Cognitive Robotics

07. Mapping with Known Poses

AbdElMoniem Bayoumi, PhD

Spring 2022

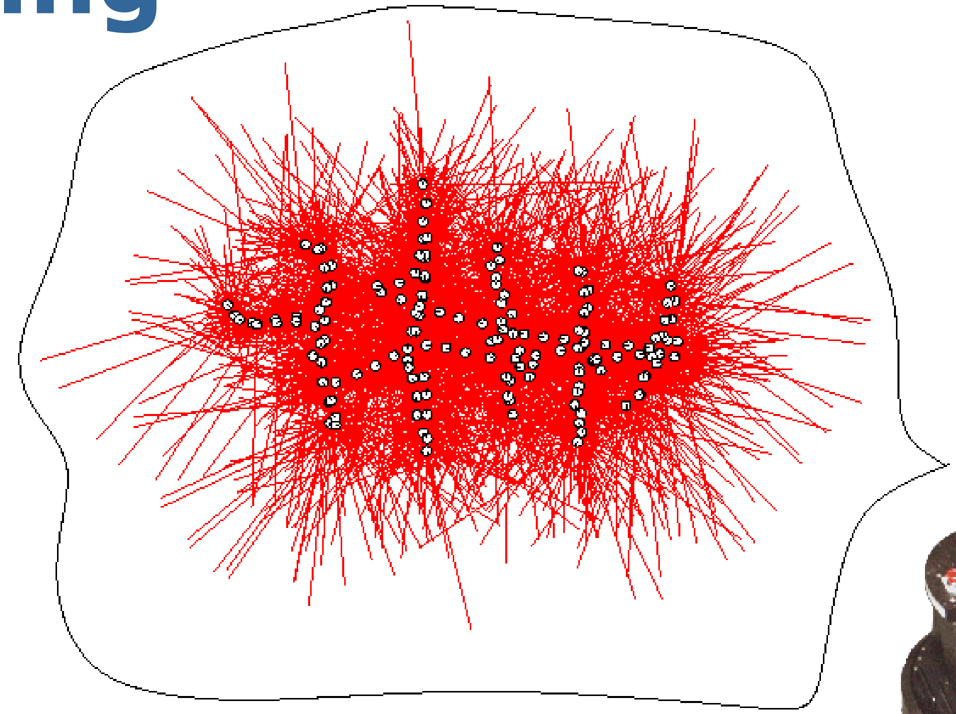
Acknowledgment

- 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

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

- to calculate the most likely map

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

The General Problem of Mapping

- Formally, mapping involves, given the sensor data

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

- to calculate the most likely map

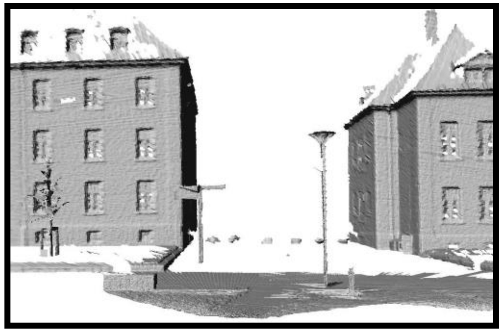
$$m^\star = \operatorname{argmax}_m P(m|d)$$

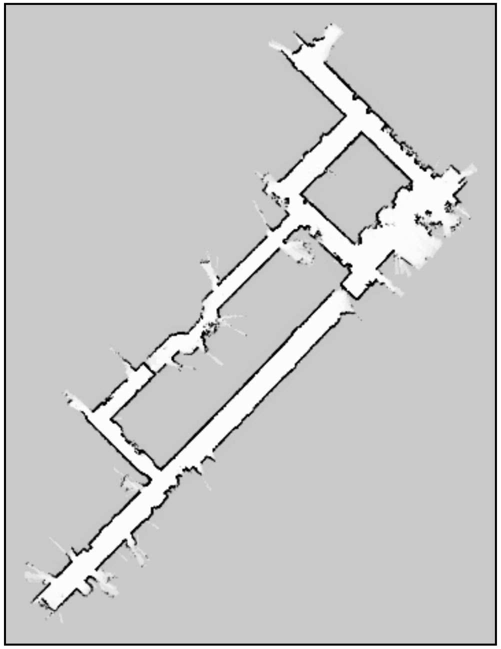
- Today we describe **how to calculate a map given the robot's pose**

