretize the space into independent cells
- Each cell is a binary random variable estimating if the cell is occupied
- Static state binary Bayes filter per cell
- Mapping with known poses is easy
- Log odds model is fast to compute
- No need for predefined features

# Alternative: Counting Model / Reflection Probability Maps

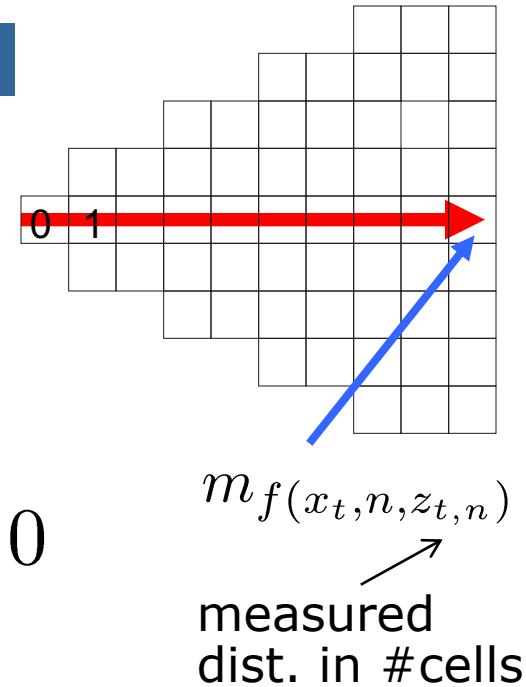
- For every cell count
  - $\text{hits}(x,y)$ : number of cases where a beam ended at  $\langle x,y \rangle$
  - $\text{misses}(x,y)$ : number of cases where a beam passed through  $\langle x,y \rangle$

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

- Value of interest:  $P(\text{reflects}(x,y))$

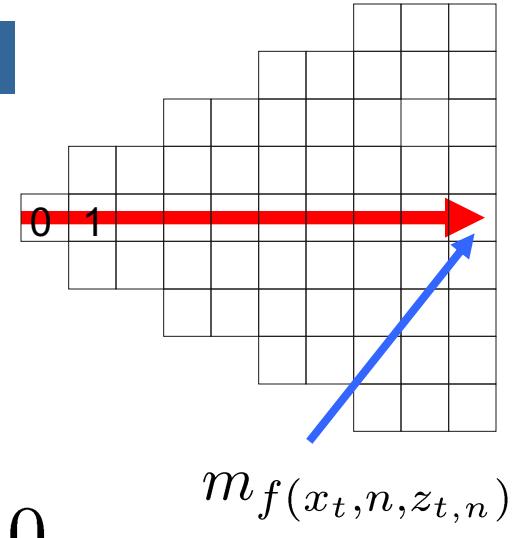
# The Measurement Model

- 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:  $\zeta_{t,n} = 0$

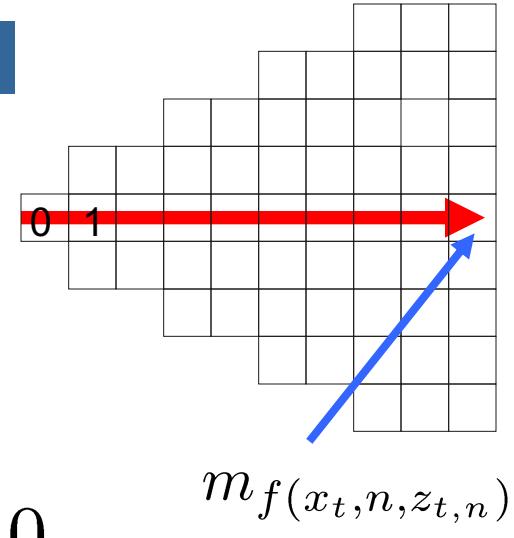


max range: “first  $z_{t,n}-1$  cells covered by the beam must be free”

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

# 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”

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

otherwise: “last cell reflected beam, all others free”

# Computing the Most Likely Map

- Compute values for  $m$  that maximize
$$m^* = \operatorname{argmax}_m P(m|z_1, \dots, z_t, x_1, \dots, x_t)$$
- Assuming a uniform prior probability for  $P(m)$ , this is equivalent to maximizing:

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

# Computing the Most Likely Map

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

# Computing the Most Likely Map

$$\begin{aligned} m^* &= \operatorname{argmax}_m \sum_{j=1}^J \sum_{t=1}^T \sum_{n=1}^N \left( I(f(x_t, n, z_{t,n}) = j) \cdot (1 - \zeta_{t,n}) \cdot \ln m_j \right. \\ &\quad \left. + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \right) \end{aligned}$$

# Computing the Most Likely Map

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

"beam  $n$  ends in cell  $j$ "  
"beam  $n$  traversed cell  $j$ "

# Computing the Most Likely Map

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

"beam  $n$  ends in cell  $j$ "  
"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 $\alpha_j$ and $\beta_j$

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

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

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

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

If we set

we obtain

$$\frac{\partial}{\partial m_j} = \frac{\alpha_j}{m_j} - \frac{\beta_j}{1-m_j} = 0 \quad m_j = \frac{\alpha_j}{\alpha_j + \beta_j}$$

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



# Example

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

$$\frac{p(\text{occ} | z)}{p(\neg \text{occ} | z)} = \left( \frac{0.55}{0.45} \right)^{n*0.6} * \left( \frac{0.45}{0.55} \right)^{n*0.4} = \left( \frac{11}{9} \right)^{n*0.6} * \left( \frac{11}{9} \right)^{-n*0.4} = \left( \frac{11}{9} \right)^{n*0.2}$$

- Whereas the reflection map yields a value of 0.6, the occupancy grid value converges to 1 as  $n$  increases

# Summary: Reflection

- Reflection probability maps are an alternative representation
- They store in each cell the probability that a beam is reflected by this cell
- Given the described sensor model, counting yields the maximum likelihood model

# Acknowledgment

- These slides have been created by Wolfram Burgard, Dieter Fox, Cyrill Stachniss and Maren Bennewitz