# Mapping with Known Poses

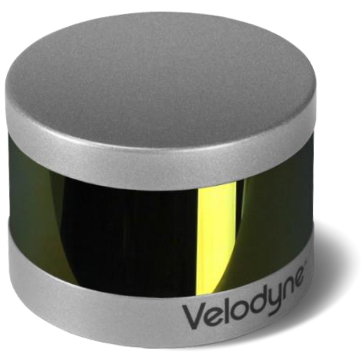
**Nived Chebrolu**

Slides adapted from Cyrill Stachniss at Uni Bonn.

# Why Mapping?

- Maps are required for most robotic tasks like localization, planning, ...
- Task of understanding how the environment looks like
- Learning maps from sensor data is one of the fundamental tasks in robotics

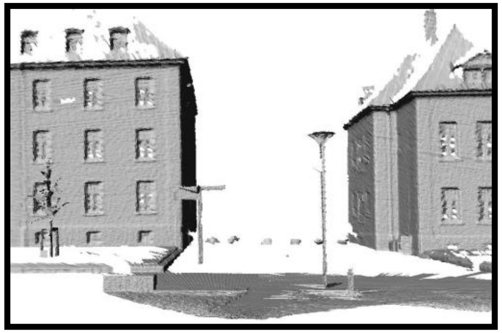
# Popular Sensors

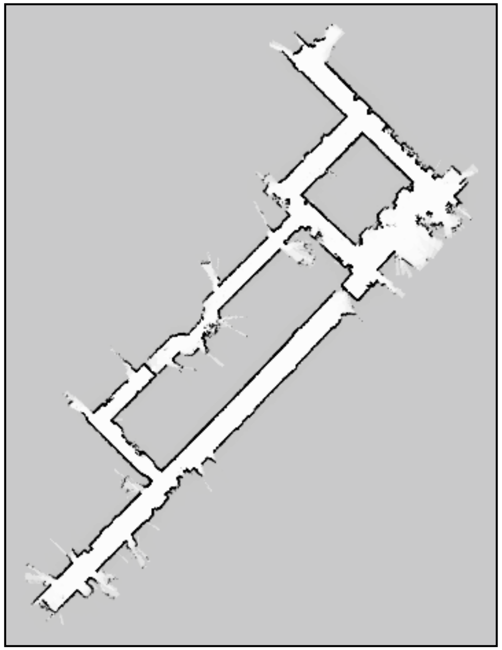





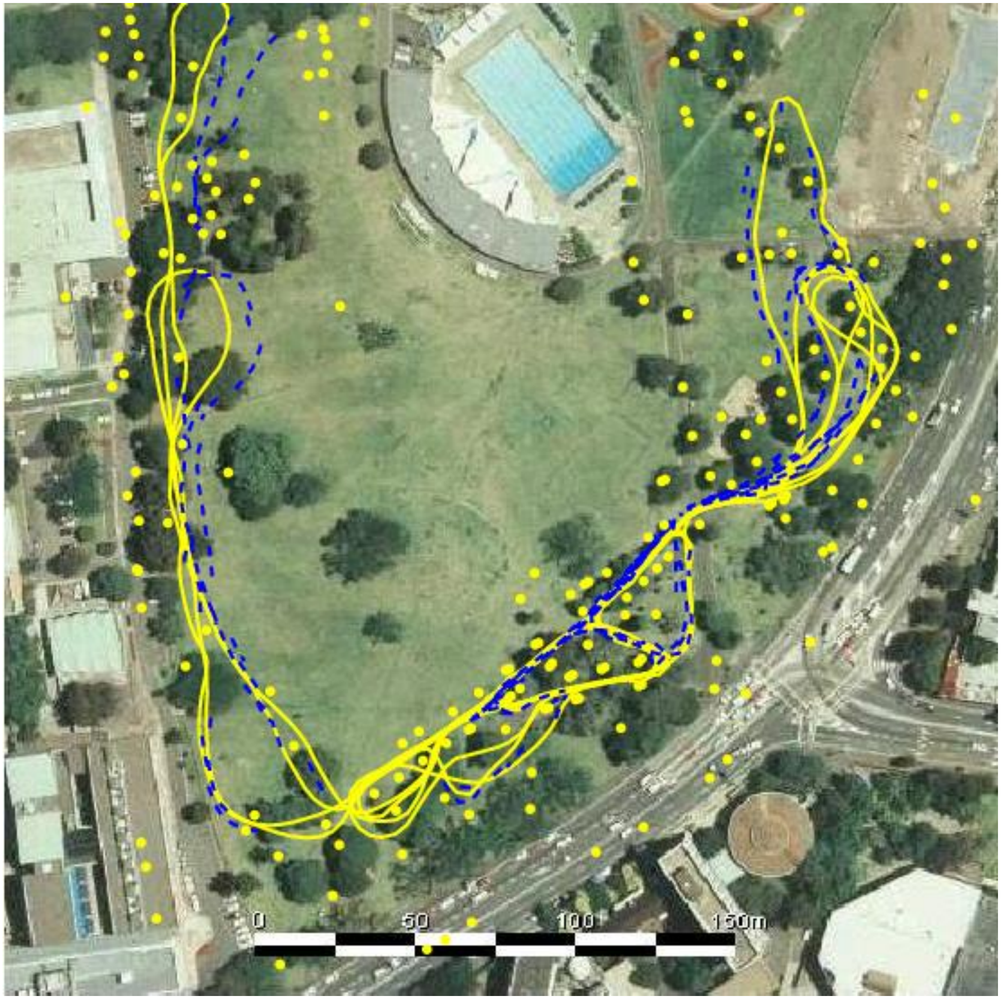
Courtesy: Matrix Vision, FLIR, Microsoft, Hokuyo, Velodyne