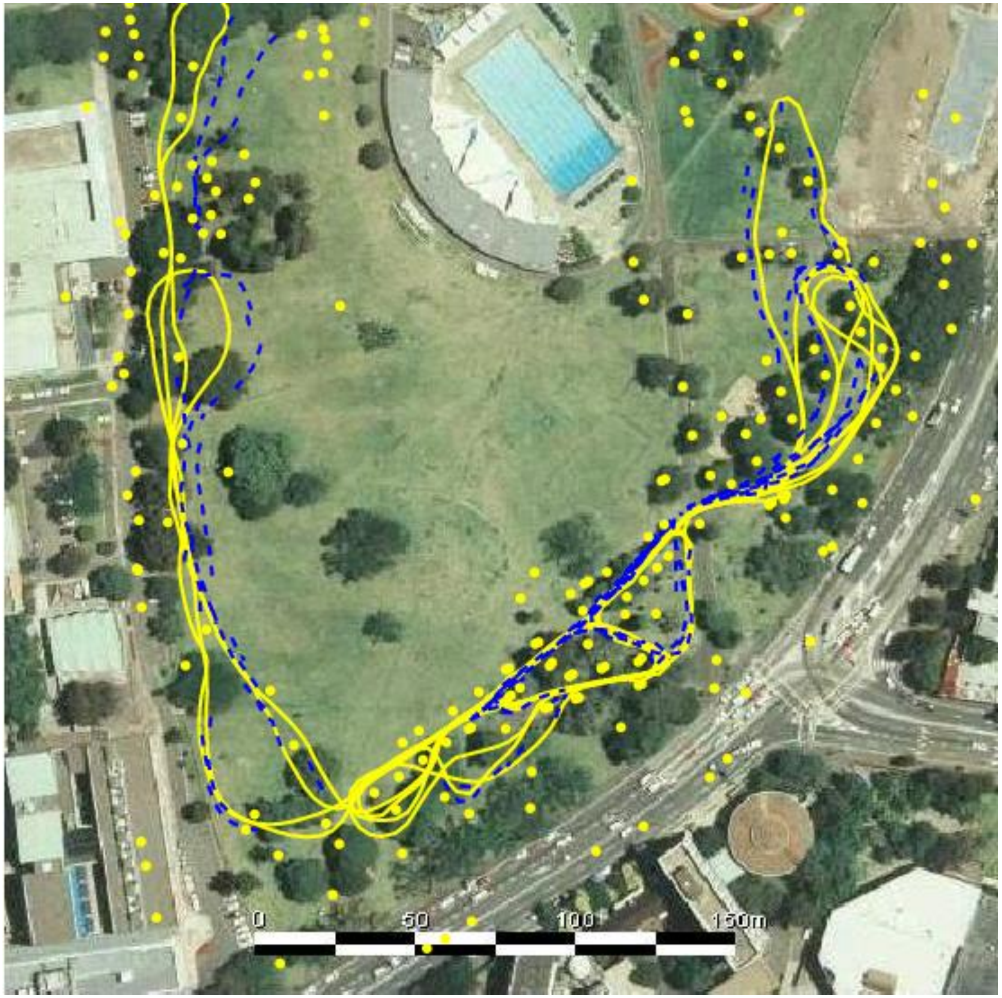
Mapping as a Chicken and Egg Problem

- 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





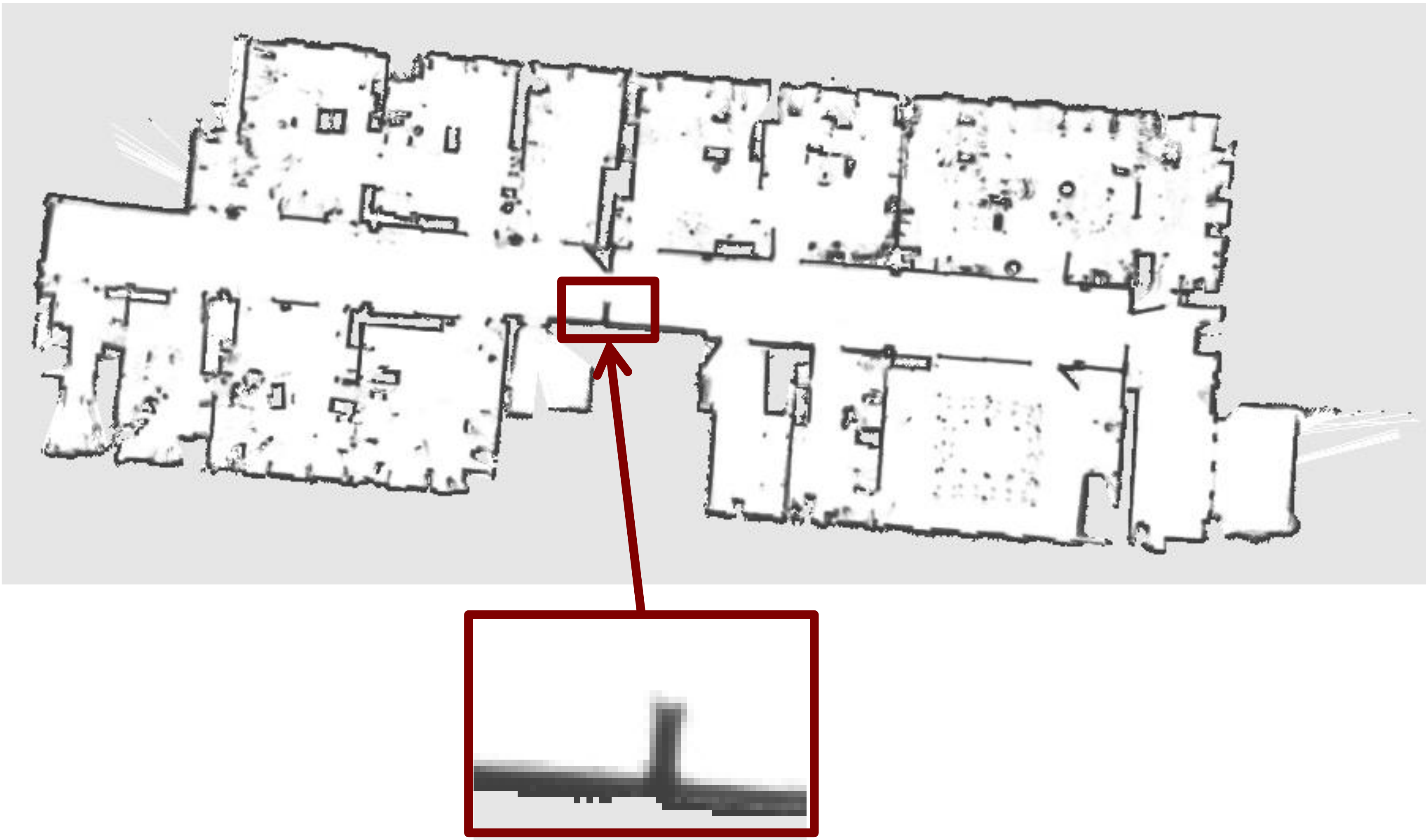


Courtesy by E. Nebot

Grid Maps

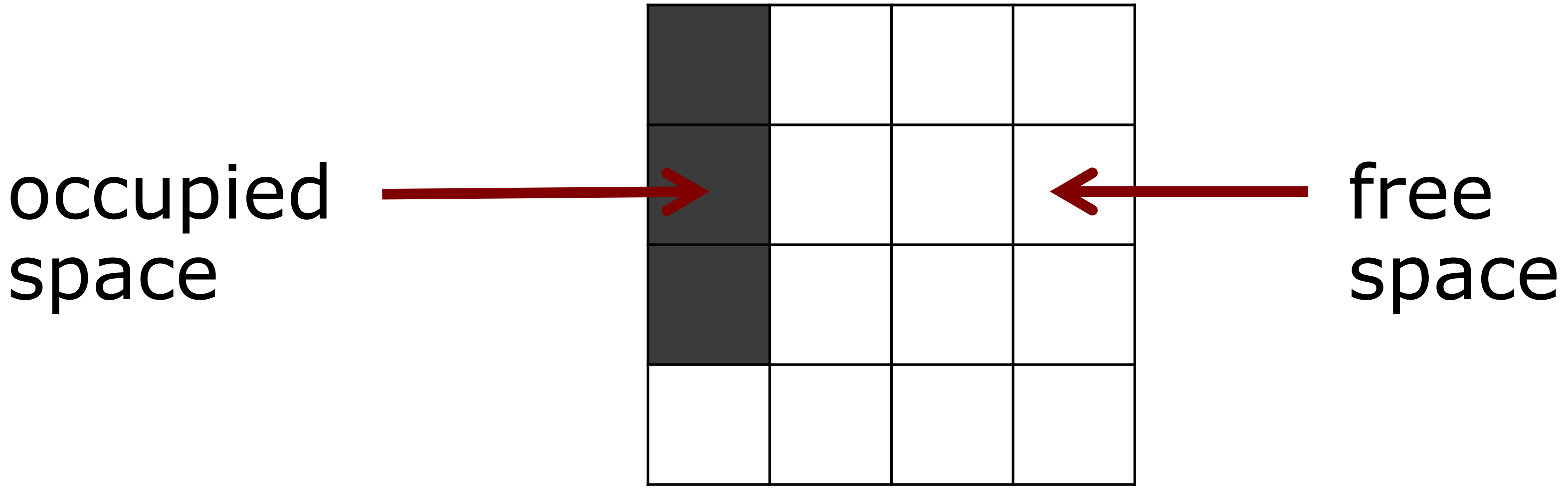
- Discretize the world into cells
- Grid structure is rigid
- Each cell is assumed to be occupied or free space
- Non-parametric model
- Large maps require substantial memory resources
- Do not rely on a feature detector

Example



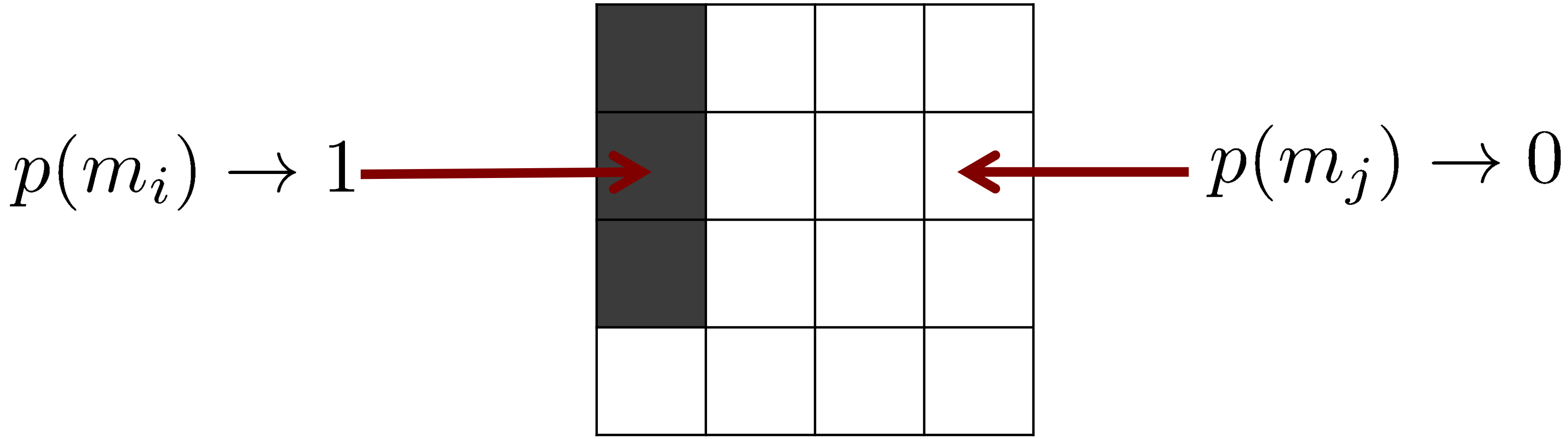
Assumption 1

- 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

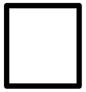
- Each cell is a **binary random variable** that models the occupancy
- Cell is occupied: $p(m_i) = 1$
- Cell is not occupied: $p(m_i) = 0$
- No knowledge: $p(m_i) = 0.5$

Occupancy Probability Example

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

$$P(M_i = occ) = p(m_i) = 1$$

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



$$P(M_i = occ) = p(m_i) = 0$$

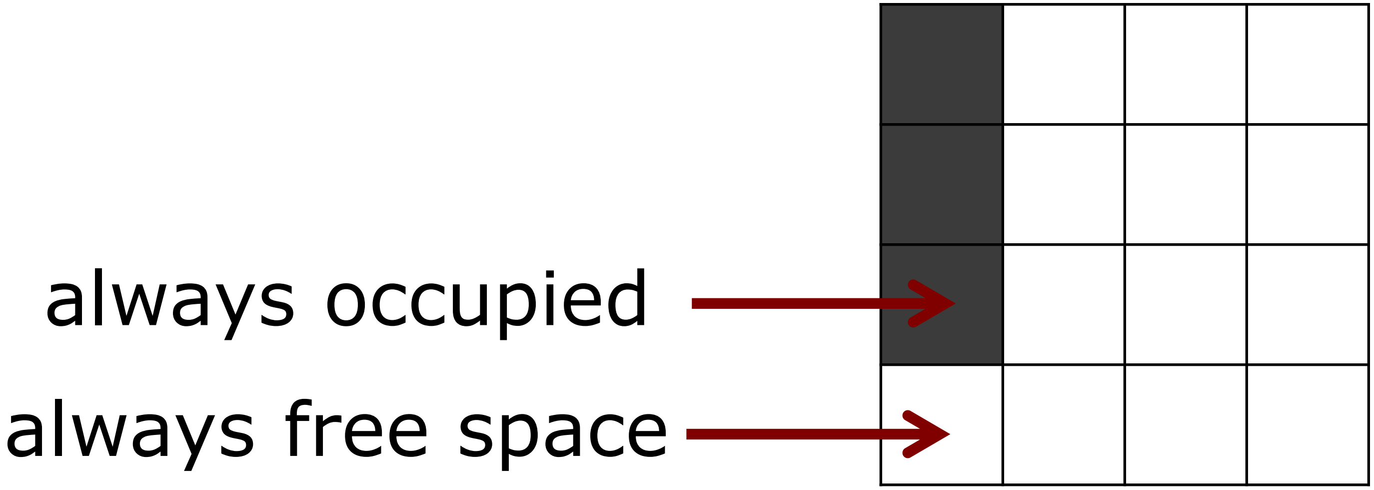
$$P(M_i = free) = p(\neg m_i) = 1 - p(m_i) = 1$$

