Introduction to Mobile Robotics

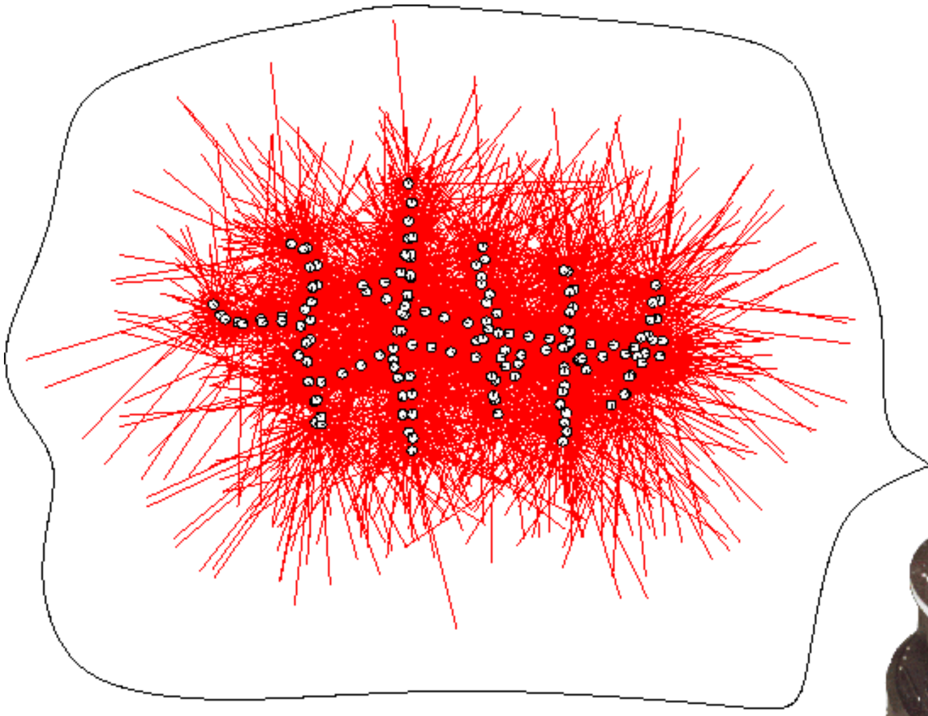
Mapping with Known Poses

Wolfram Burgard, Cyrill Stachniss, Maren Bennewitz, Diego Tipaldi, Luciano Spinello

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

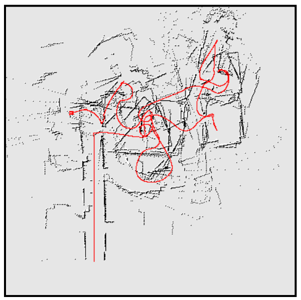
$$m^\star = \mathrm{argmax}_m P(m|d)$$

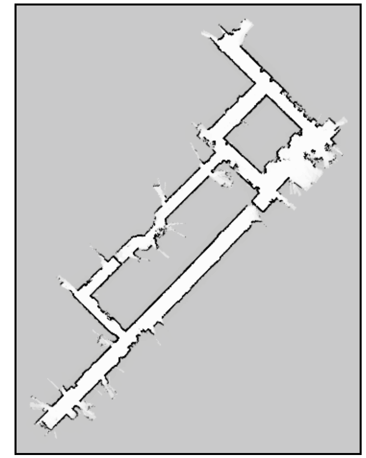
Mapping as a Chicken and Egg Problem

- 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**

Types of SLAM-Problems

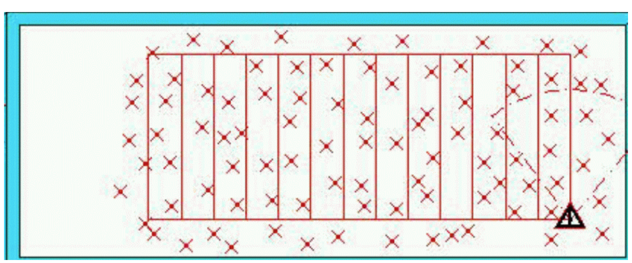
- Grid maps or scans

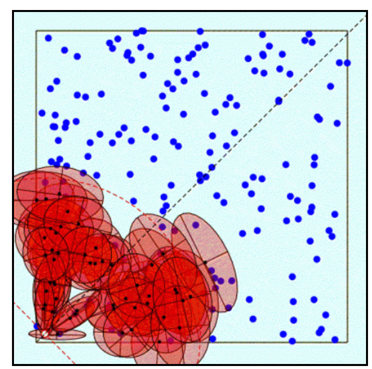


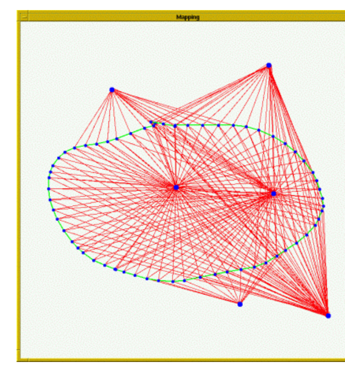


[Lu & Milios, 97; Gutmann, 98: Thrun 98; Burgard, 99; Konolige & Gutmann, 00; Thrun, 00; Arras, 99; Haehnel, 01;…]

- Landmark-based







[Leonard et al., 98; Castelanos et al., 99: Dissanayake et al., 2001; Montemerlo et al., 2002;…

Problems in Mapping

- Sensor interpretation
  - How do we extract relevant information from raw sensor data?
  - How do we represent and integrate this information over time?
- Robot locations have to be estimated
  - How can we identify that we are at a previously visited place?
  - This problem is the so-called data association problem.

Occupancy Grid Maps

- Introduced by Moravec and Elfes in 1985
- Represent environment by a grid
- Estimate the probability that a location is occupied by an obstacle
- Key assumptions
  - Occupancy of individual cells $m^{[xy]}$ is independent

$$\begin{aligned} Bel(m_t) &= P(m_t | u_1, z_1, \cdots, u_t, z_t) \\ &= \prod_{x,y} Bel(m_t^{[xy]}) \end{aligned}$$

  - Robot positions are known!

Updating Occupancy Grid Maps

- Idea: Update each individual cell using a binary Bayes filter

$$Bel(m_t^{[xy]}) =$$
$$\eta p(z_t|m_t^{[xy]}) \int p(m_t^{[xy]}|m_{t-1}^{[xy]}, u_{t-1}) Bel(m_{t-1}^{[xy]}) dm_{t-1}^{[xy]}$$

- Additional assumption: Map is static

$$Bel(m_t^{[xy]}) = \eta p(z_t|m_t^{[xy]}) Bel(m_{t-1}^{[xy]})$$

Updating Occupancy Grid Maps

- Update the map cells using the inverse sensor model

$$Bel(m_t^{[xy]}) =$$

$$\left[1 + \frac{1 - P(m_t^{[xy]} | z_t, u_{t-1})}{P(m_t^{[xy]} | z_t, u_{t-1})} \cdot \frac{P(m_t^{[xy]})}{1 - P(m_t^{[xy]})} \frac{1 - Bel(m_{t-1}^{[xy]})}{Bel(m_{t-1}^{[xy]})}\right]^{-1}$$

- Or use the log-odds representation

$$\begin{aligned}\bar{B}(m_t^{[xy]}) \quad &= \quad \log odds(m_t^{[xy]} | z_t, u_{t-1}) \\ &\quad -\log odds(m_t^{[xy]}) \\ &\quad +\bar{B}(m_{t-1}^{[xy]})\end{aligned}$$