# Features vs. Volumetric Maps



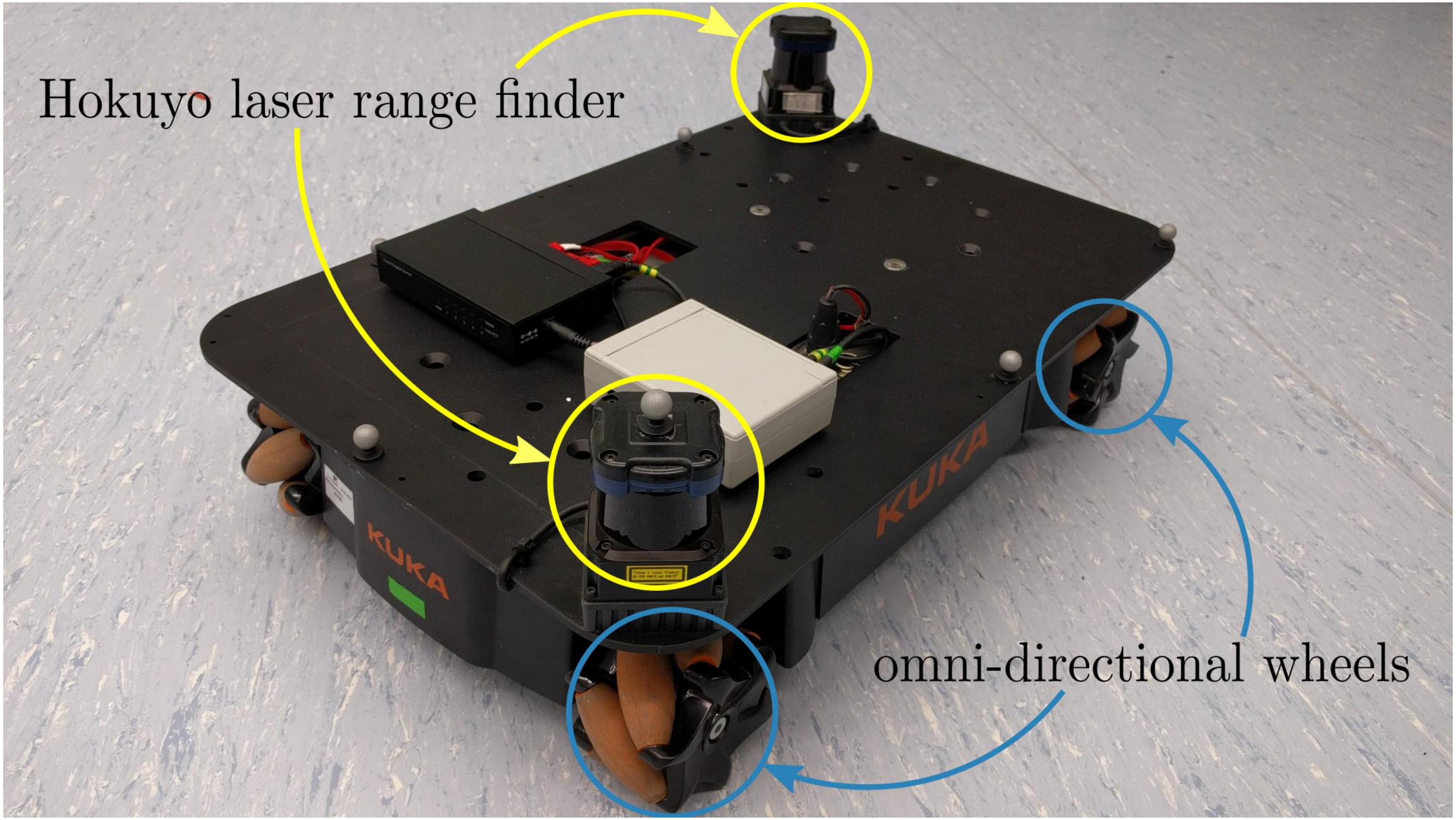


Courtesy: D. Hähnel

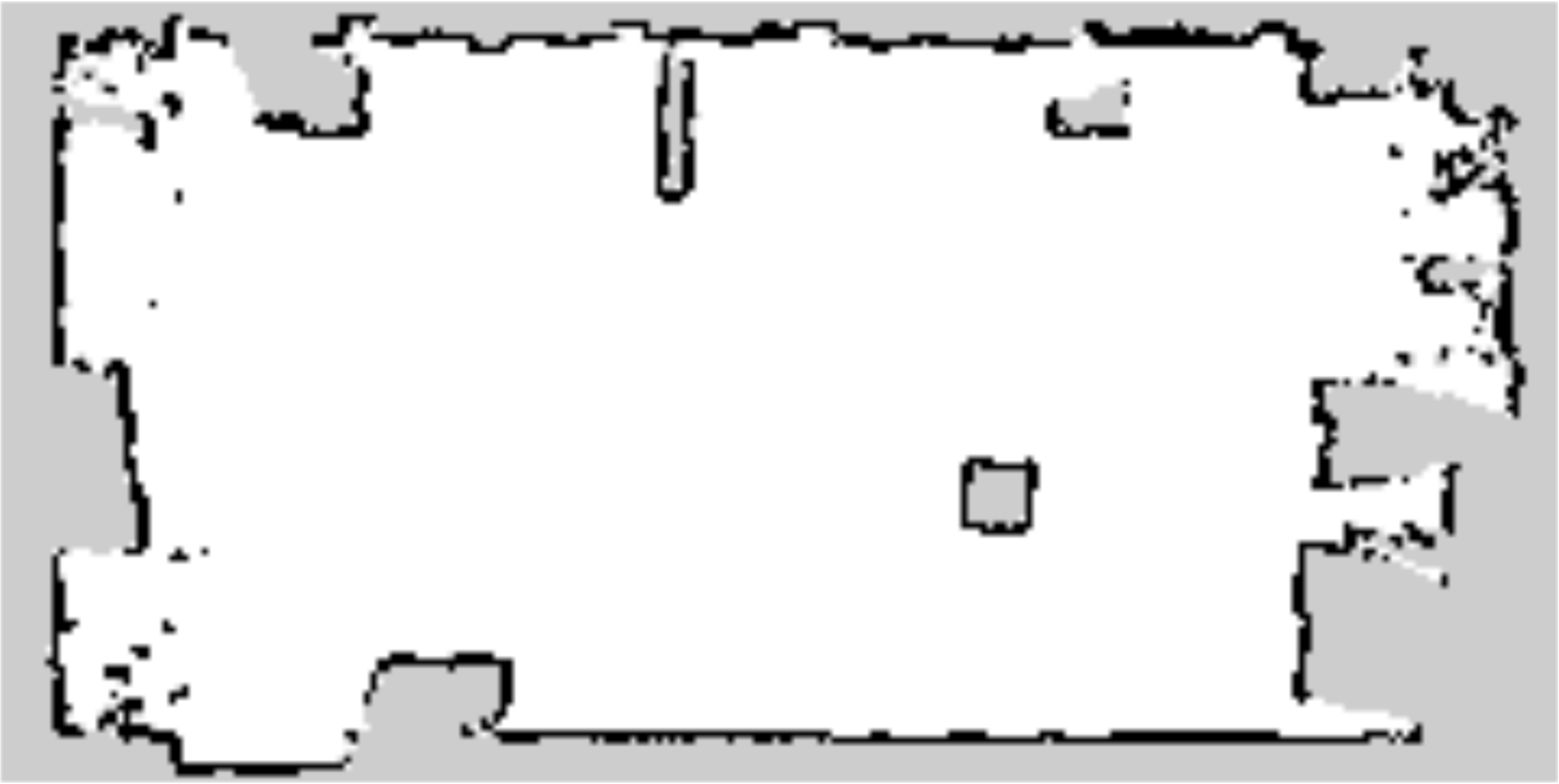



Courtesy: E. Nebot

# Mapping Example

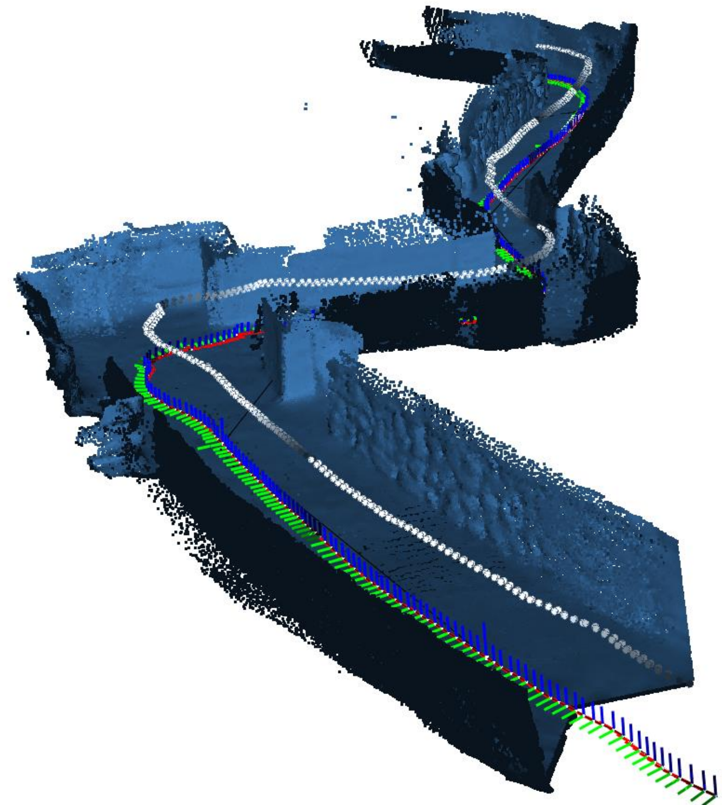


# Mapping Example



Occupancy grid map

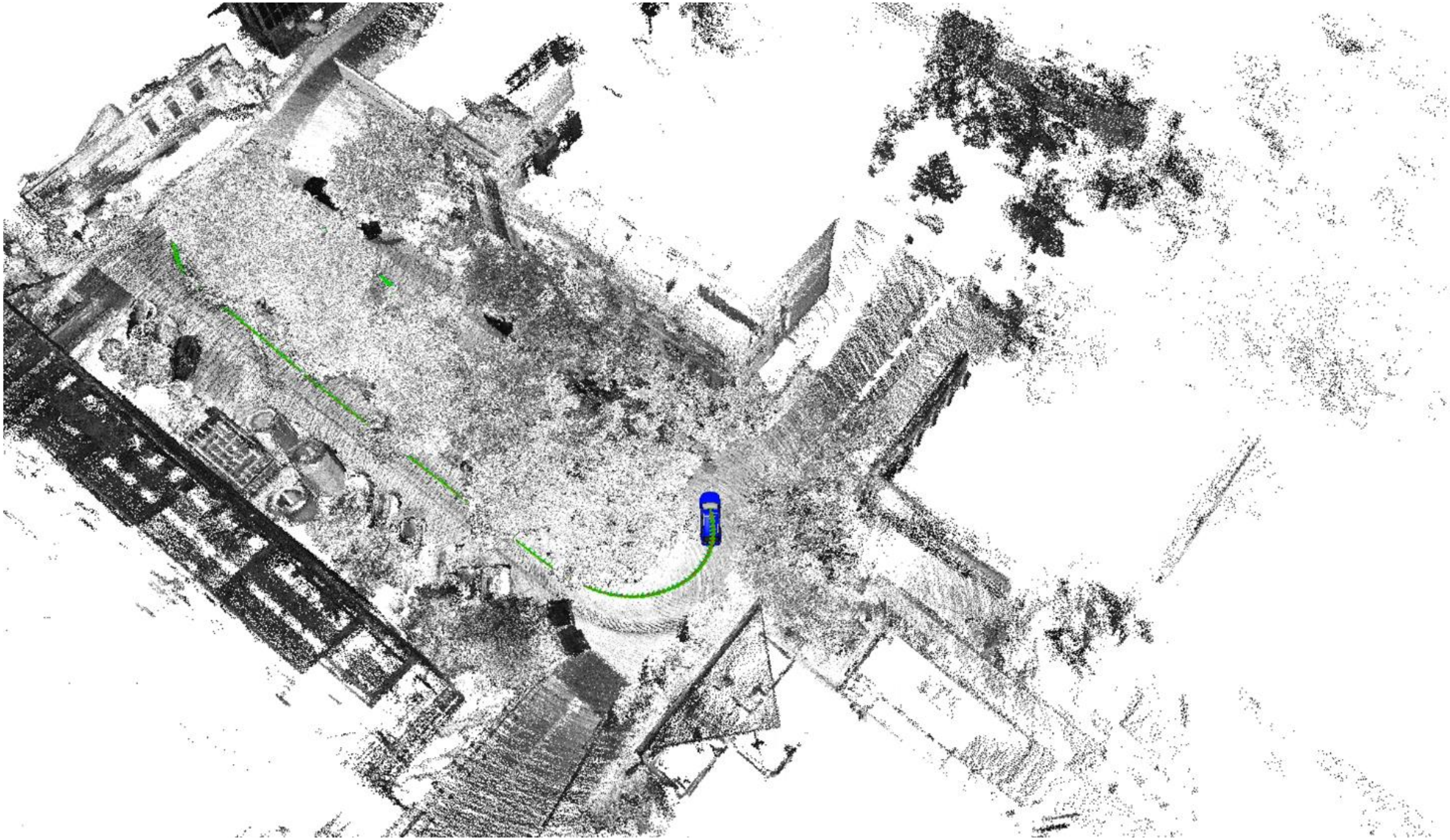
# Mapping Example



Point cloud

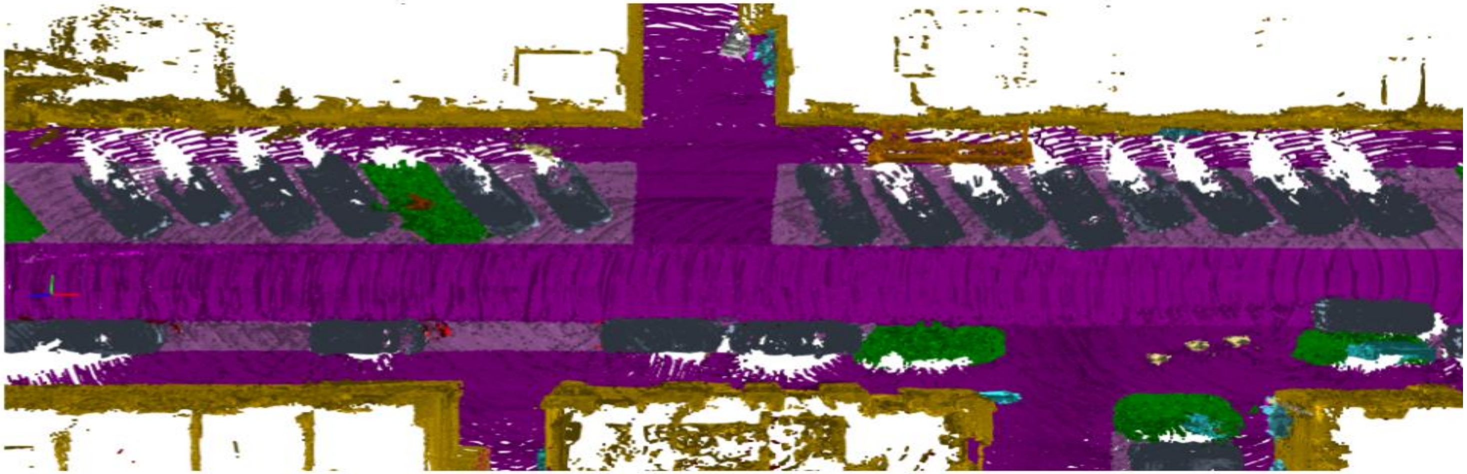
# Mapping Example

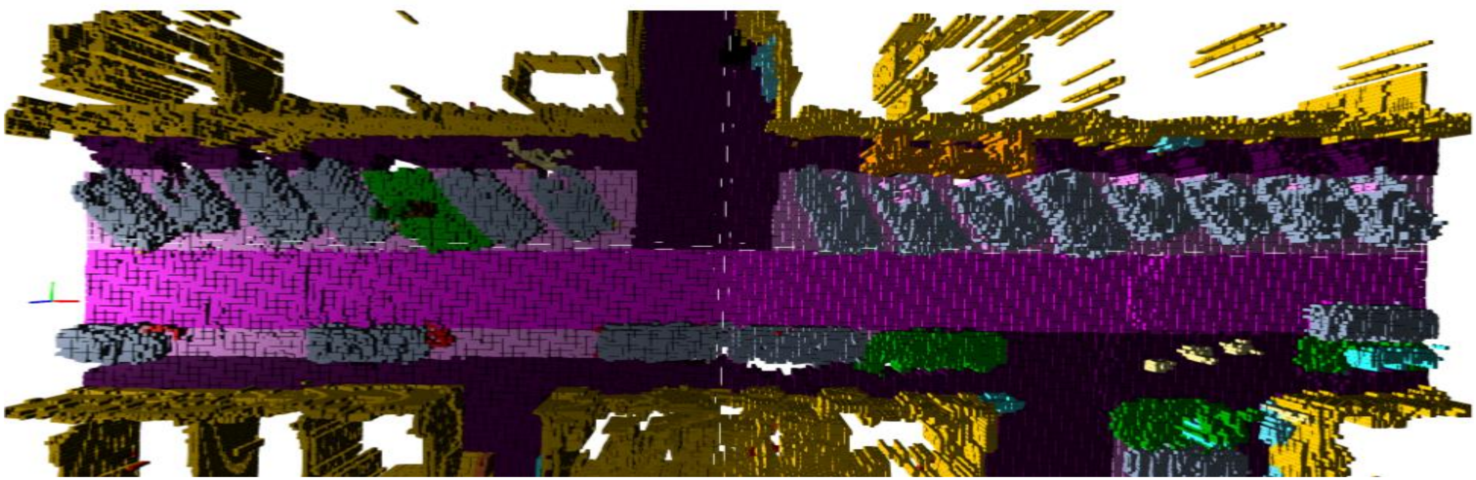
# Mapping Example



Point cloud

# Mapping Example





Voxel map

# Description of Mapping Task

- Compute the most likely map given the sensor data

$$m^* = \operatorname{argmax} P(m|u_1, z_1 \dots u_t, z_t)$$

# Description of Mapping Task

- Compute the most likely map given the sensor data