$$P(M_i = occ) = p(m_i) = 0.75$$

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

Assumption 2

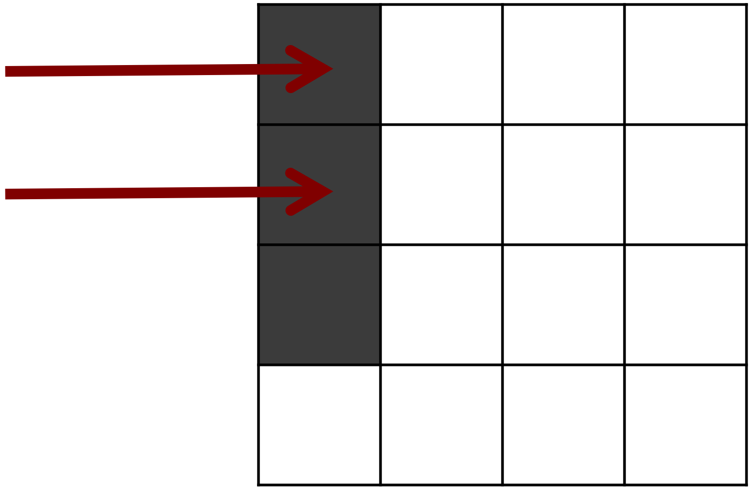
- The world is **static** (most mapping systems make this assumption)



Assumption 3

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

no dependency
between the cells



Joint Distribution

$$p(m) \quad = \quad p(m_1, m_2, \ldots, m_N)$$

map

cell 1

cell N

“and”

Representation

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

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

Example A

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

$$p\left(M = \begin{matrix}\blacksquare & \blacksquare \\ \blacksquare & \blacksquare\end{matrix}\right) = p(M_1 = \blacksquare)p(M_2 = \blacksquare)$$
$$p(M_3 = \blacksquare)p(M_4 = \blacksquare)$$

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

Example B

$$
\begin{aligned}
p\left(M = \begin{matrix}\blacksquare & \square \\ \blacksquare & \square\end{matrix}\right) &= p(M_1 = \blacksquare)\,p(M_2 = \square) \\
&\quad\; p(M_3 = \blacksquare)\,p(M_4 = \square) \\
&= p(M_1 = \blacksquare)(1 - p(M_2 = \blacksquare)) \\
&\quad\; (1 - p(M_3 = \blacksquare))\,p(M_4 = \blacksquare)
\end{aligned}
$$

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

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

binary random variable

$\Rightarrow$ 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

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

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

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

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

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

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

Static State Binary Bayes Filter

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

Static State Binary Bayes Filter

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

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

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

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

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:

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

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:

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

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

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

From Ratio to Probability

- We can easily turn the ration into the probability

$$
\begin{aligned}
\frac{p(x)}{1-p(x)} &= Y \\
p(x) &= Y - Y\,p(x) \\
p(x)\,(1+Y) &= Y \\
p(x) &= \frac{Y}{1+Y} \\
p(x) &= \frac{1}{1+\frac{1}{Y}}
\end{aligned}
$$

From Ratio to Probability

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

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

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

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

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

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

Log Odds Notation

- Log odds ratio is defined as

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

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

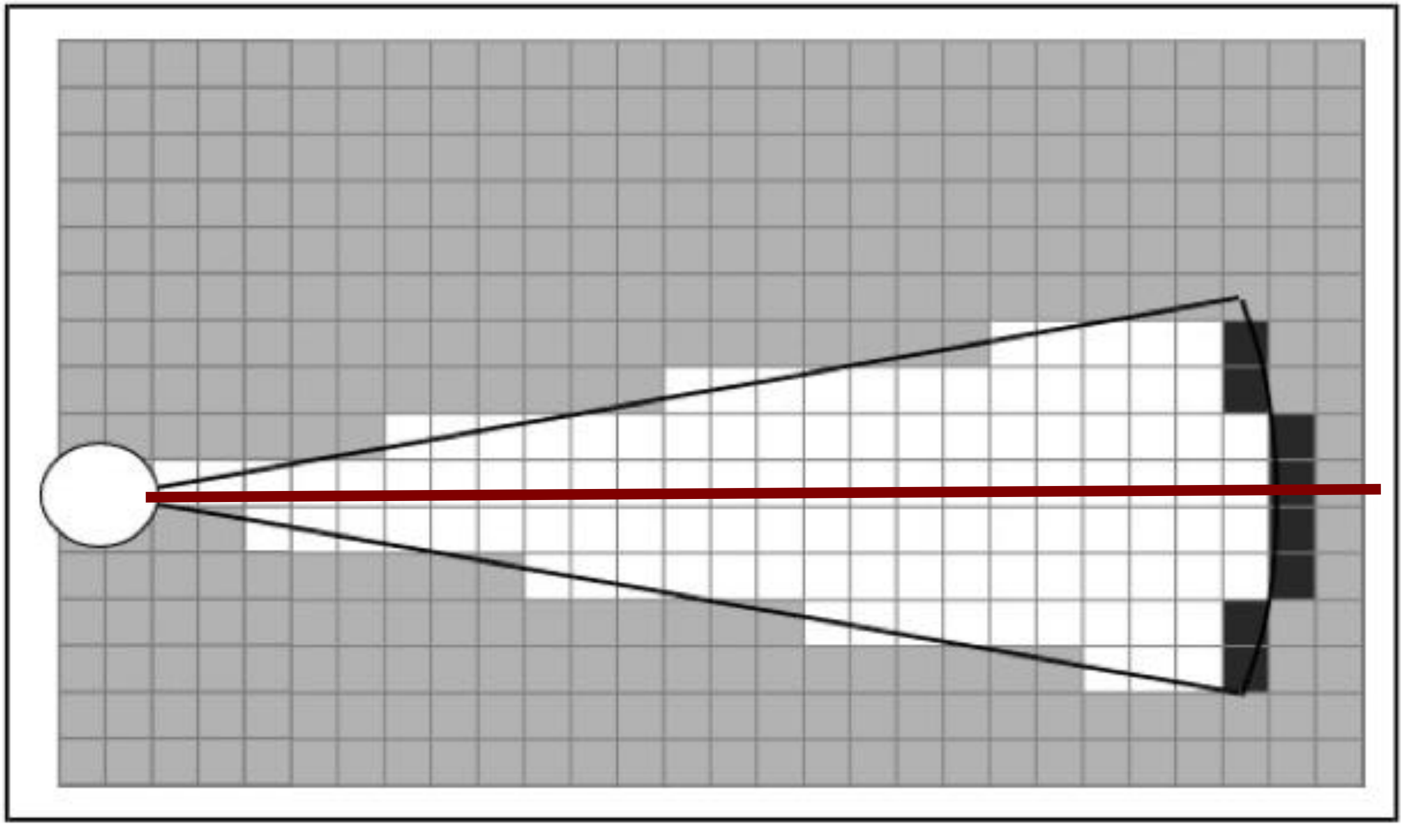
for all cells m_i *do*
    if m_i *in perceptual field of* z_t *then*
        $l_{t,i} = l_{t-1,i} + \text{inv\_sensor\_model}(m_i, x_t, z_t) - l_0$
    else
        $l_{t,i} = l_{t-1,i}$
    endif
endfor
return $\{l_{t,i}\}$