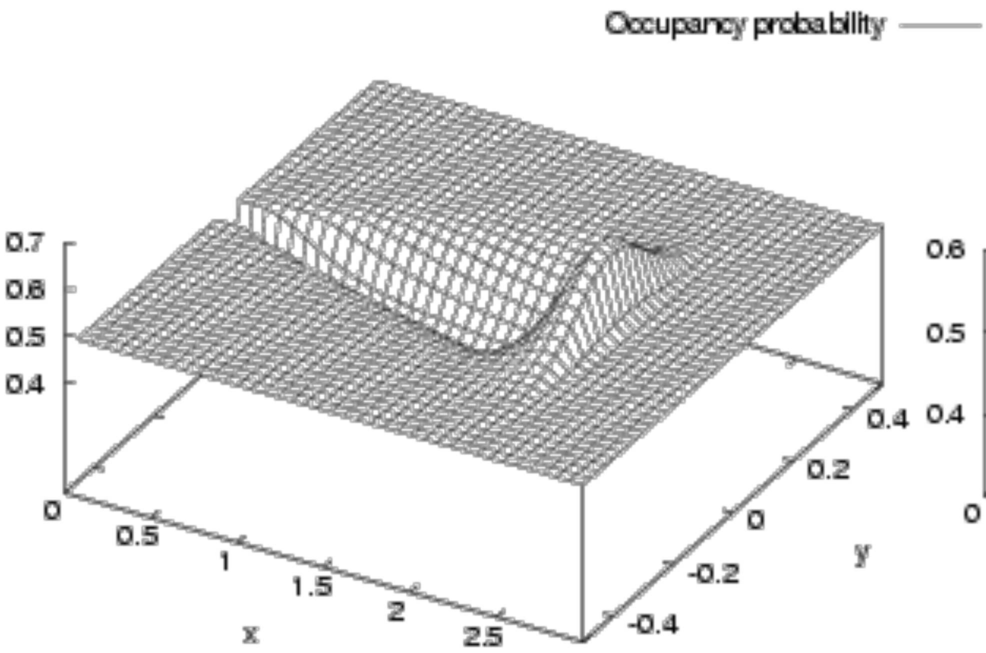
with:

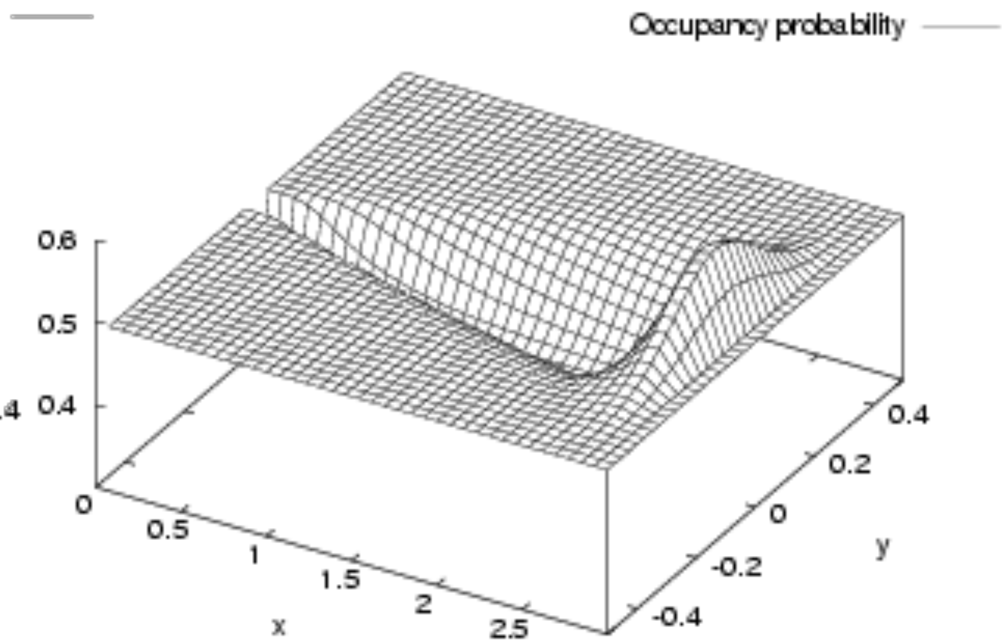
$$odds(x) := \frac{P(x)}{1-P(x)}$$

$$\bar{B}(m_t^{[xy]}) := \log odds(m_t^{[xy]})$$

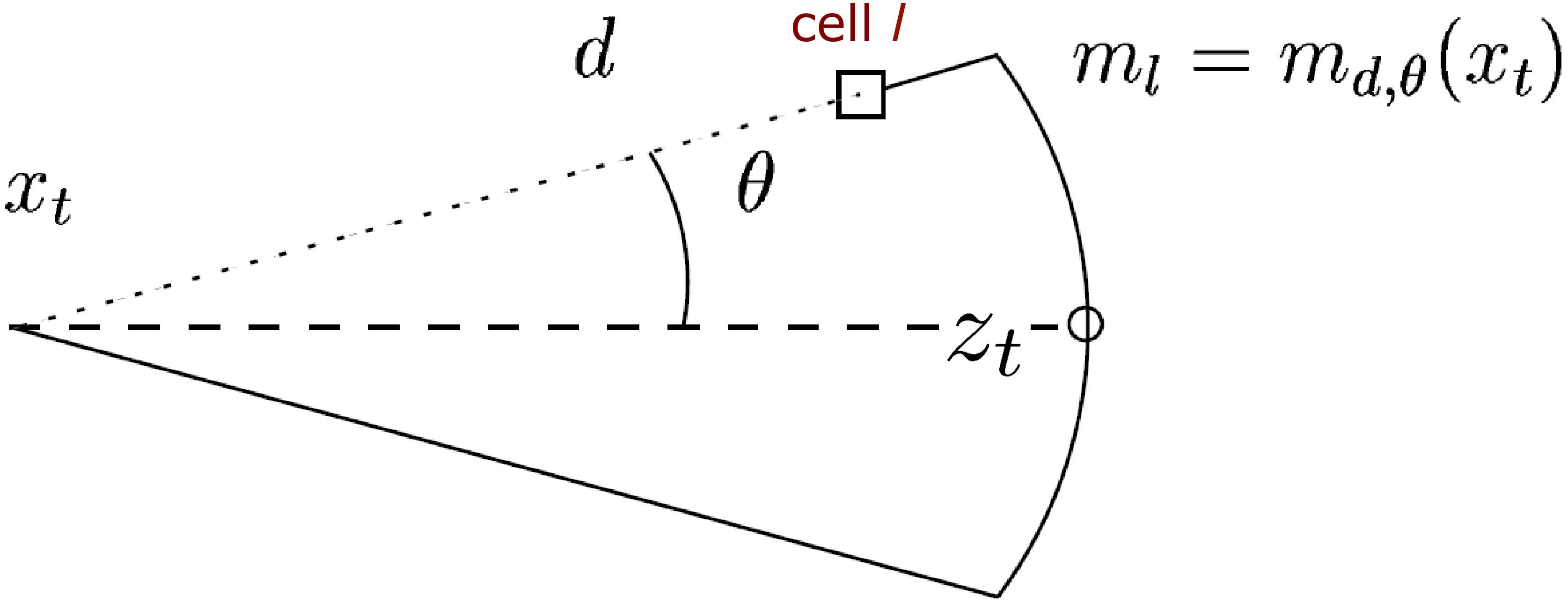
Typical Sensor Model for Occupancy Grid Maps (Sonar)

Combination of a linear function and a Gaussian:



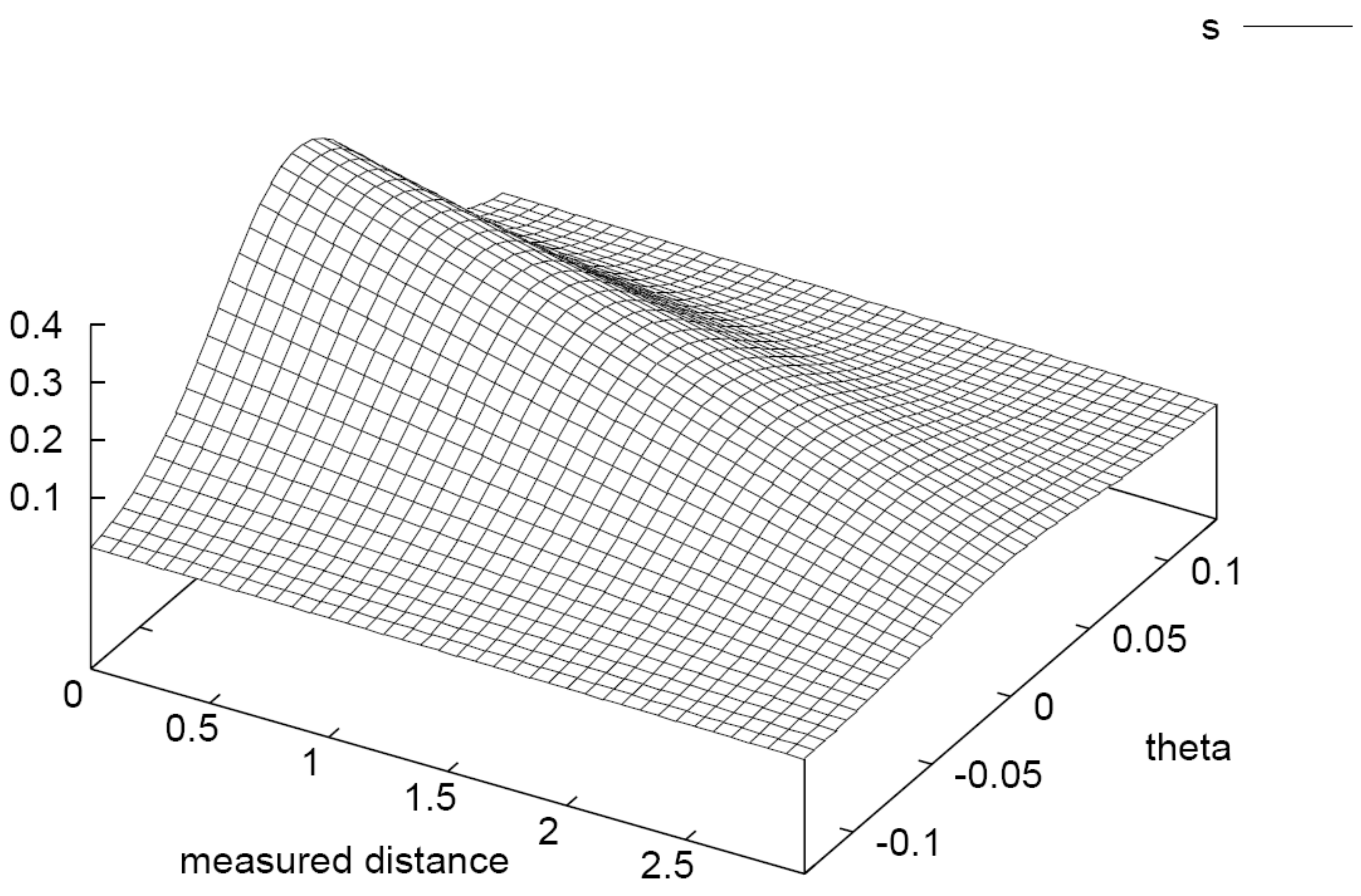


Key Parameters of the Model

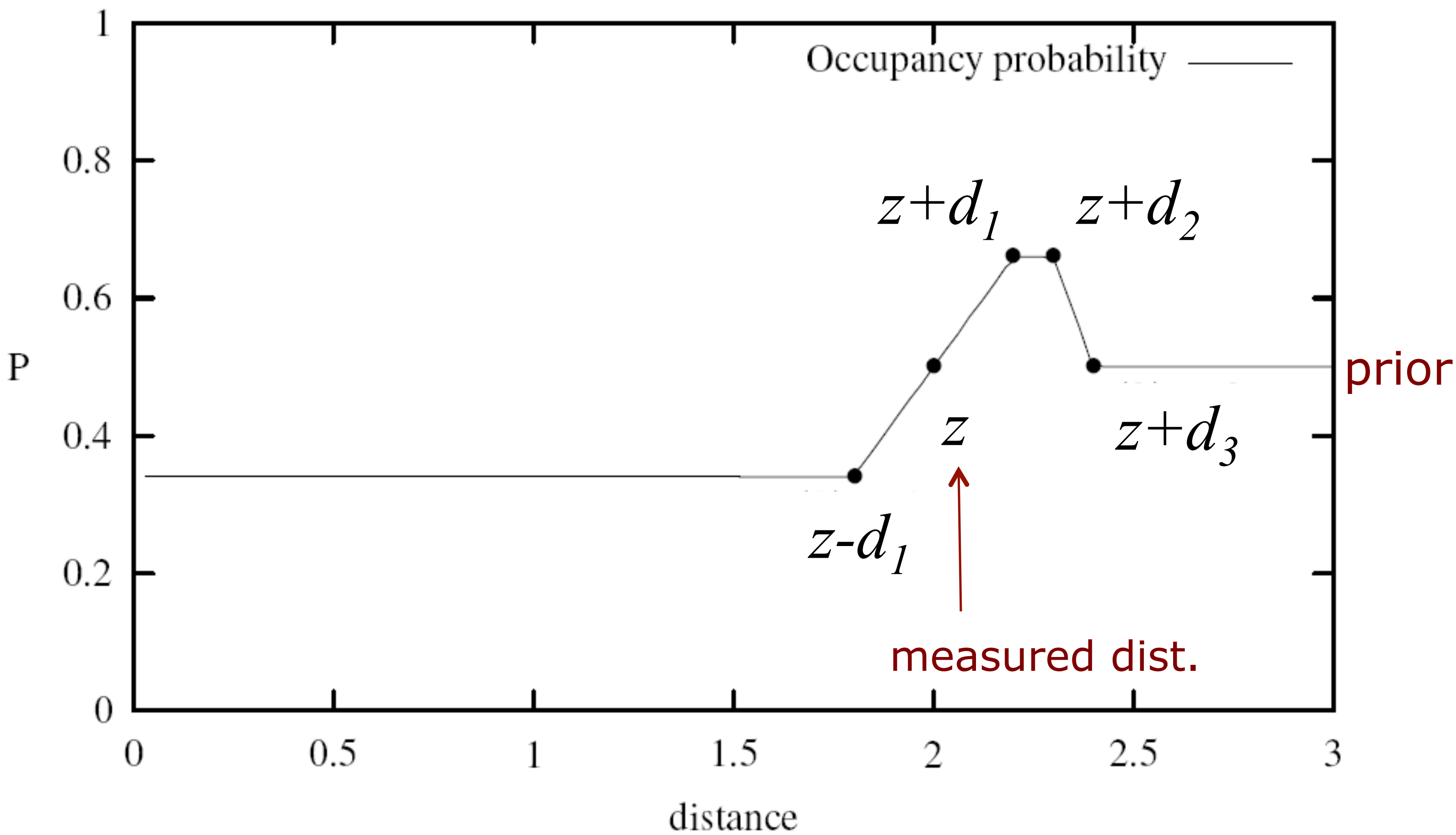


- Linear in z_t
- Gaussian in θ

Intensity of the Update

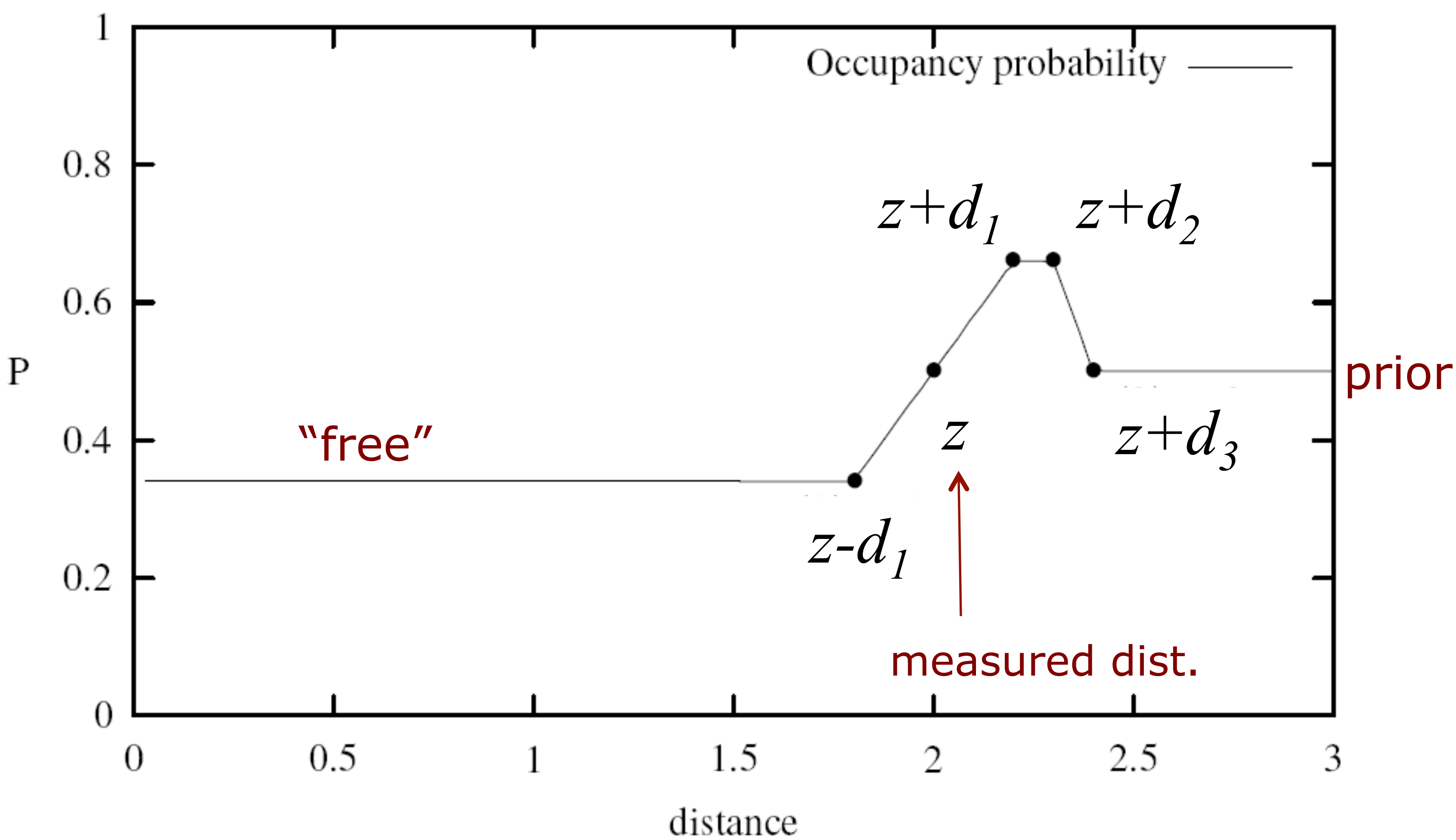


Occupancy Value Depending on the Measured Distance



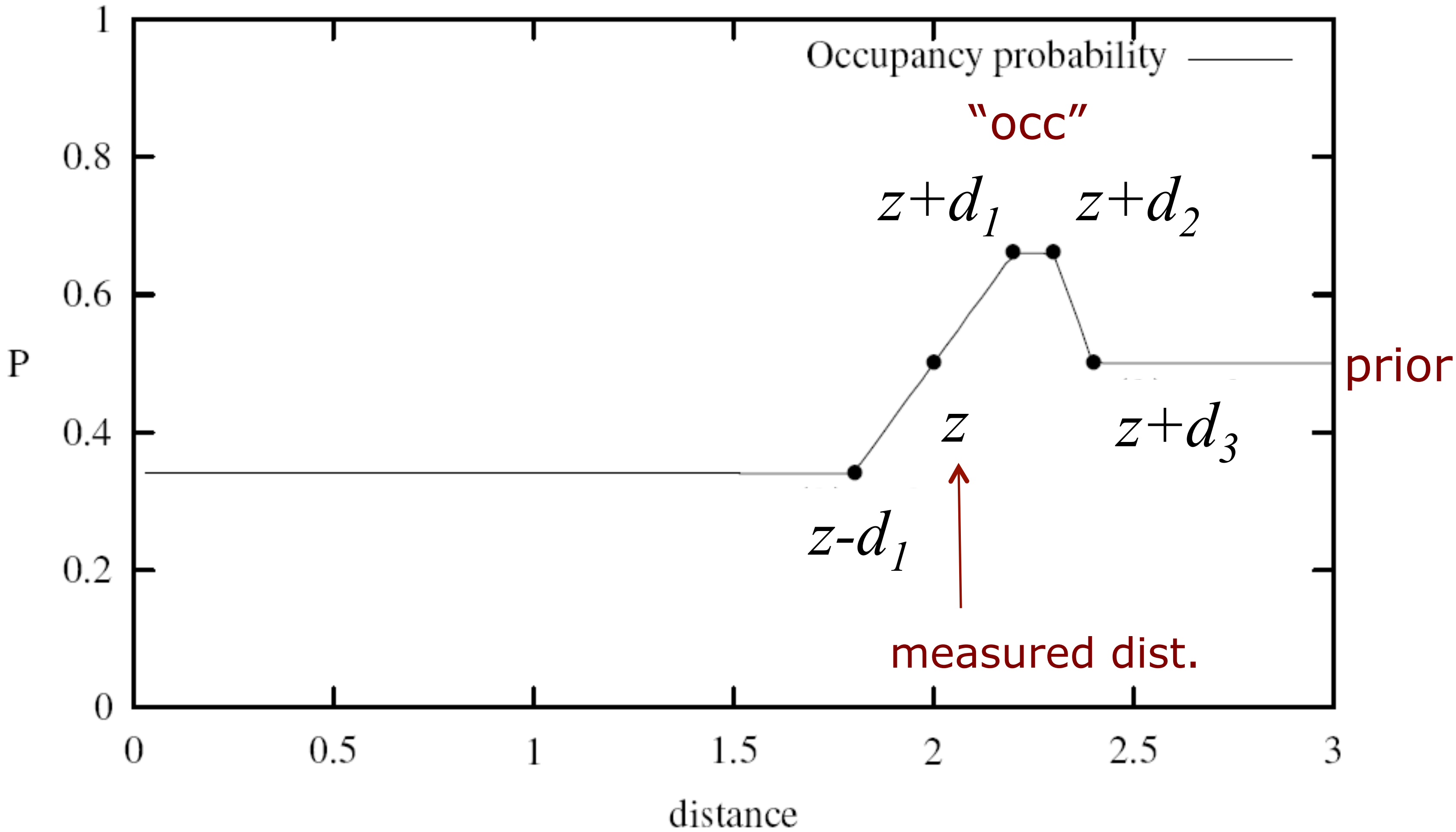
distance of cell from sensor