$$m^* = \operatorname{argmax}_m P(m \mid u_1, z_1, \dots, u_t, z_t)$$

- Today, we see **how to compute the map given the robot's pose**

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

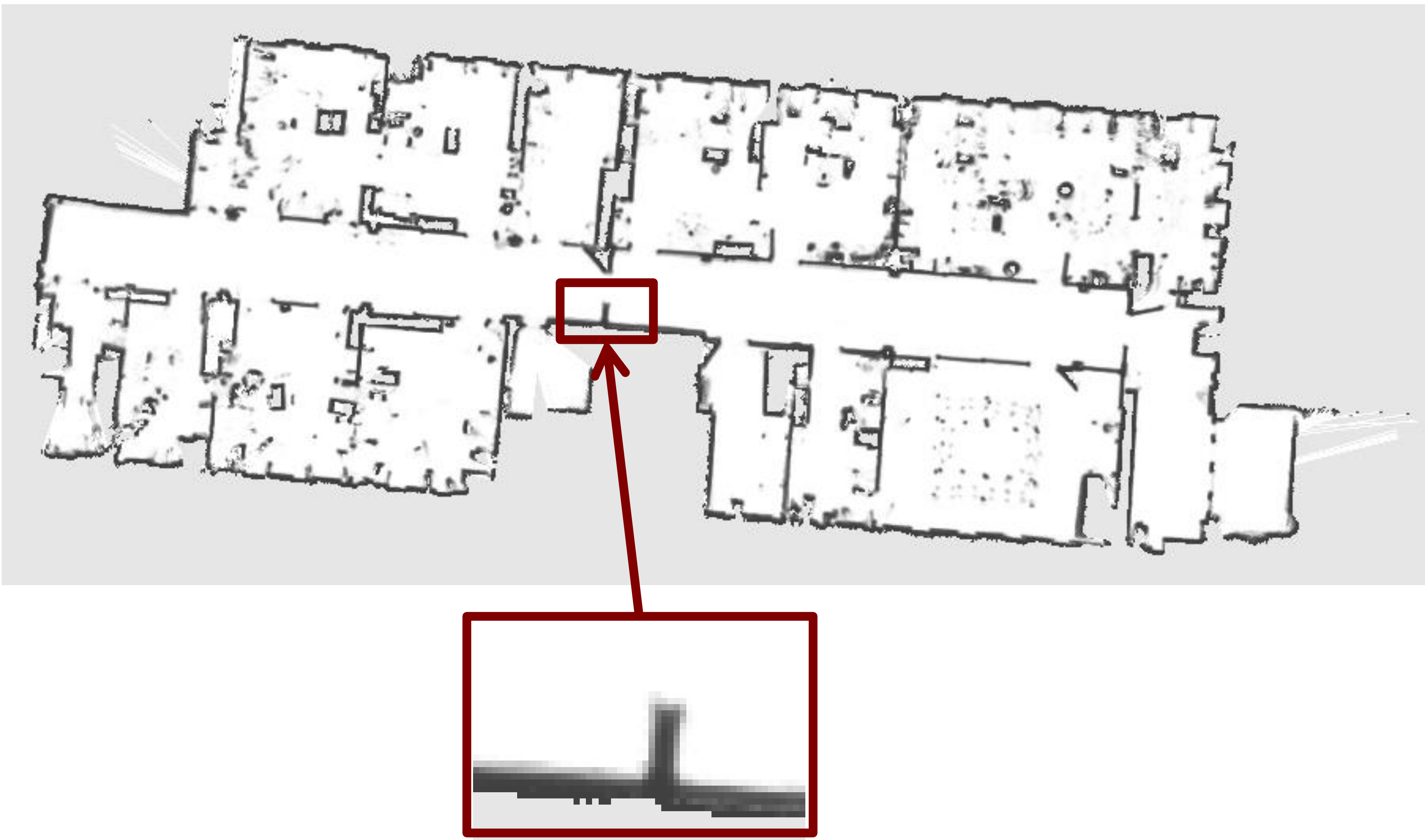
# Features

- Map is a set of features (often points)
- Standard approach in photogrammetry and computer vision
- Multiple feature observations improve the landmark position estimate

# Grid Maps

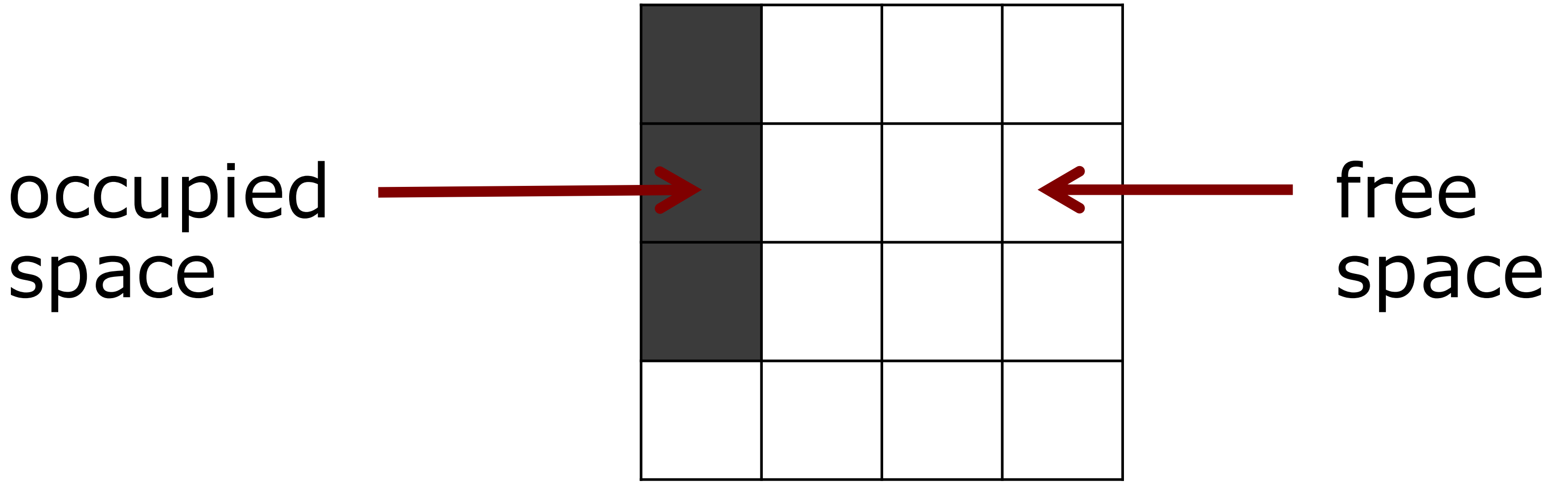
- Map models occupancy of the space
- Discretize the world into cells
- Grid structure is rigid
- Each cell is assumed to be occupied or free space
- Non-parametric model
- Large maps may require substantial memory resources
- Do not rely on a feature detector