highly efficient, 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

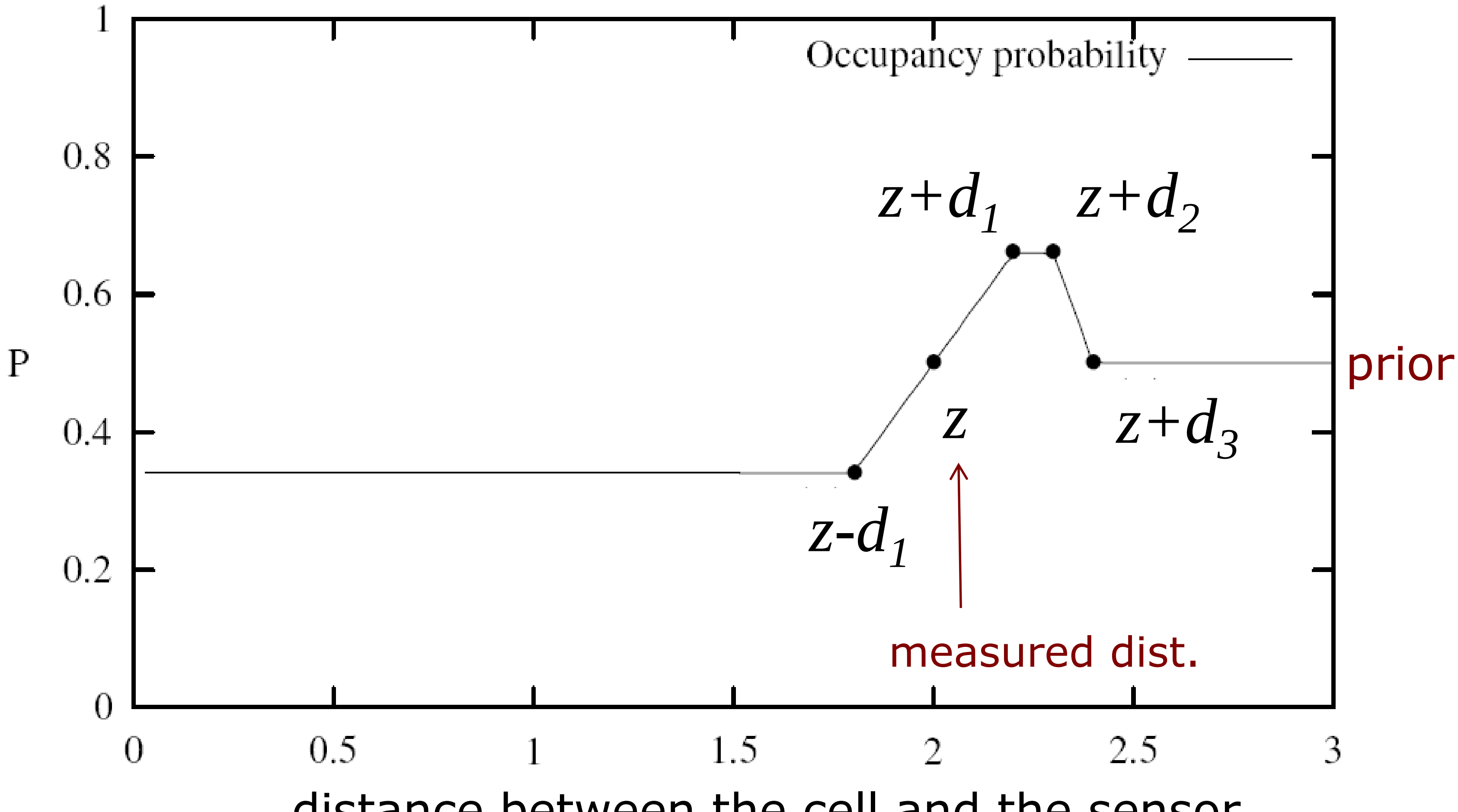
Inverse Sensor Model for Sonar Range Sensors



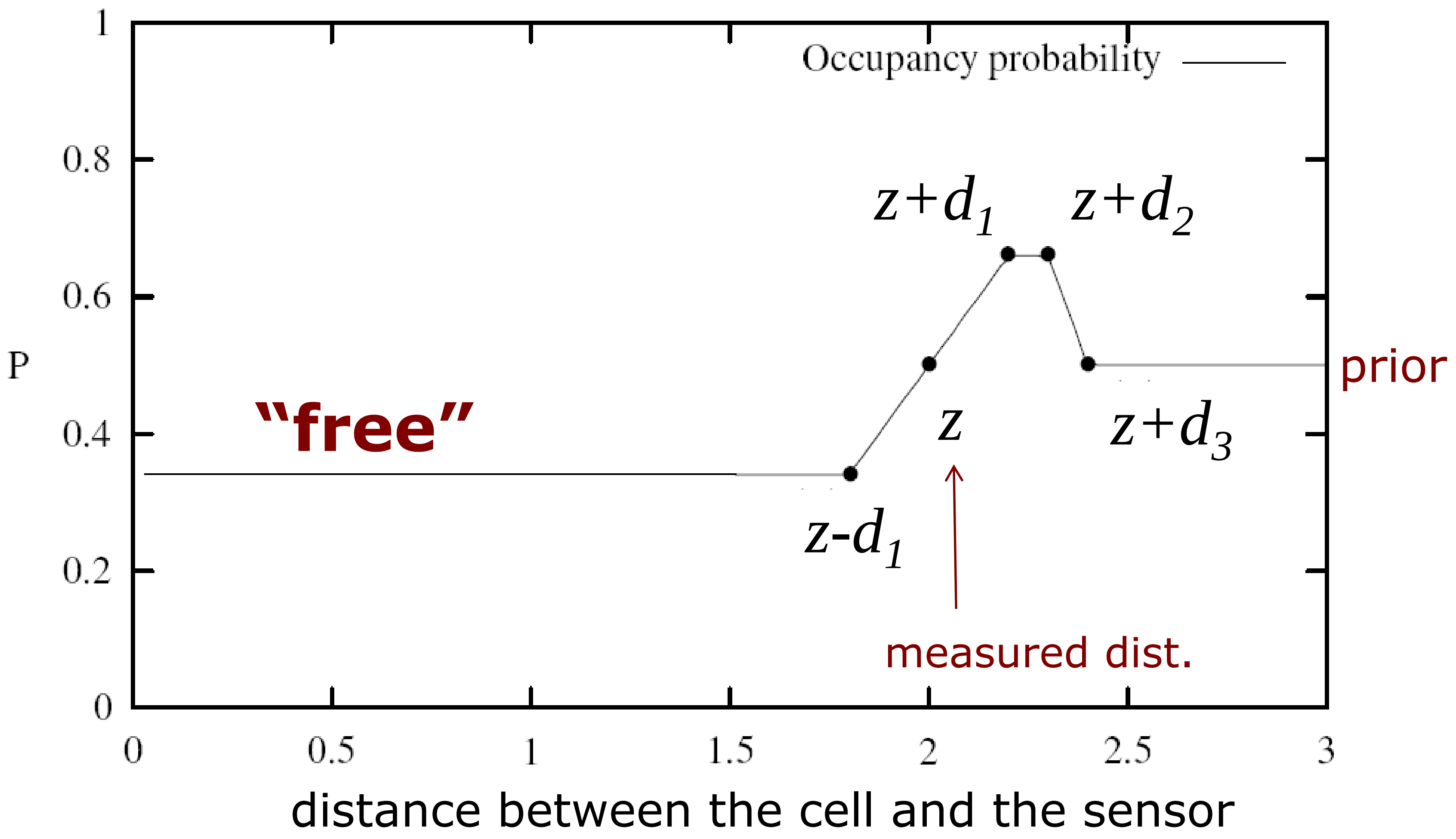
In the following, consider the cells along the optical axis (red line)

Courtesy: Thrun, Burgard, Fox

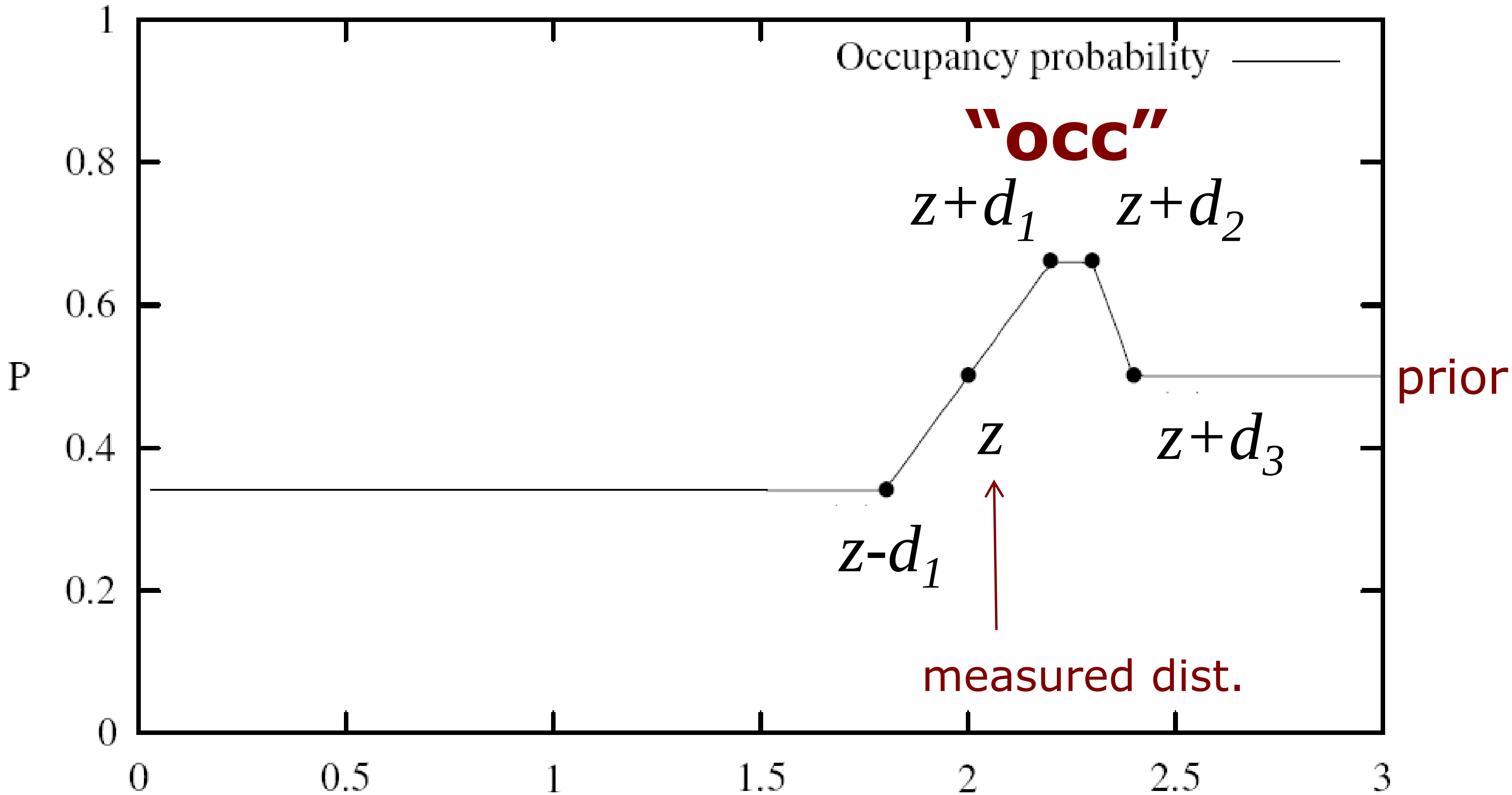
Occupancy Value Depending on the Measured Distance