Occupancy Value Depending on the Measured Distance



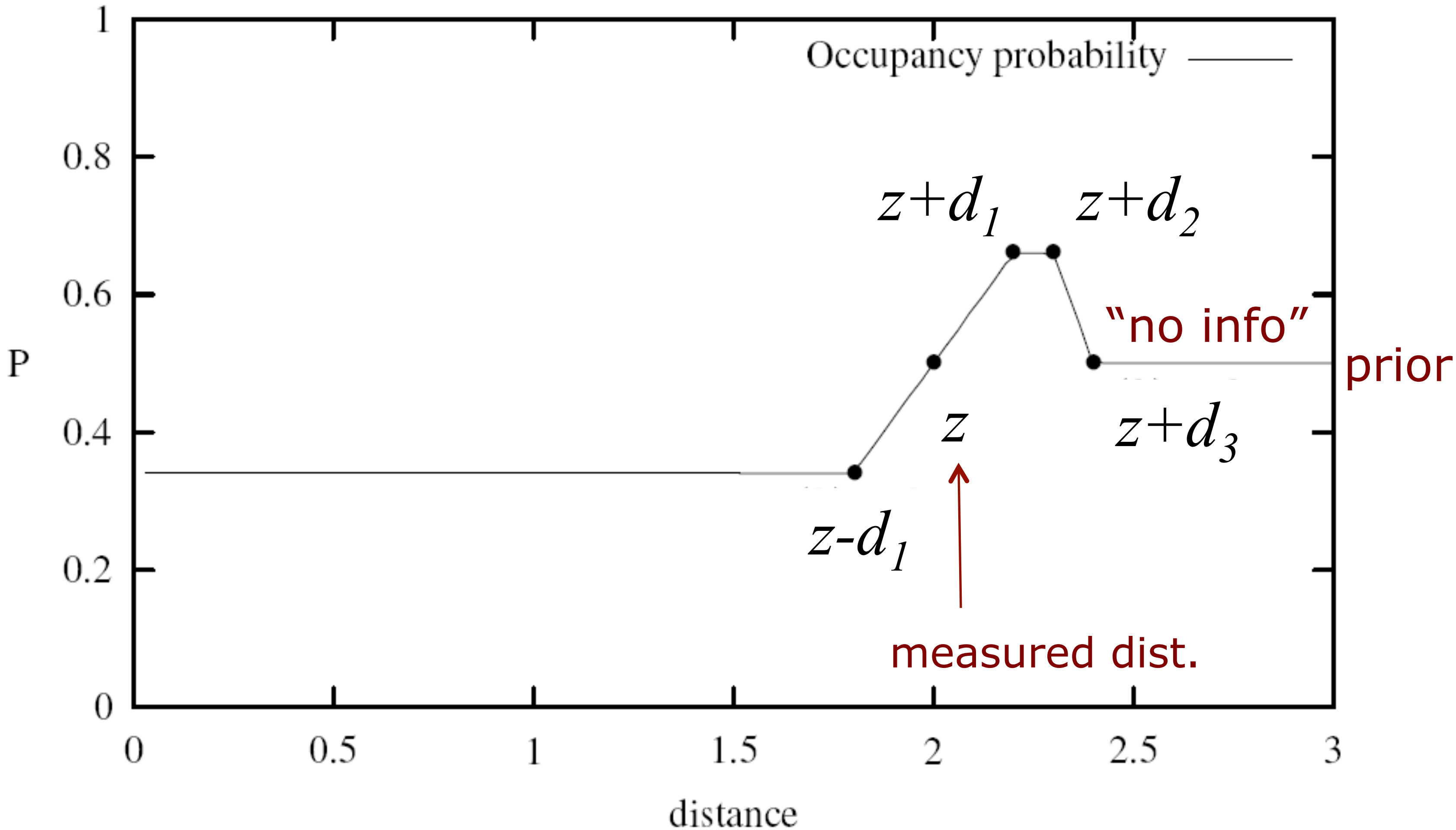
distance of cell from sensor

Occupancy Value Depending on the Measured Distance



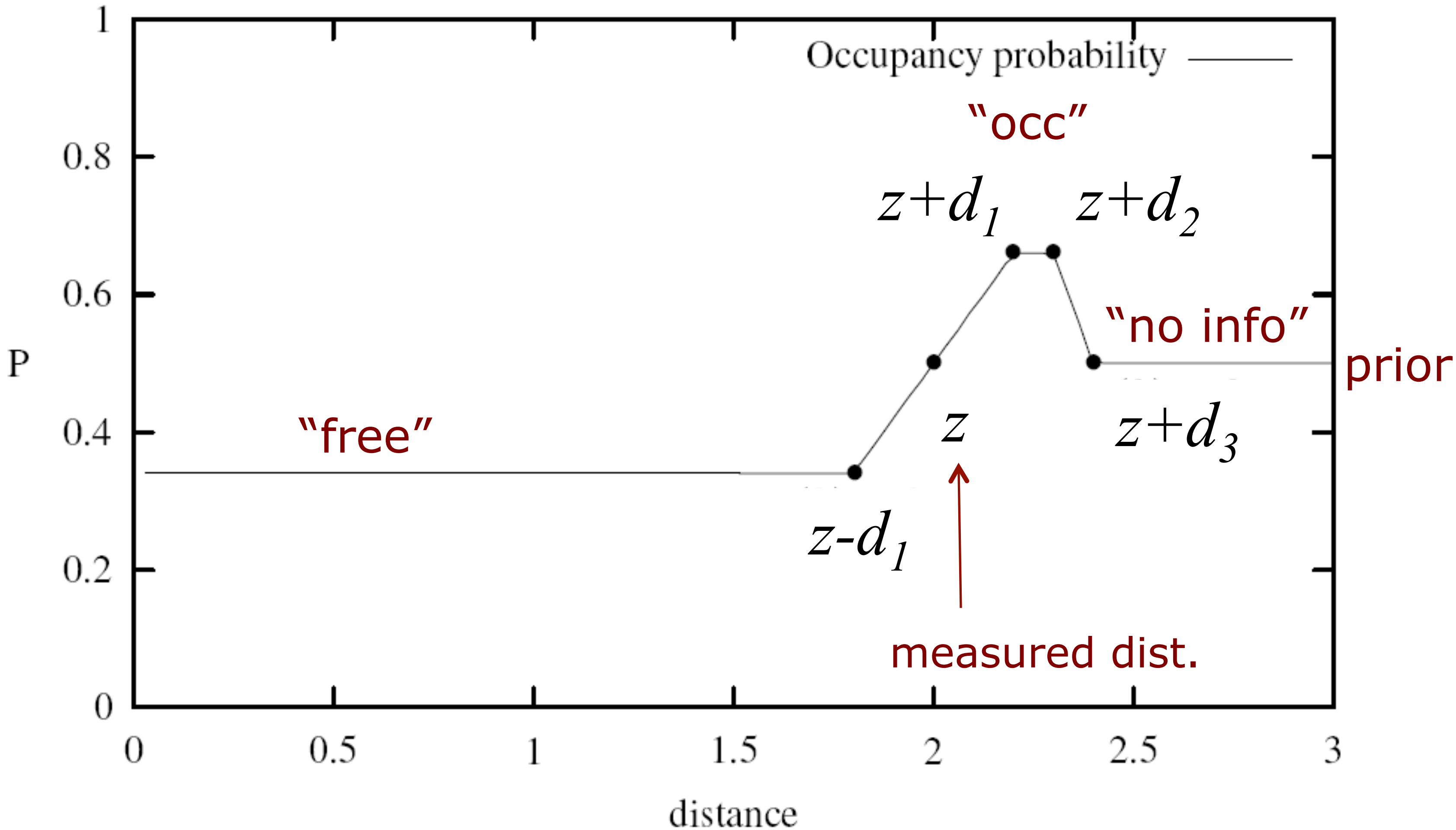
distance of cell from sensor

Occupancy Value Depending on the Measured Distance