# Occupancy Gird Map Example



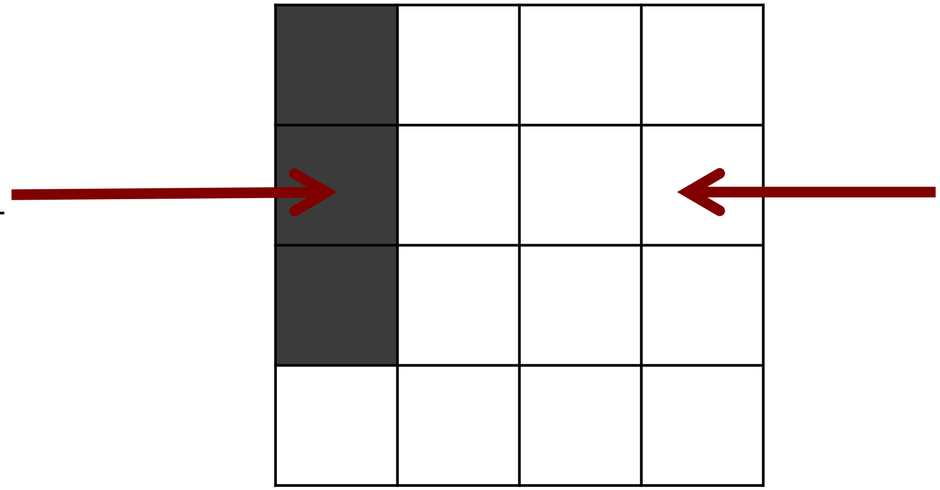
# Assumption 1

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



$p(m_i) \to 1$

$p(m_j) \to 0$

# 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

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

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

- The opposite event

$$P(M_i = free) = P_{\text{free}}(M_i)$$

$$= 1 - P_{\text{occ}}(M_i) = p(\neg m_i)$$

# Occupancy Probability Example

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

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

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

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

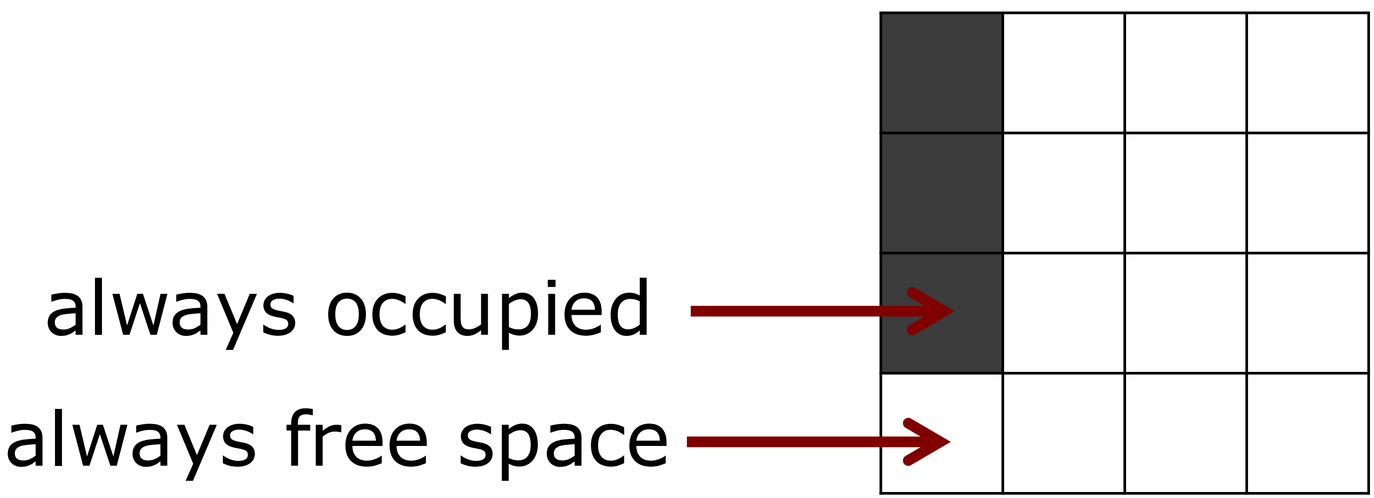
$$P(M_i = free) = p(\neg m_i) = 1 - p(m_i) = 1$$

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

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

# Assumption 2

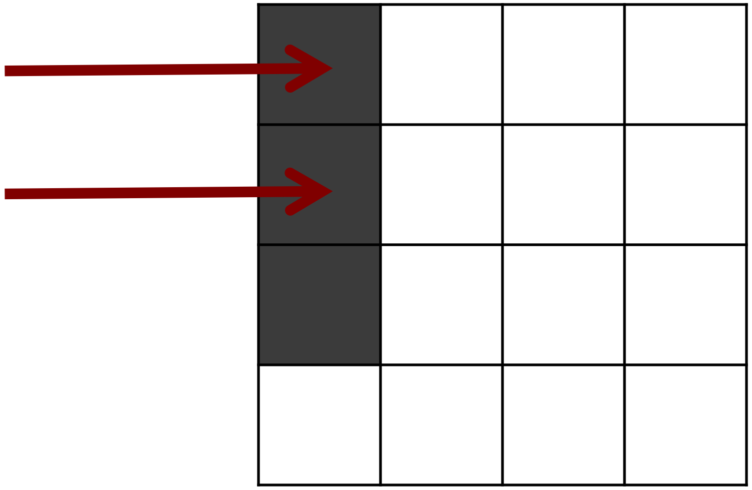
The world is **static** (most mapping systems make this assumption)



# 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

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

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

➡ 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

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

# Static State Binary Bayes Filter

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

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

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

# 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

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

# 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:

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

# Static State Binary Bayes Filter

By computing the ratio of both probabilities, we obtain:

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

# Static State Binary Bayes Filter

By computing the ratio of both probabilities, we obtain:

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

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

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

# From Ratio to Probability

We can easily turn this ratio called the **odds ratio** into the probability:

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

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

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

# From Ratio to Probability

Using $p(x) = \left[1 + Odds(x)^{-1}\right]^{-1}$ gives us

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

**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

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

- or in short

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

# Occupancy Mapping Algorithm

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

```
for all cells m_i do
    if m_i in perceptual field of z_t then
        l_{t,i} = l_{t-1,i} + 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}}
```

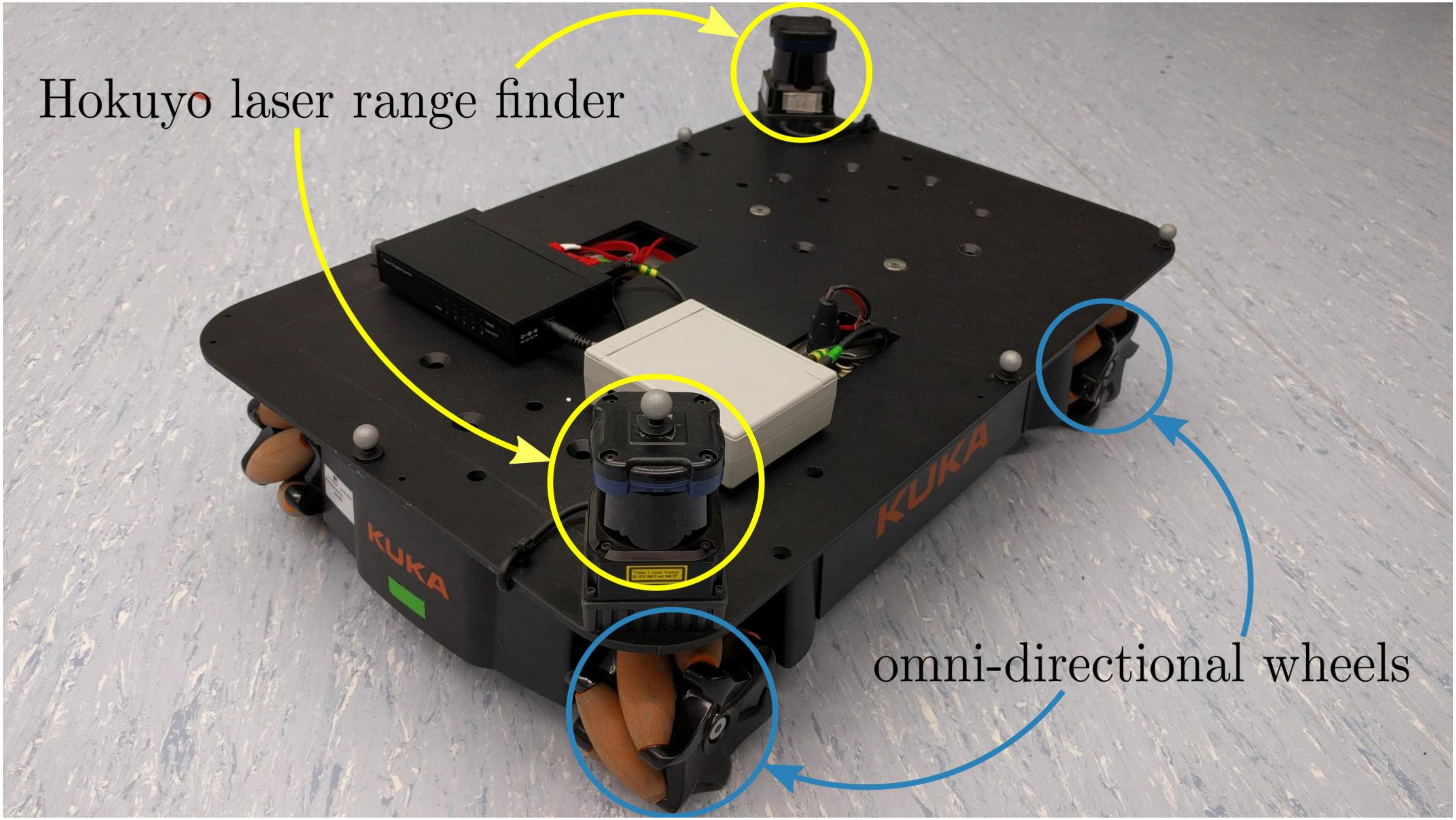
**highly efficient, we only have to compute sums**