Occupancy Value Depending on the Measured Distance

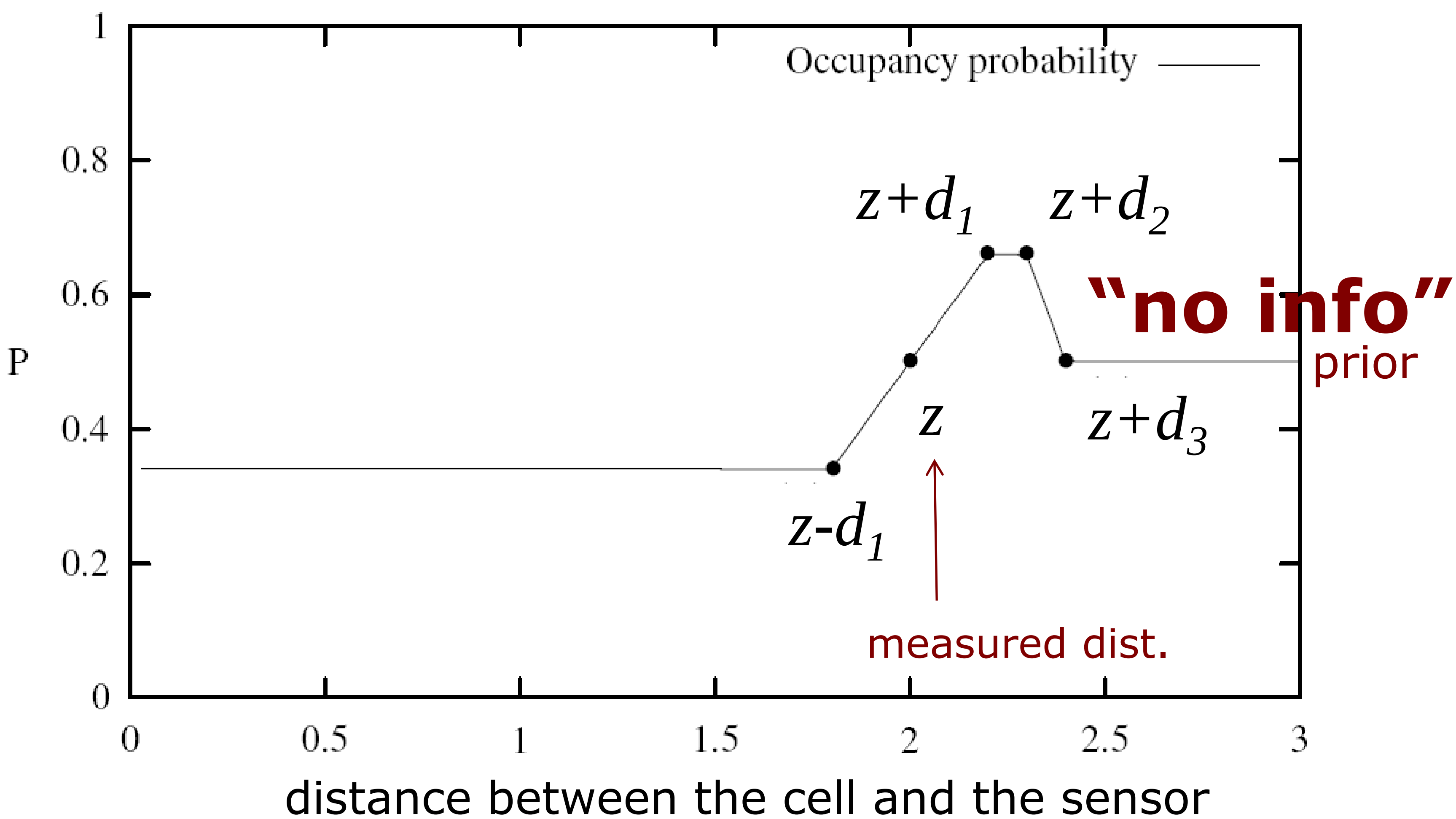


Occupancy Value Depending on the Measured Distance

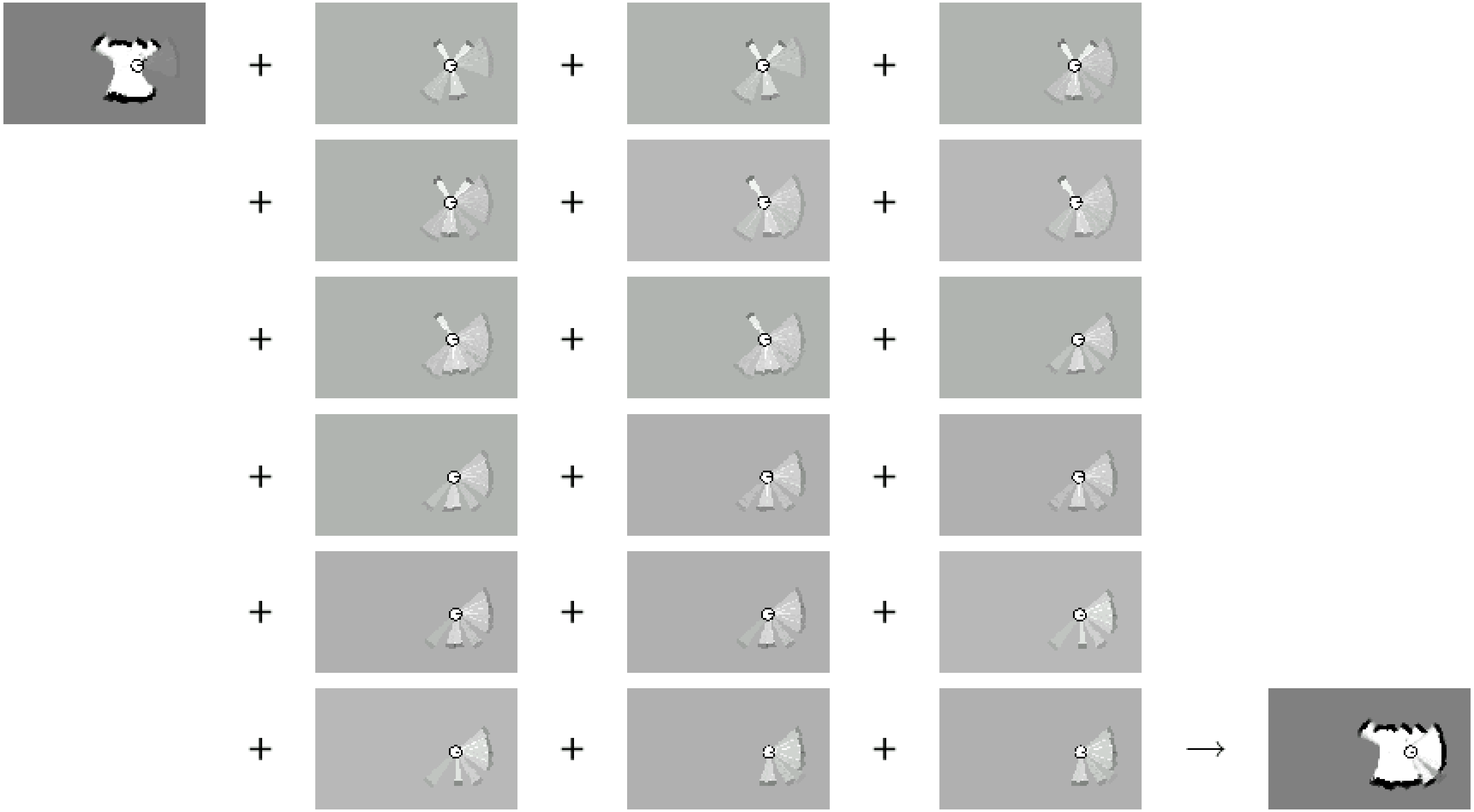


distance between the cell and the sensor

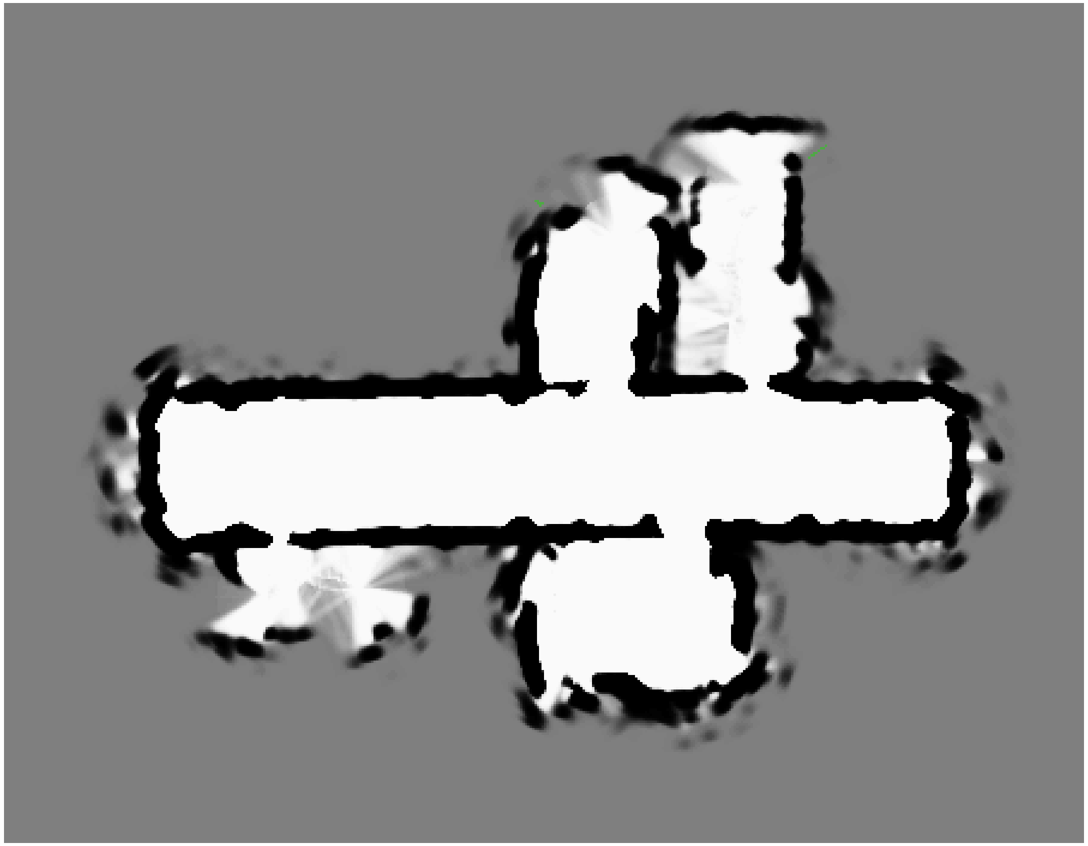
Occupancy Value Depending on the Measured Distance