distance of cell from sensor

Occupancy Value Depending on the Measured Distance



distance of cell from sensor

Calculating the Occupancy Probability Based on a Single Observation

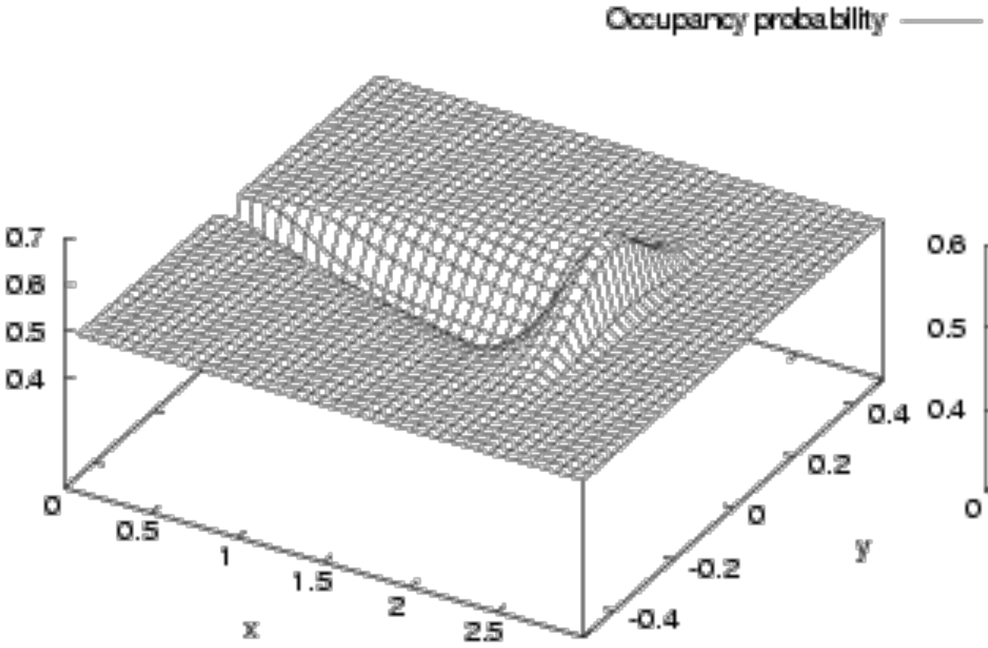
prior

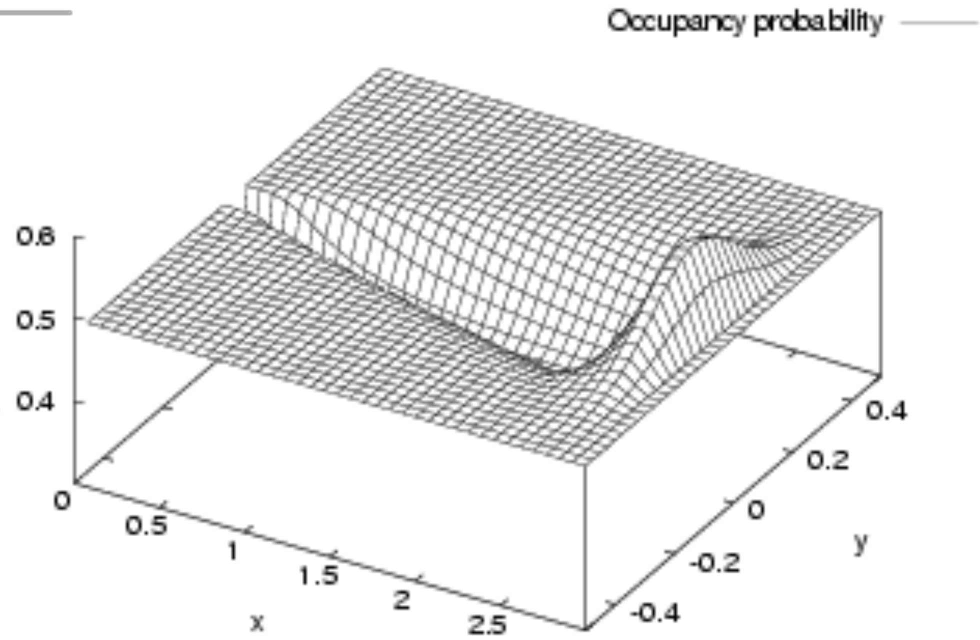
$$P(m_{d,\theta}(x_t) \mid z_t, x_t) = P(m_{d,\theta}(x_t))$$