Courtesy: Thrun, Burgard, Fox

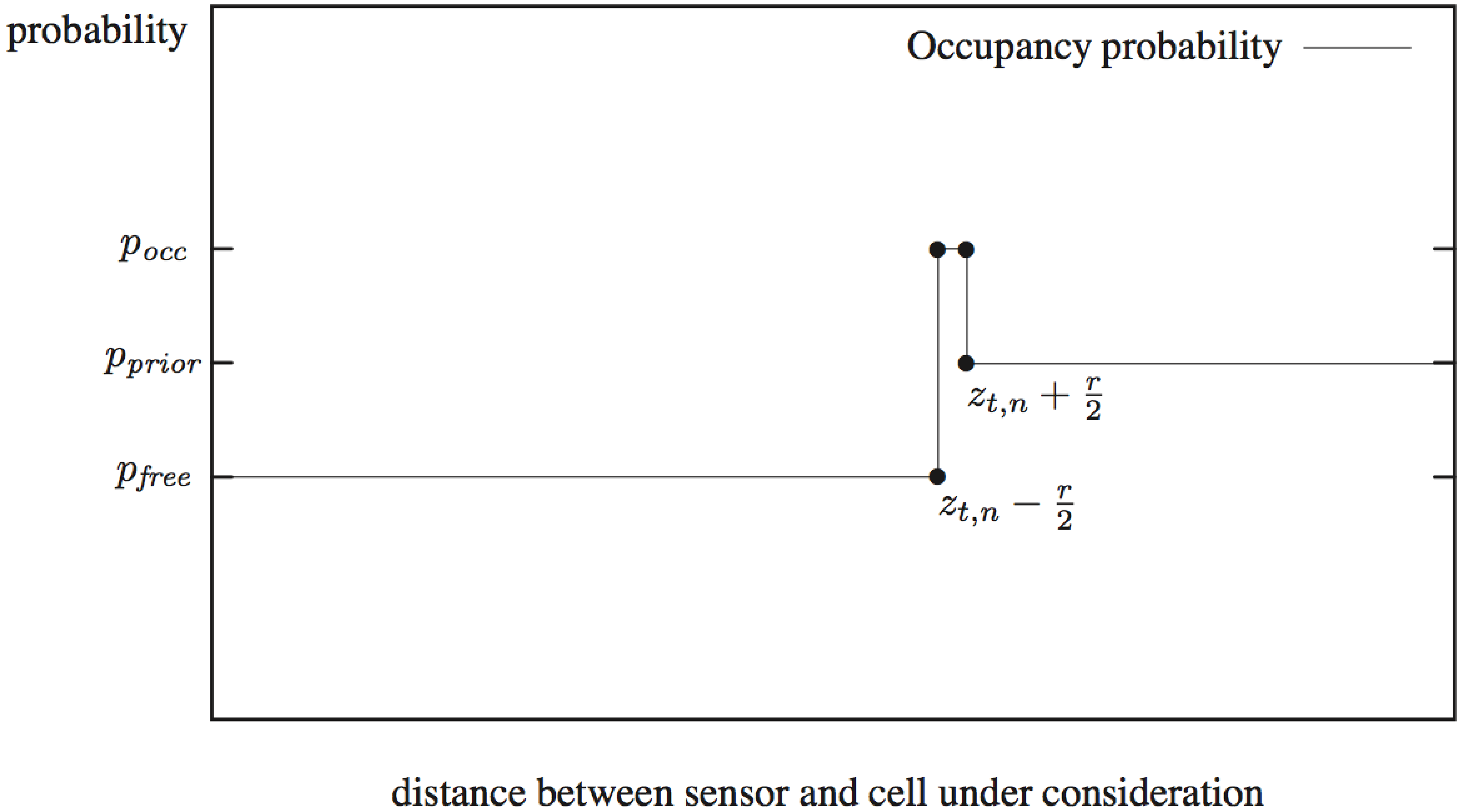
# Occupancy Grid Mapping

- Moravec and Elfes proposed occupancy grid mapping in the mid 80'ies
- Developed for noisy sonar sensors
- Also called "mapping with known poses"

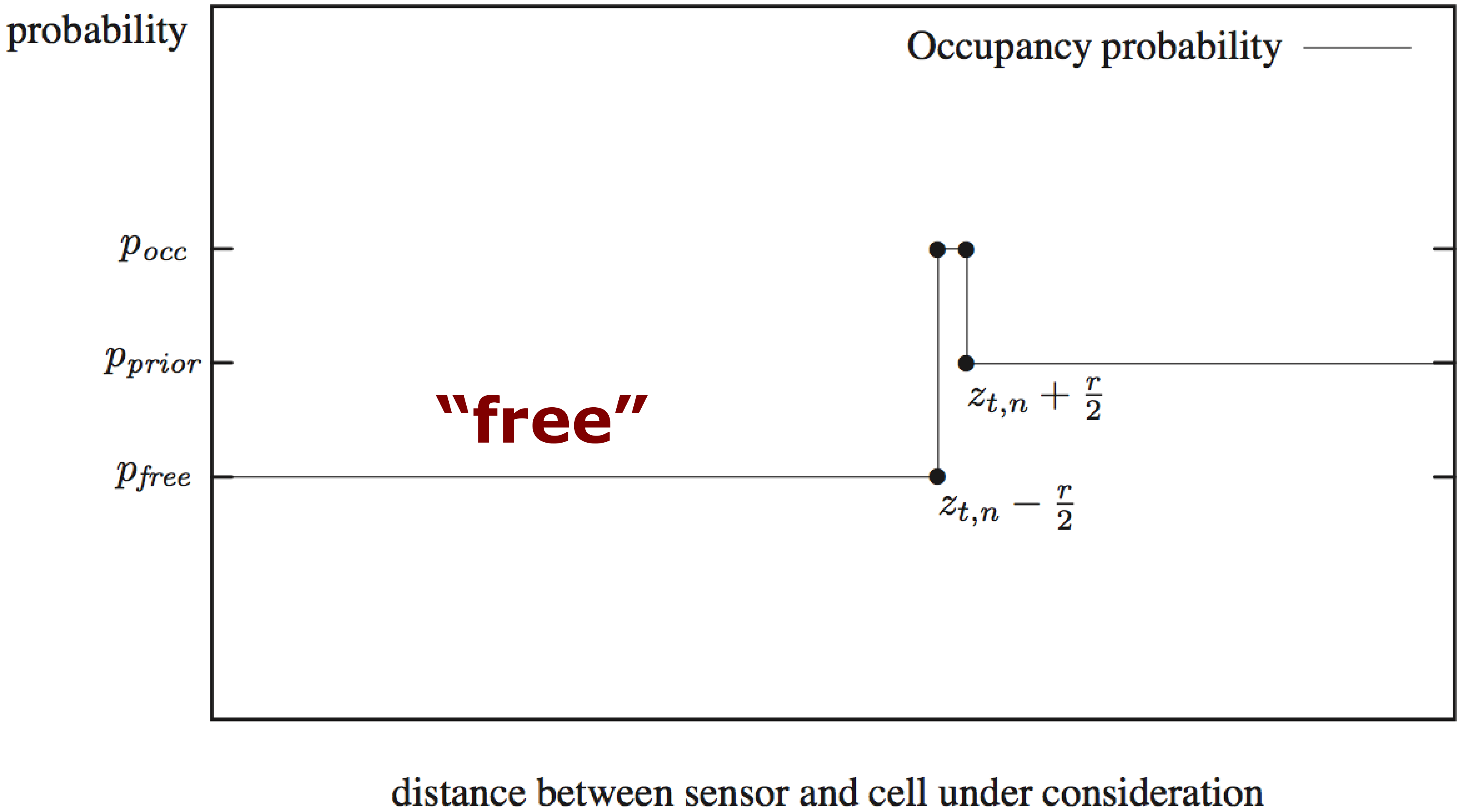
# 2D Laser Scanner Inverse Model



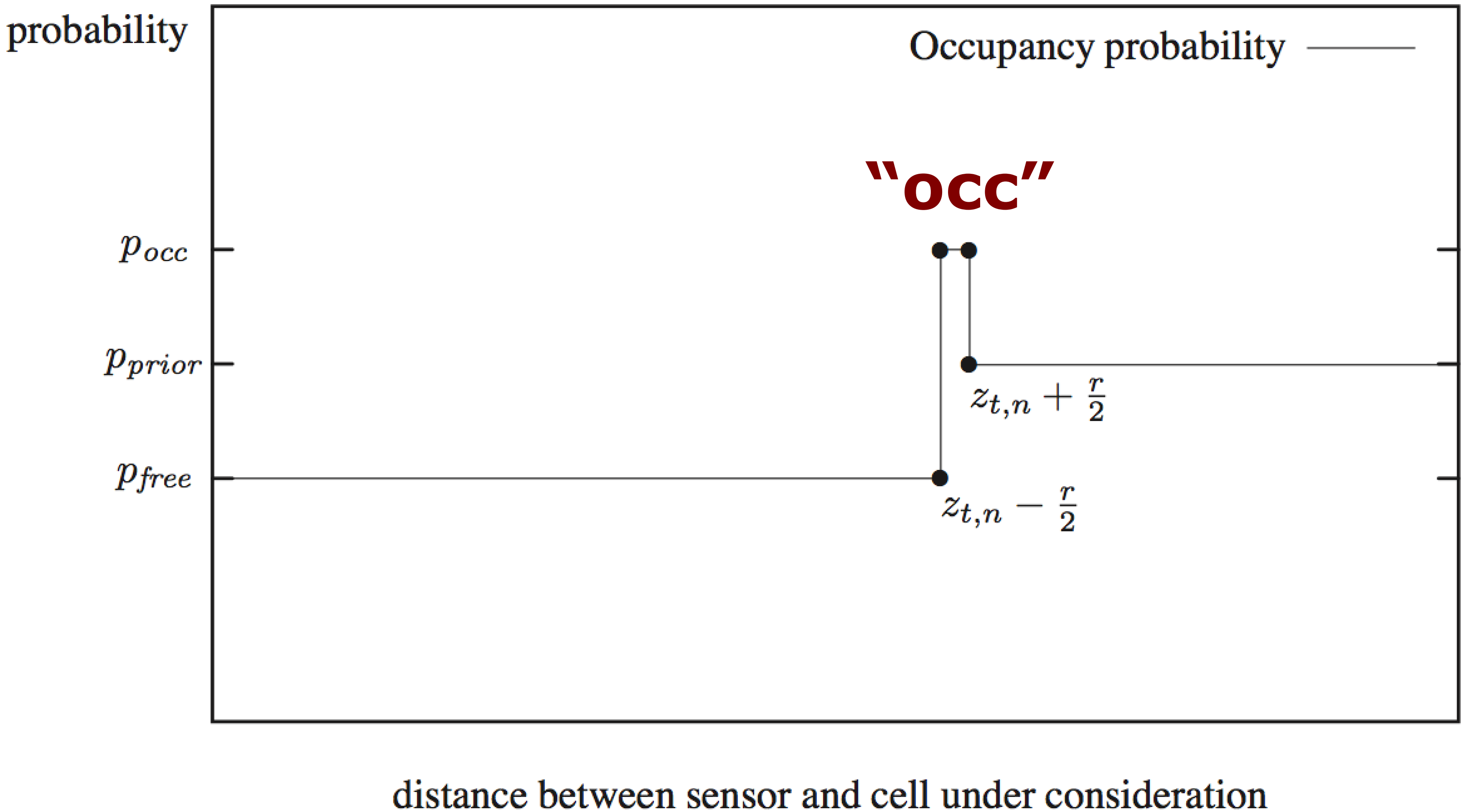
# Inverse Sensor Model for Laser Range Finders