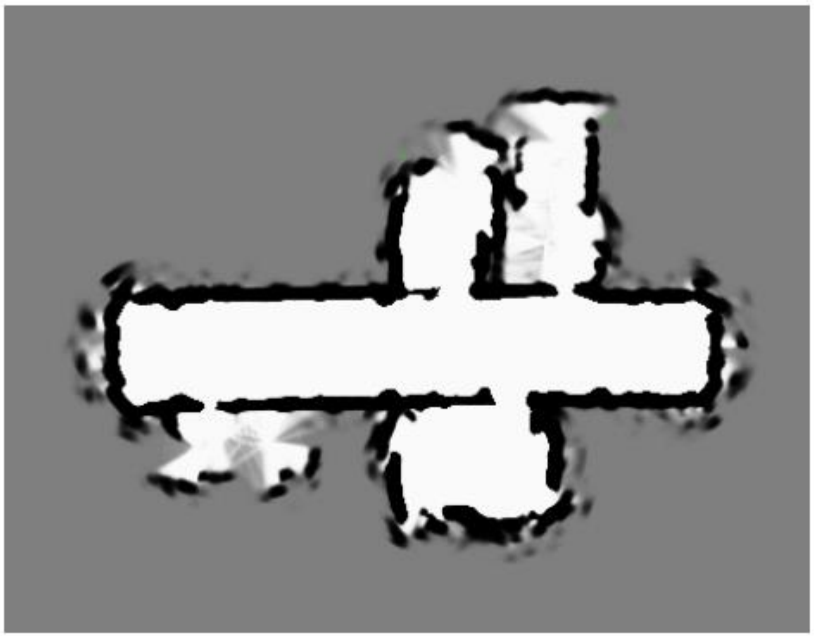
Example: Incremental Updating of Occupancy Grids

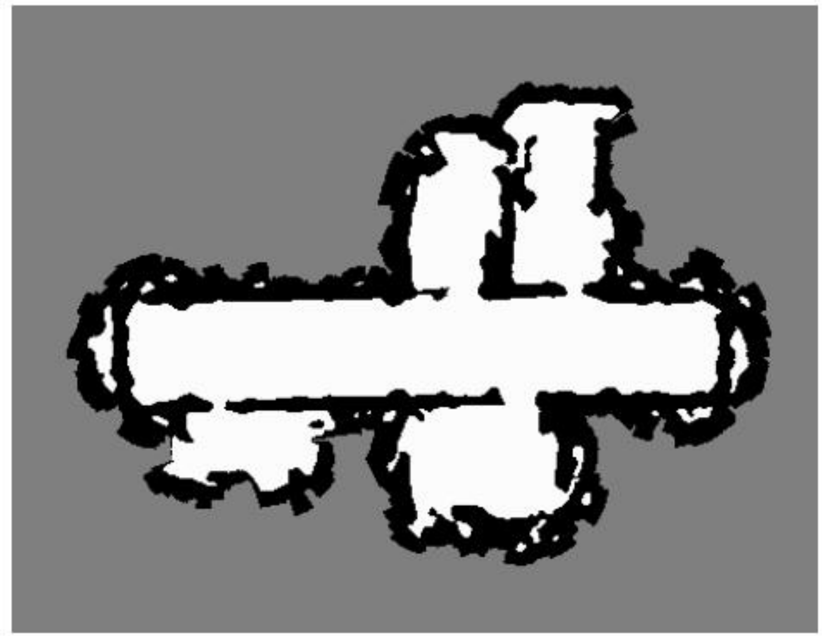


Resulting Map Obtained with 24 Sonar Range Sensors



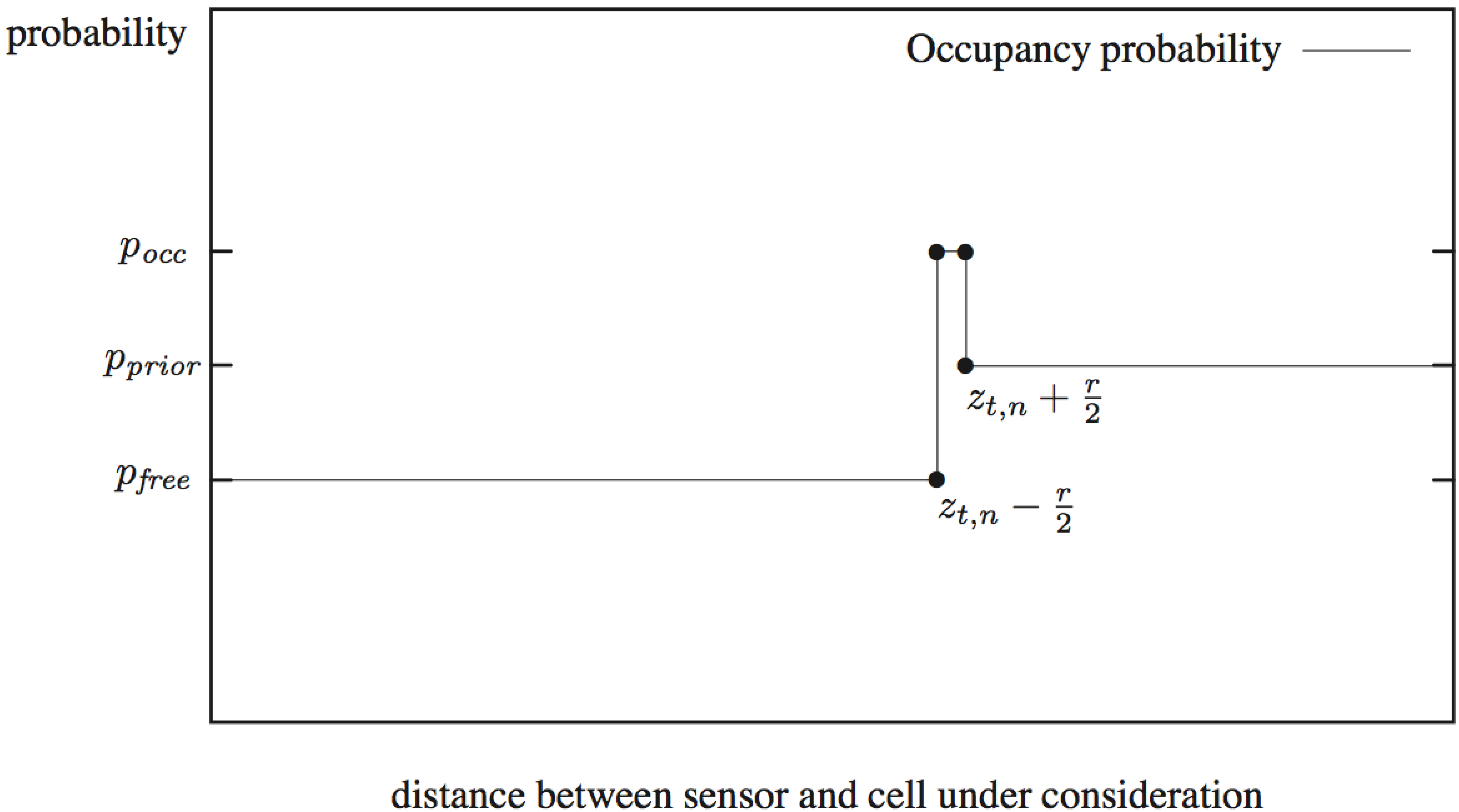
Resulting Occupancy and Maximum Likelihood Map