$$+ \begin{cases} -s(z_t,\theta) & d < z_t - d_1 \quad \text{“free”} \\ -s(z_t,\theta) + \frac{s(z_t,\theta)}{d_1}\,(d - z_t + d_1) & d < z_t + d_1 \\ s(z_t,\theta) & d < z_t + d_2 \quad \text{“occ”} \\ s(z_t,\theta) - \frac{s(z_t,\theta)}{d_3 - d_2}\,(d - z_t - d_2) & d < z_t + d_3 \\ 0 & \text{otherwise.} \quad \text{“no info”} \end{cases}$$

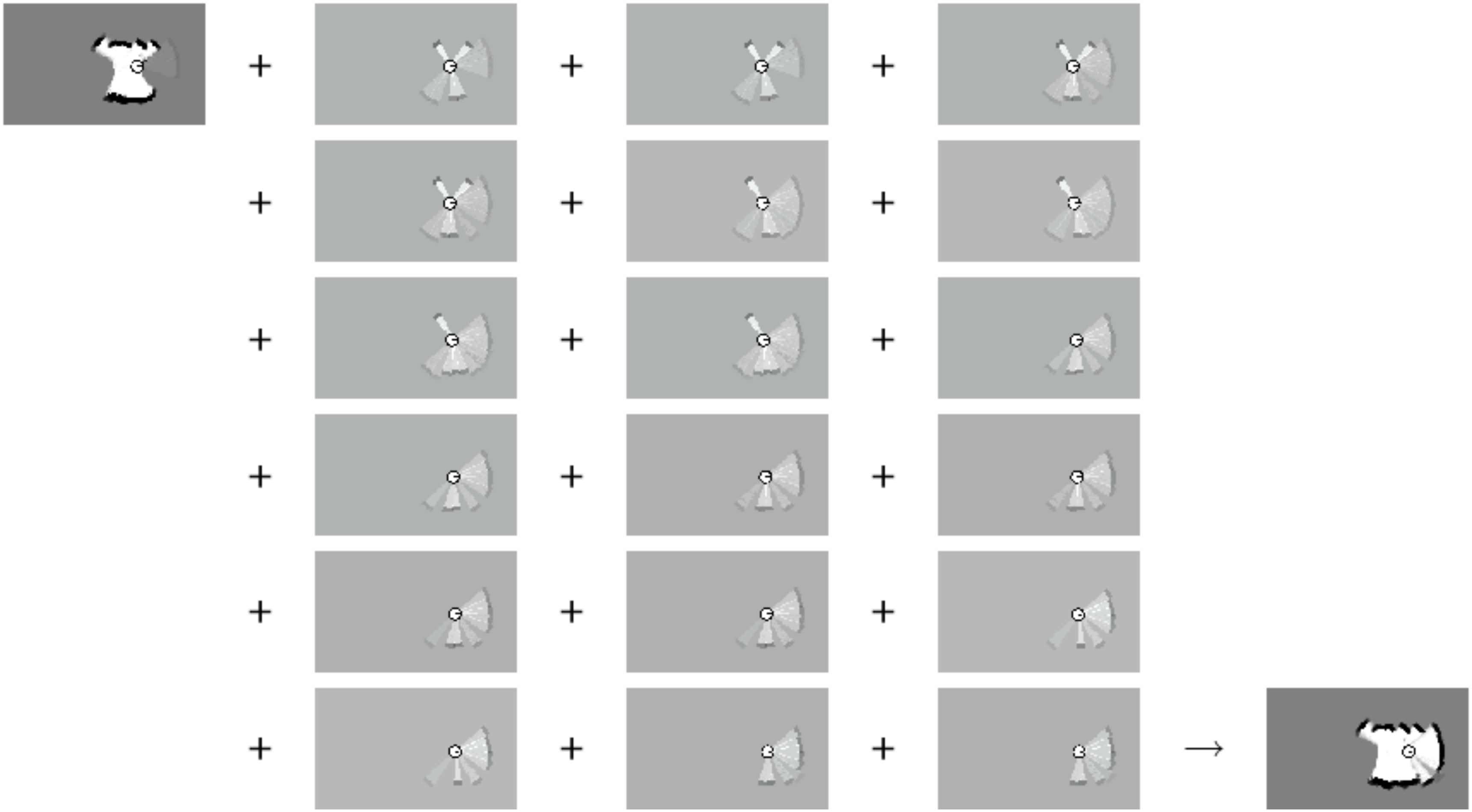
$s(z_t, \theta)$ intensity of the update (S. 13)