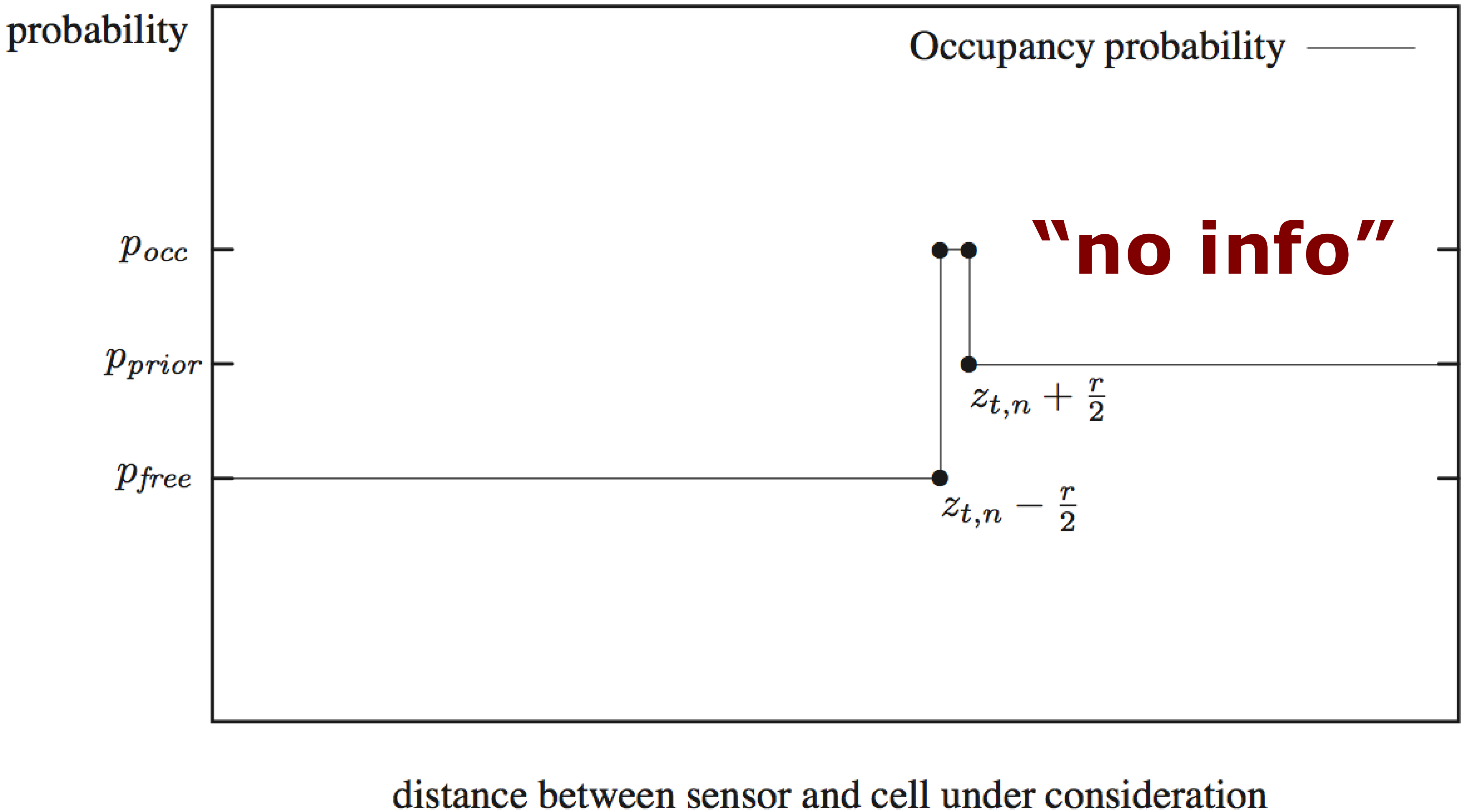
# Inverse Sensor Model for Laser Range Finders



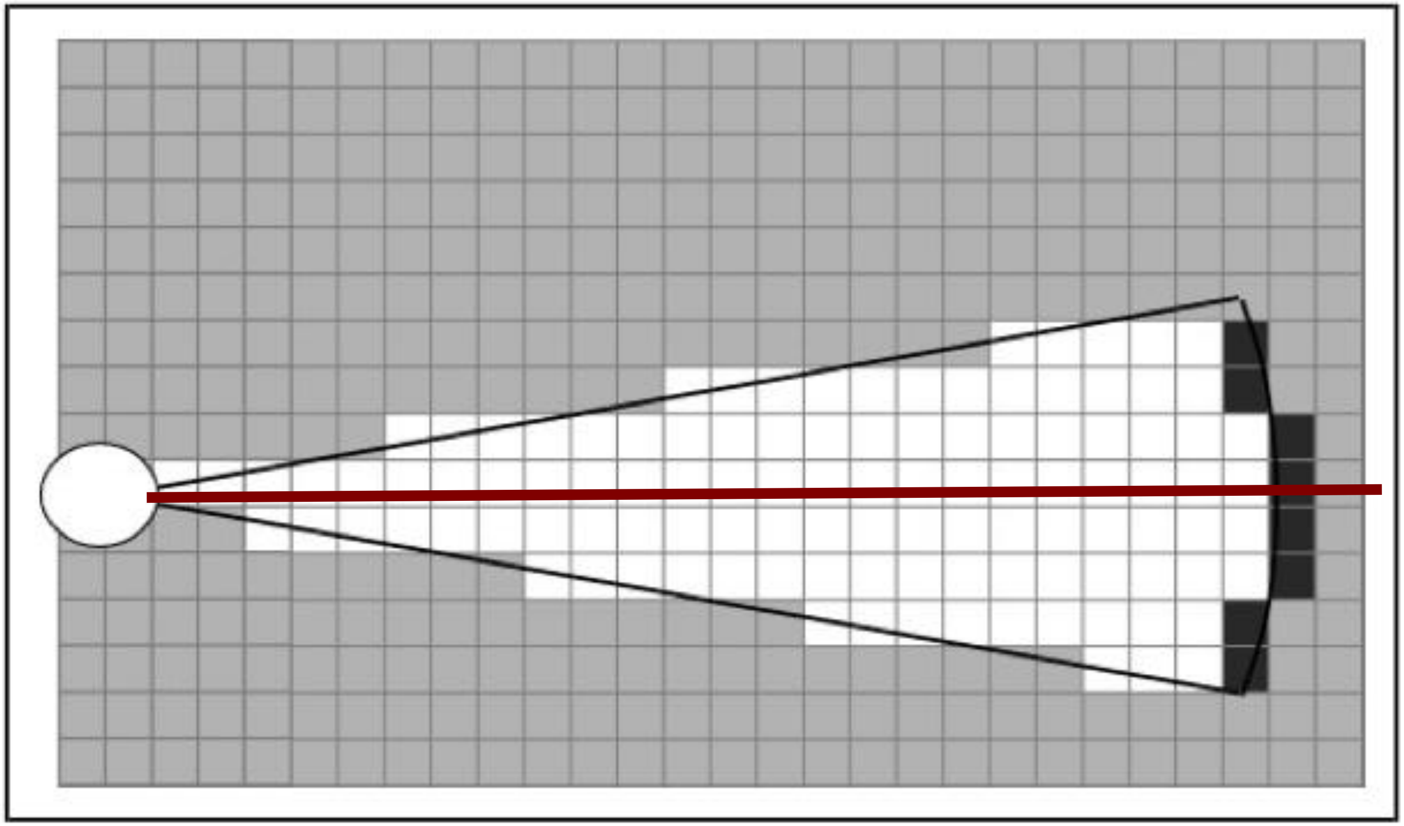
# Inverse Sensor Model for Laser Range Finders