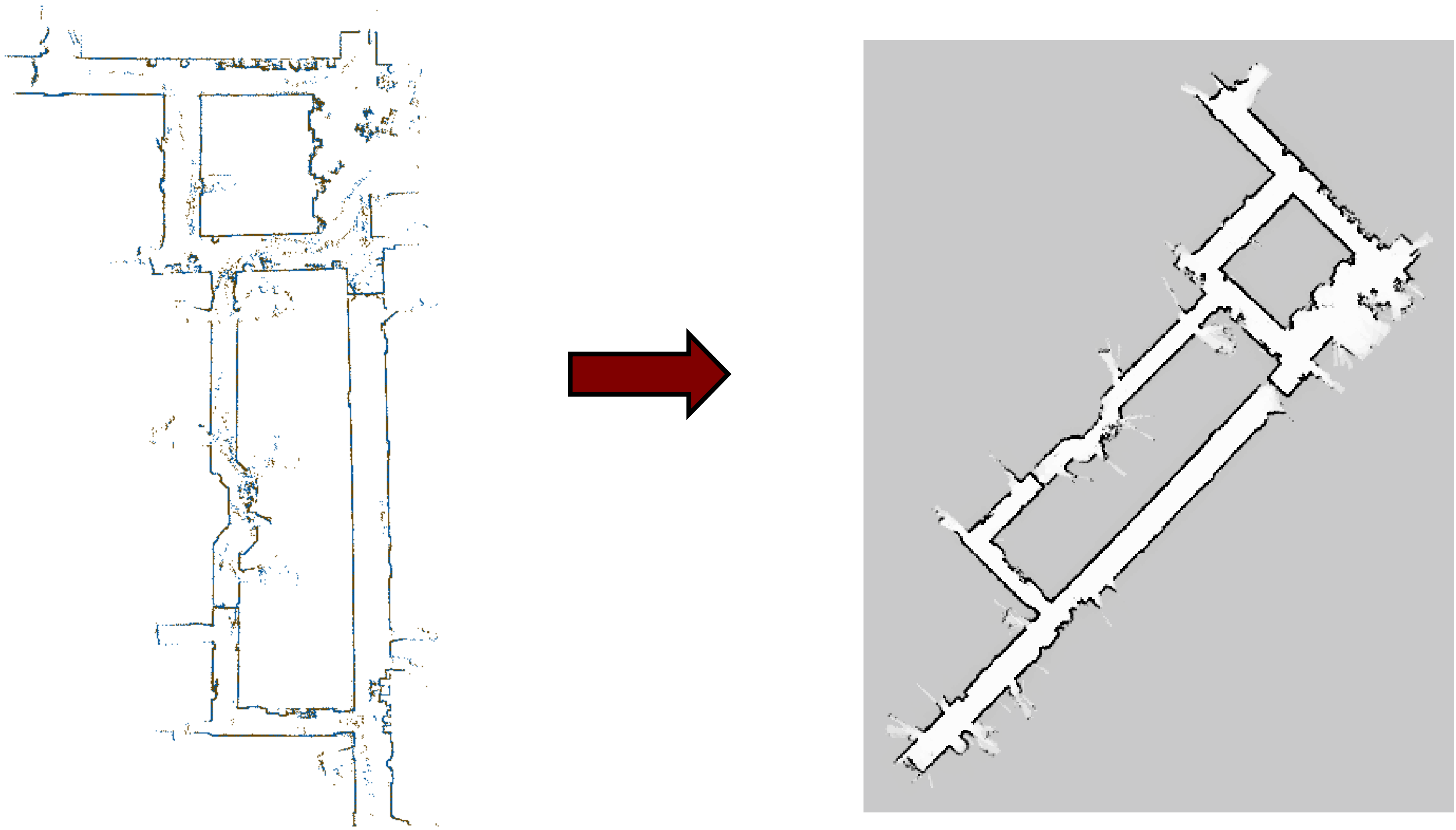


The maximum likelihood map is obtained by rounding the probability for each cell to 0 or 1