Resulting Model

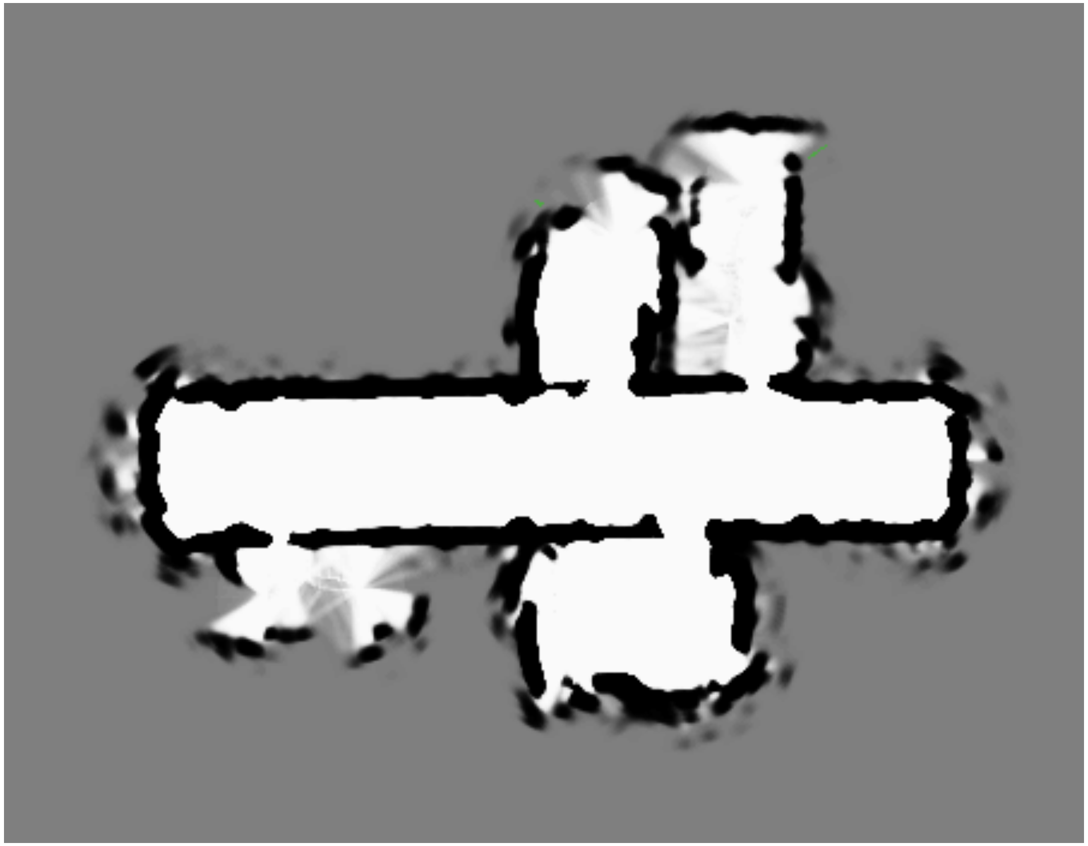




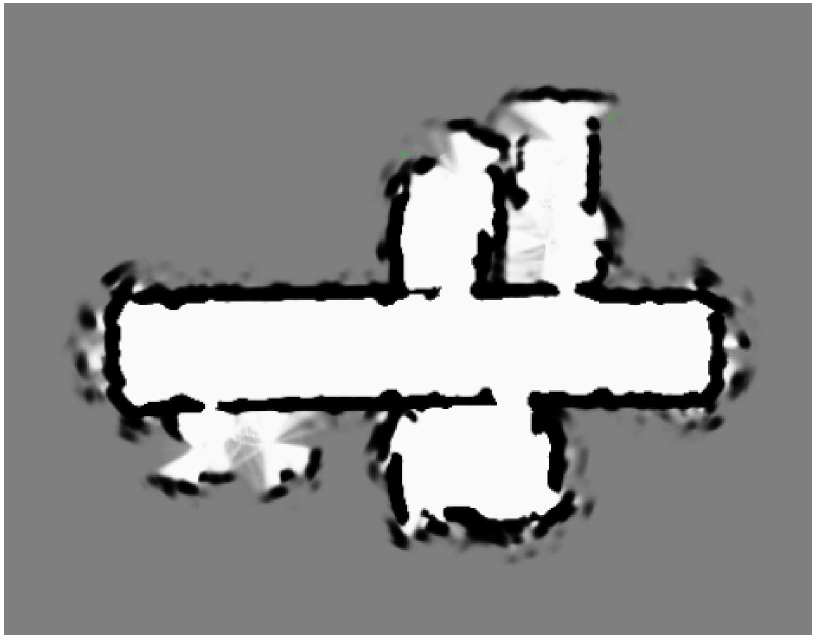
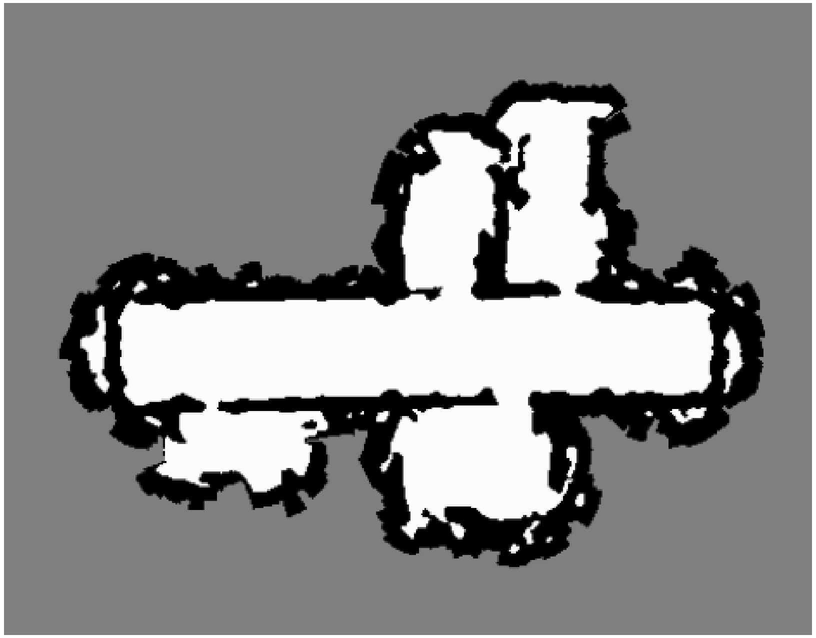
Incremental Updating of Occupancy Grids (Example)



Resulting Map Obtained with Ultrasound Sensors

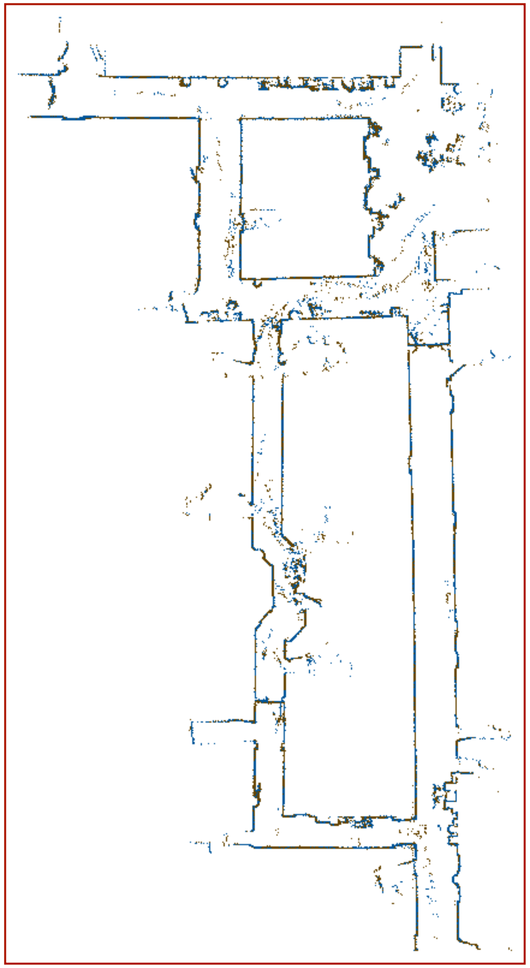


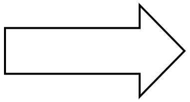
Resulting Occupancy and Maximum Likelihood Map

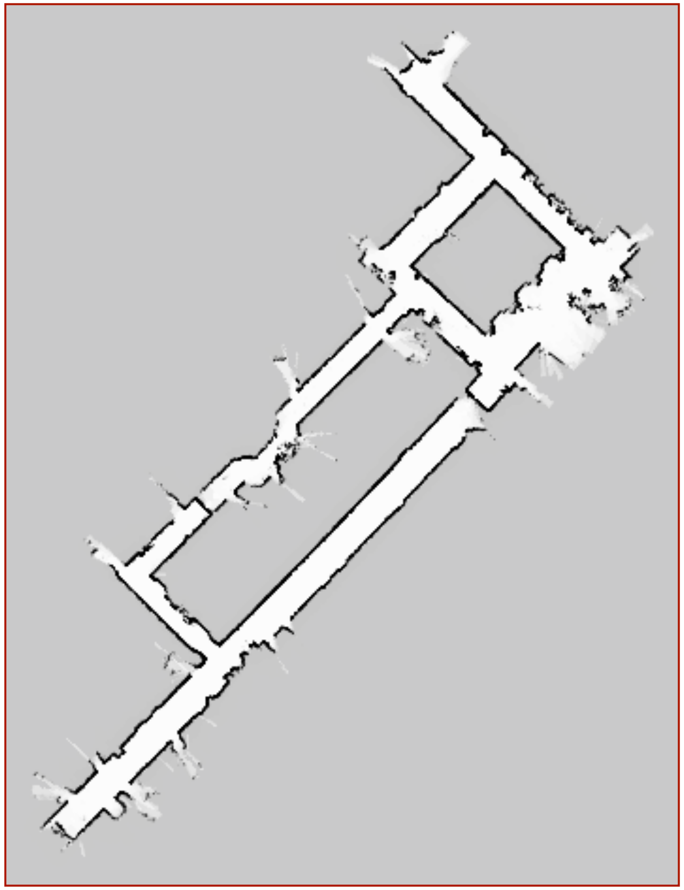
 

The maximum likelihood map is obtained by clipping the occupancy grid map at a threshold of 0.5