# Inverse Sensor Model for Laser Range Finders



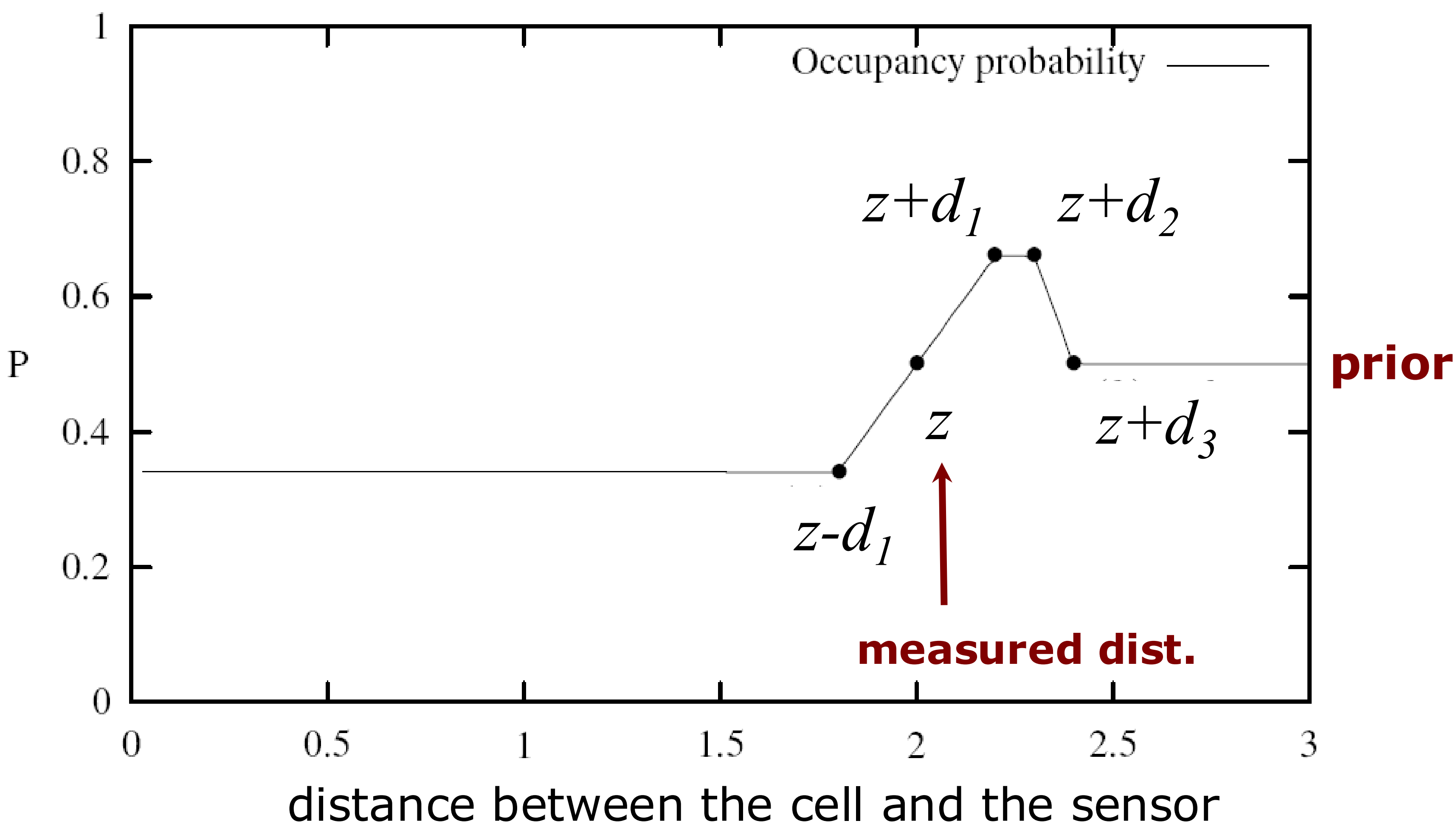
# Inverse Sensor Model for Sonar Range Sensors



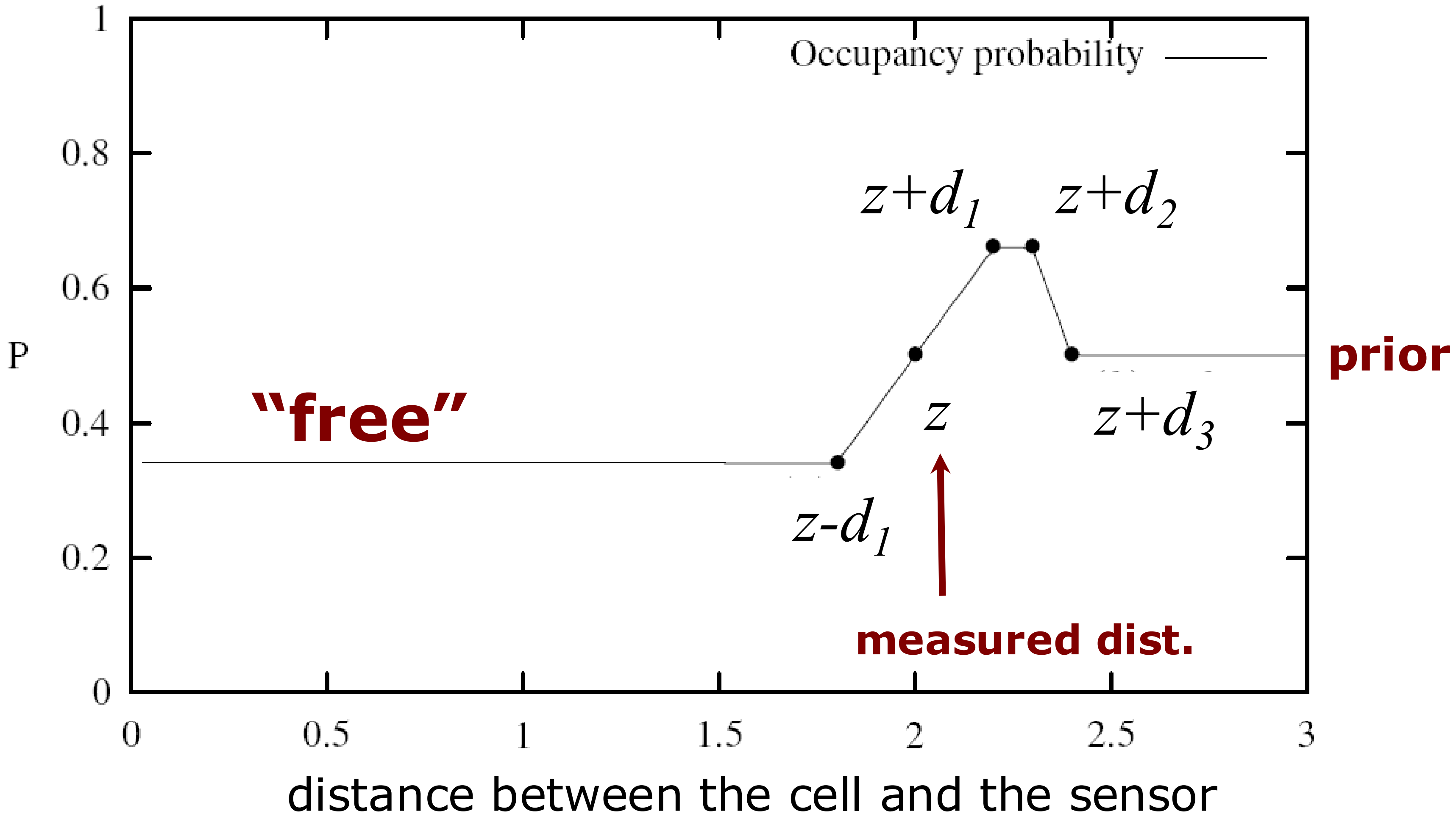
In the following, we only consider the cells along the optical axis (red line)

Courtesy: Thrun, Burgard, Fox

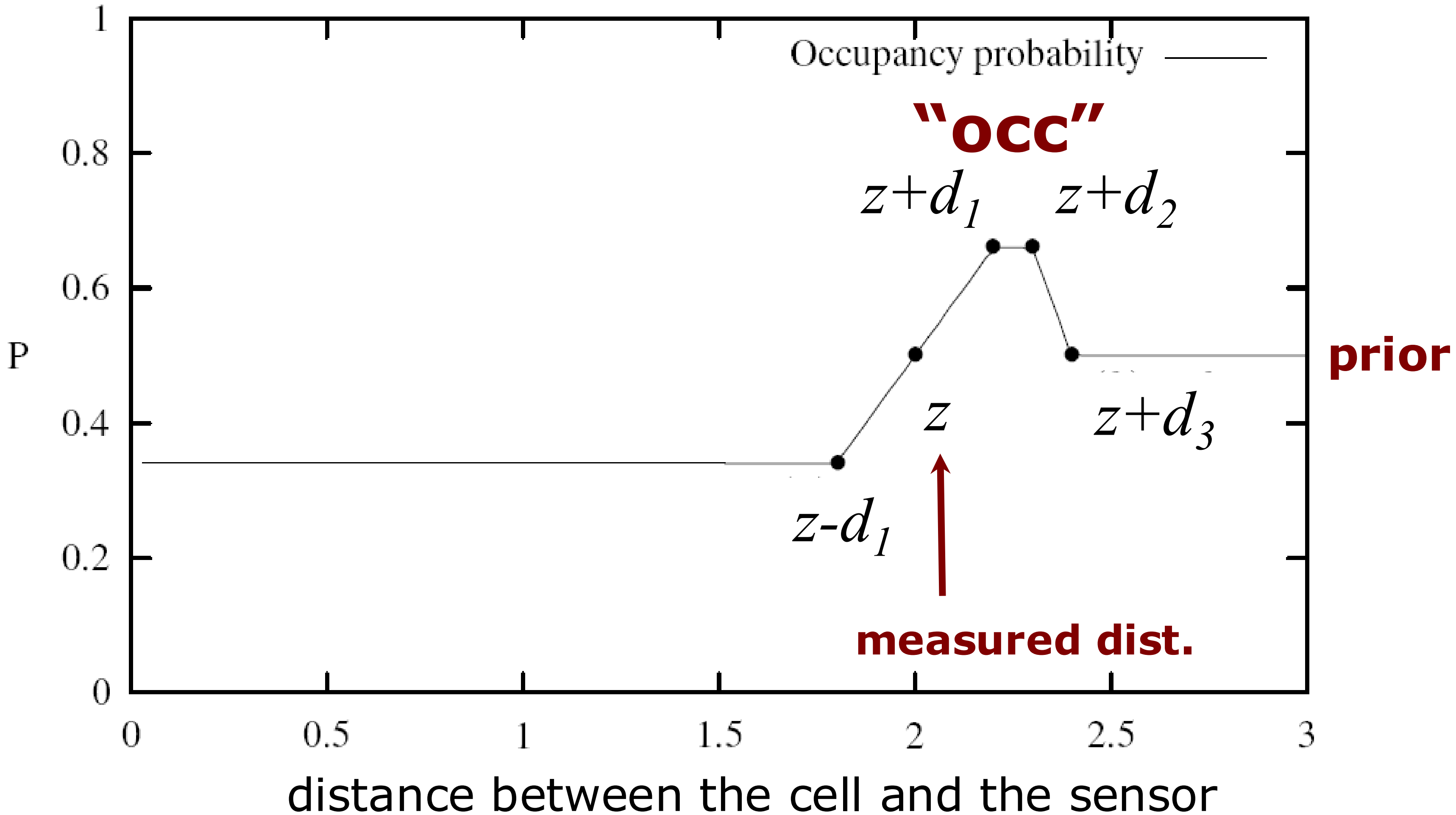
# Occupancy Value Depending on the Measured Distance