Inverse Sensor Model for Laser Range Finders

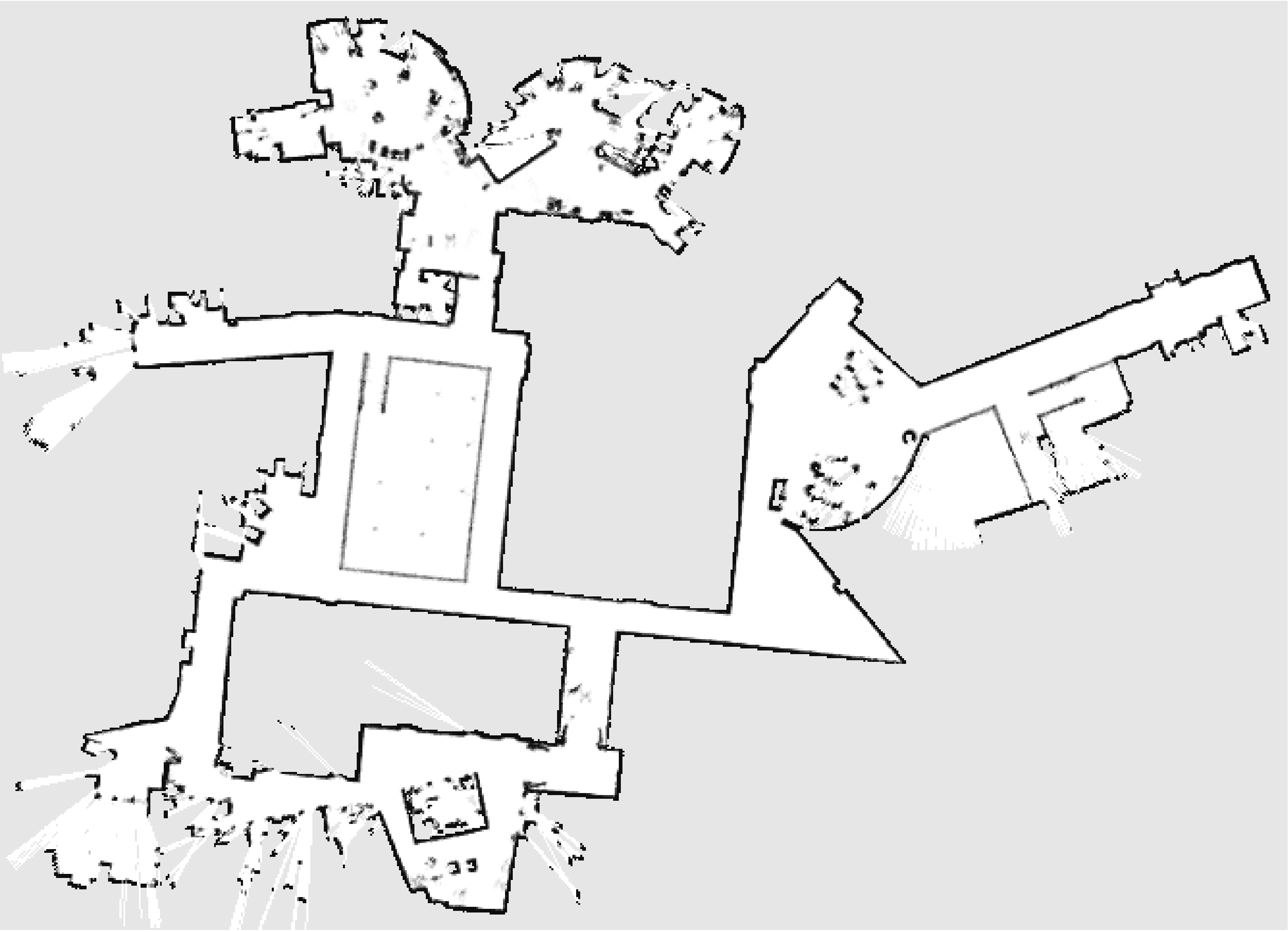


Occupancy Grids
From Laser Scans to Maps

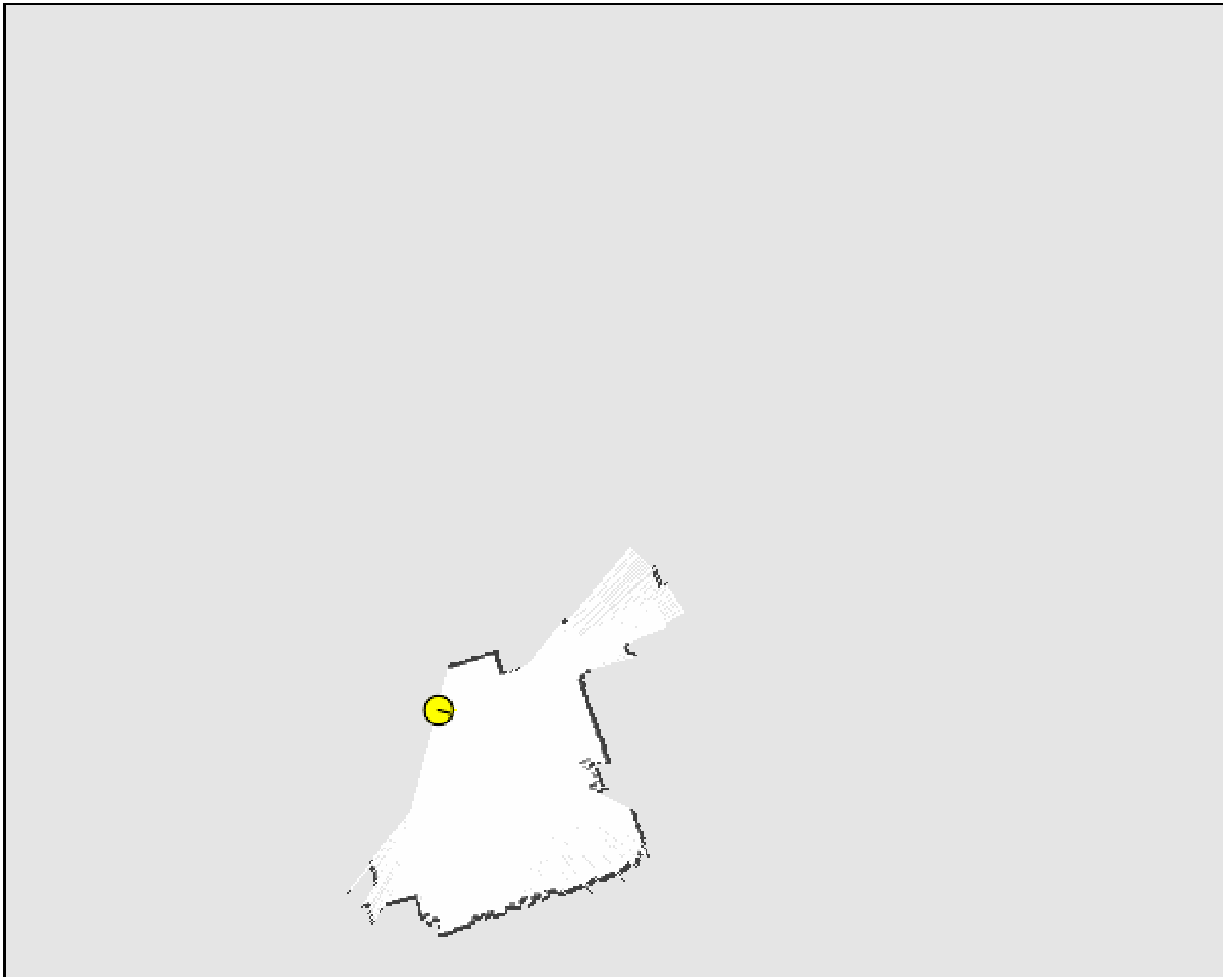


Courtesy: D. Hähnel

Example: MIT CSAIL 3rd 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

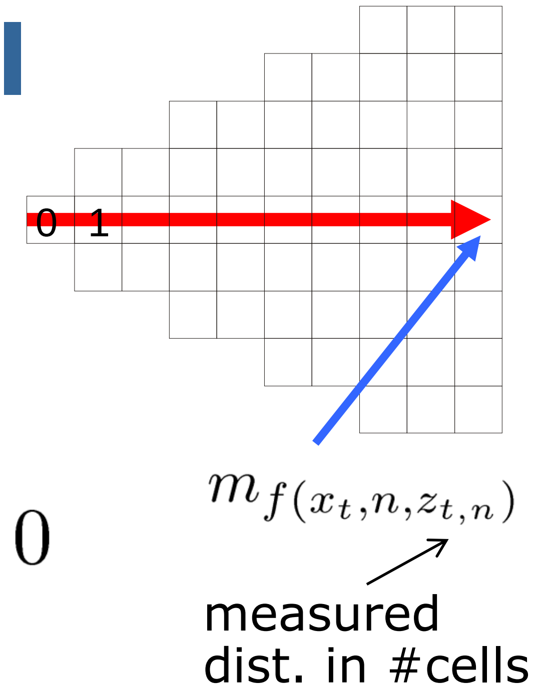
- For every cell count
  - hits(x,y): number of cases where a beam ended at <x,y>
  - misses(x,y): number of cases where a beam passed through <x,y>

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

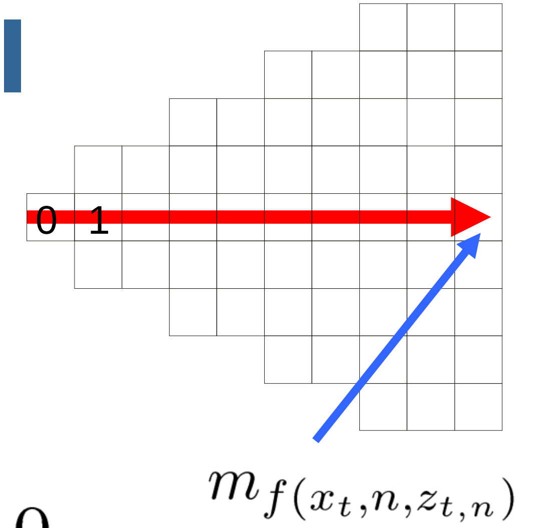
- Value of interest: P(reflects(x,y))

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

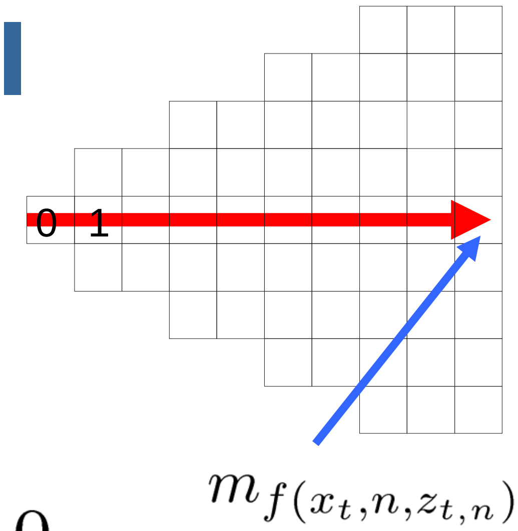


- Pose at time *t*: x_t
- Beam *n* of scan at time *t*: $z_{t,n}$
- Maximum range reading: $\zeta_{t,n} = 1$
- Beam reflected by an object: $\zeta_{t,n} = 0$

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

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

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”

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

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

Computing the Most Likely Map

- Compute values for m that maximize

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

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

$$\begin{aligned} m^\star &= \operatorname{argmax}_m P(z_1, \cdots, z_t | m, x_1, \cdots, x_t) \\ &= \operatorname{argmax}_m \prod_{t=1}^{T} P(z_t | m, x_t) \\ &= \operatorname{argmax}_m \sum_{t=1}^{T} \ln P(z_t | m, x_t) \end{aligned}$$

since z_t independent and only depend on x_t

Computing the Most Likely Map

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

Computing the Most Likely Map

"beam n ends in cell j"

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

Computing the Most Likely Map

$$
\begin{aligned}
m^\star \quad = \quad & \operatorname{argmax}_m \sum_{j=1}^{J} \sum_{t=1}^{T} \sum_{n=1}^{N} \Big(\overbrace{I(f(x_t, n, z_{t,n}) = j)}^{\text{“beam } n \text{ ends in cell } j\text{”}} \cdot (1 - \zeta_{t,n}) \cdot \ln m_j \\
& + \sum_{k=0}^{z_{t,n}-1} \underbrace{I(f(x_t, n, k) = j)}_{\text{“beam } n \text{ traversed cell } j\text{”}} \cdot \ln(1 - m_j) \Big)
\end{aligned}
$$

Computing the Most Likely Map

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

$$+ \sum_{k=0}^{z_{t,n}-1} \underbrace{I(f(x_t, n, k) = j)}_{\text{“beam } n \text{ traversed cell } j\text{”}} \cdot \ln(1 - m_j) \Big)$$

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

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

As the m_j's are independent 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

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(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

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

- 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