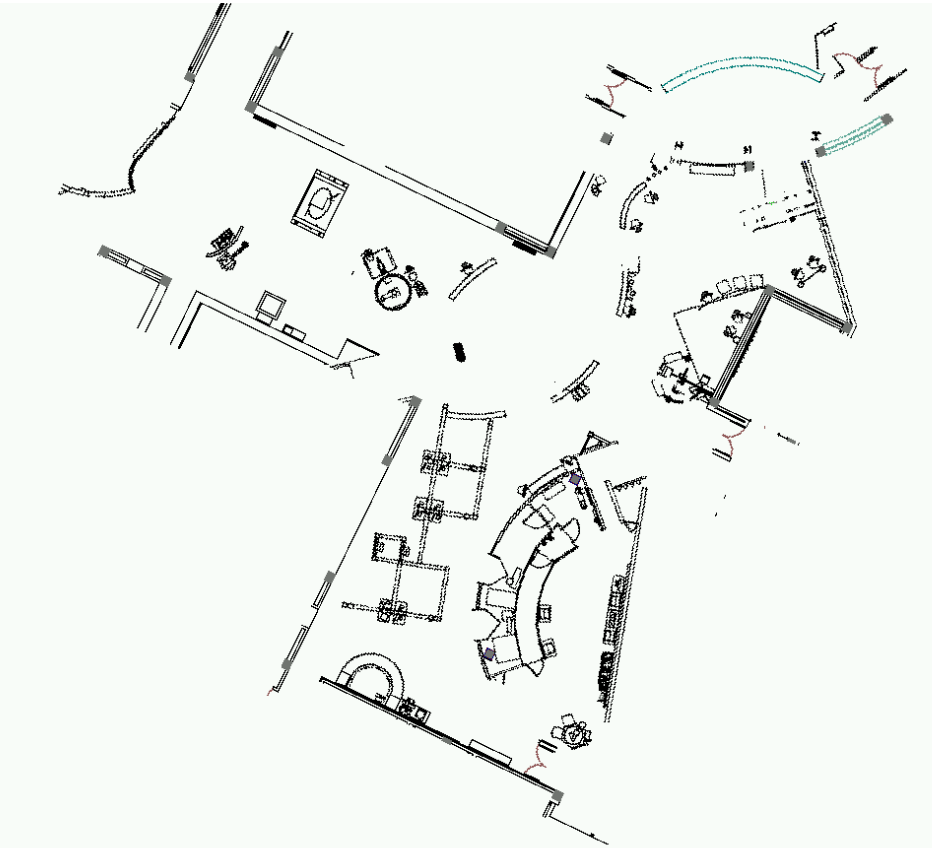
Occupancy Grids: From Scans to Maps (Laser)



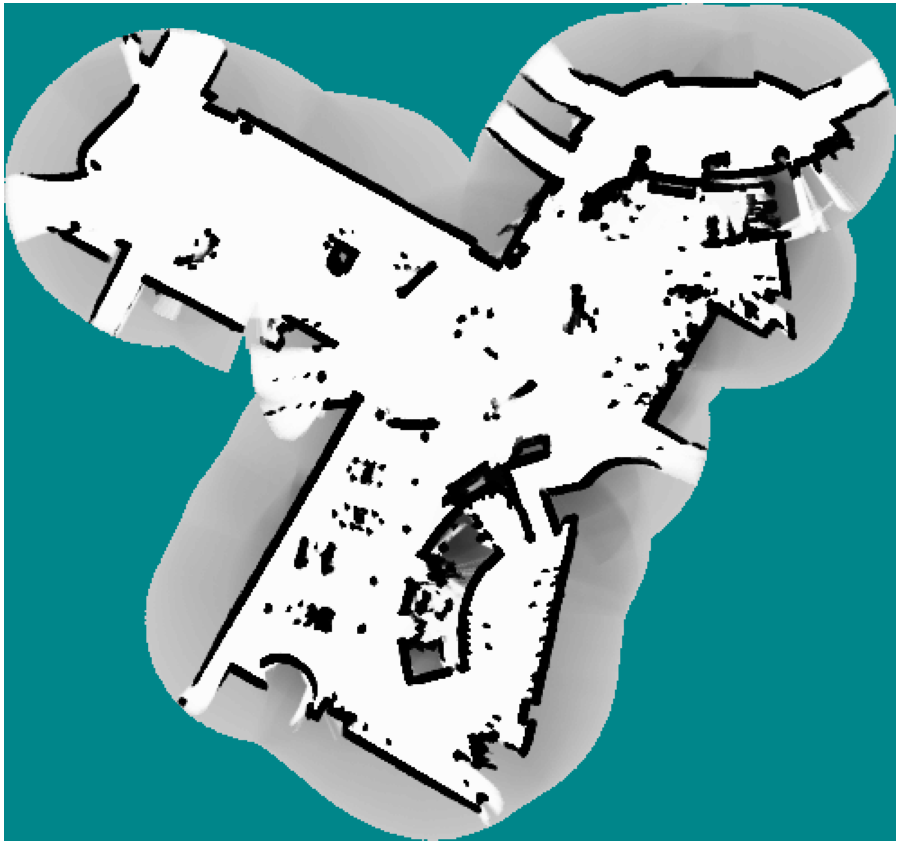




Tech Museum, San Jose



CAD map



occupancy grid map

Alternative: Counting Model

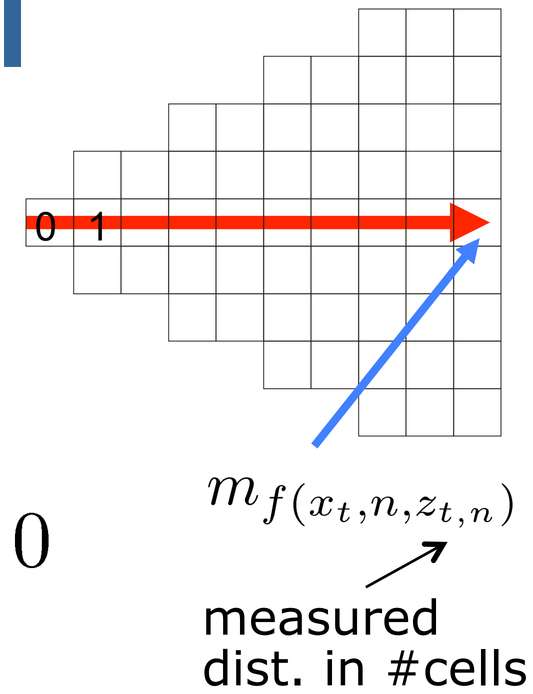
- For every cell count
  - hits(*x*,*y*): number of cases where a beam ended at <*x*,*y*>
  - misses(*x*,*y*): number of cases where a beam passed through <*x*,*y*>

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

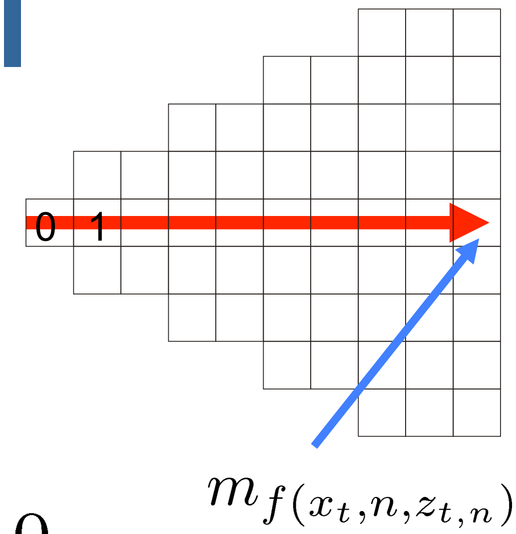
- Value of interest: *P*(*reflects*(*x*,*y*))

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

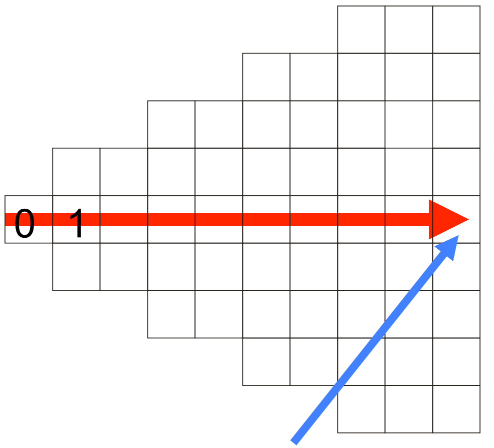


- Pose at time *t*: x_t
- Beam *n* of scan at time *t*: $z_{t,n}$
- Maximum range reading: $\zeta_{t,n} = 1$
- Beam reflected by an object: $\zeta_{t,n} = 0$

max range: "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: “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

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

- Assuming a uniform prior probability for P(*m*), this is equivalent to maximizing (Bayes' rule)

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

since z_t independent and only depend on x_t

Computing the Most Likely Map

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

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

Computing the Most Likely Map

"beam *n* ends in cell *j*"

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

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

Computing the Most Likely Map

"beam n ends in cell j"

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

"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

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

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 amounts 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 1000 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

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

- Whereas the reflection map yields a value of 0.6, the occupancy grid value converges to 1.

Summary

- Occupancy grid maps are a popular approach to represent the environment given known poses.
- Each cell is considered independently from all others.
- Occupancy grids store the probability that the corresponding area in the environment is occupied.
- They can be learned efficiently using a probabilistic approach.
- Reflection maps are an alternative representation.
- They store in each cell the probability that a beam is reflected by this cell.
- The counting procedure underlying reflection maps yield the optimal map given the proposed sensor model.