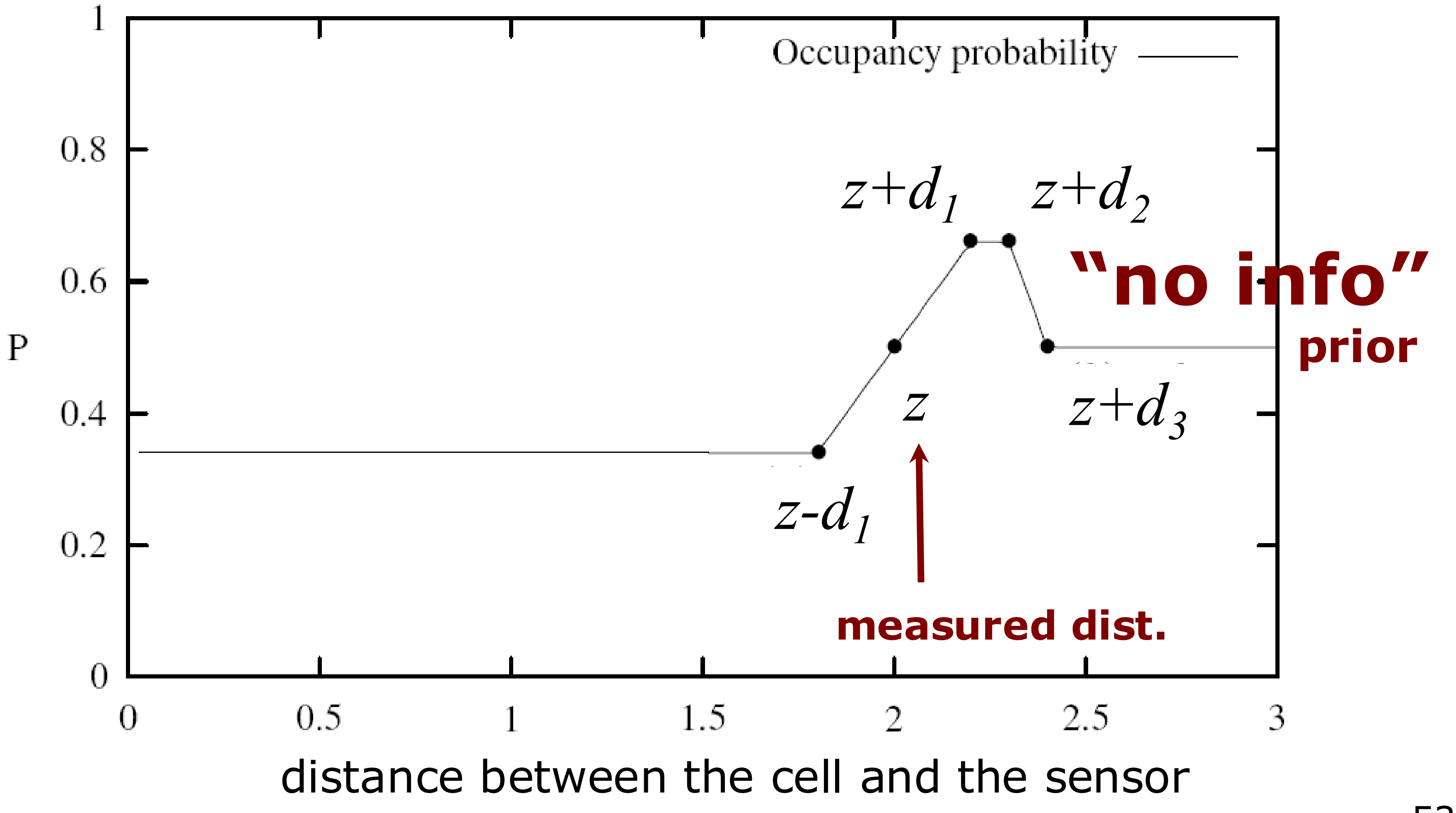
# Occupancy Value Depending on the Measured Distance



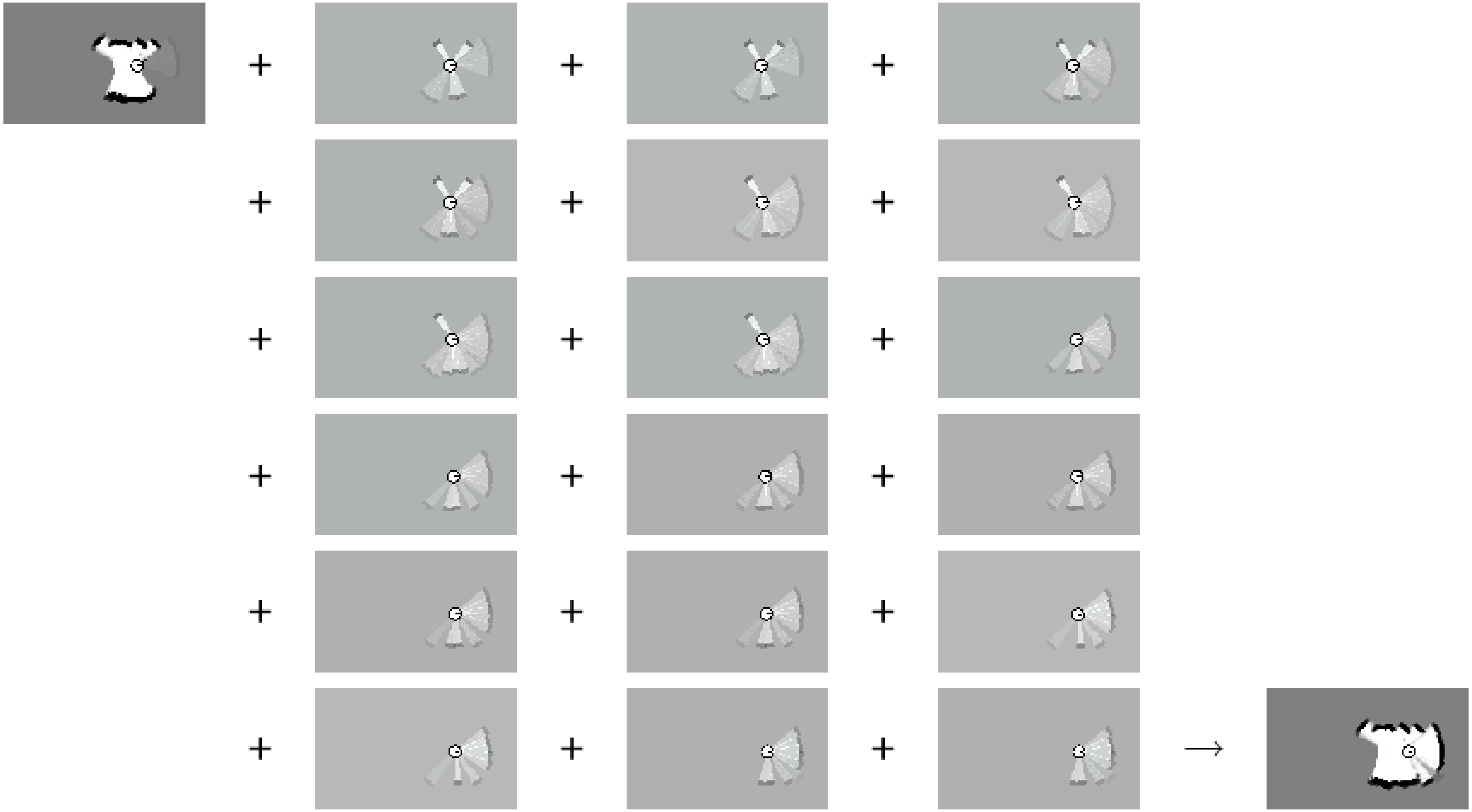
# Occupancy Value Depending on the Measured Distance



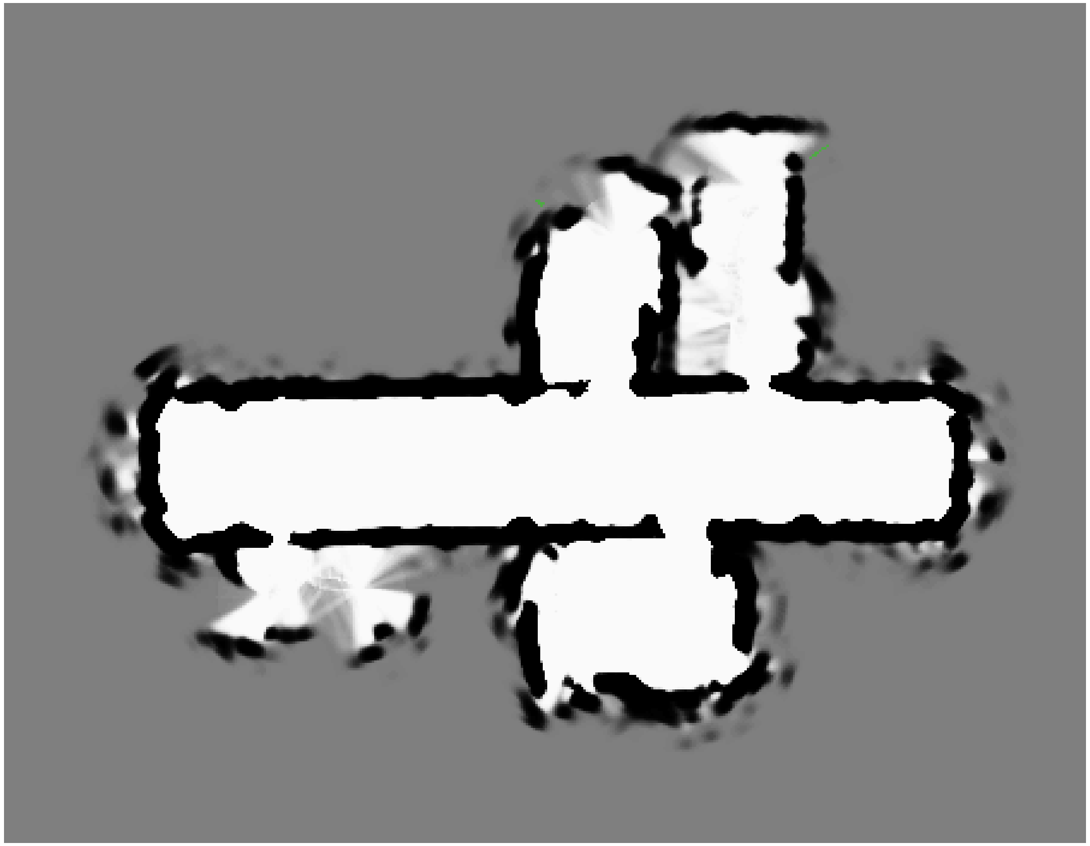
# Occupancy Value Depending on the Measured Distance



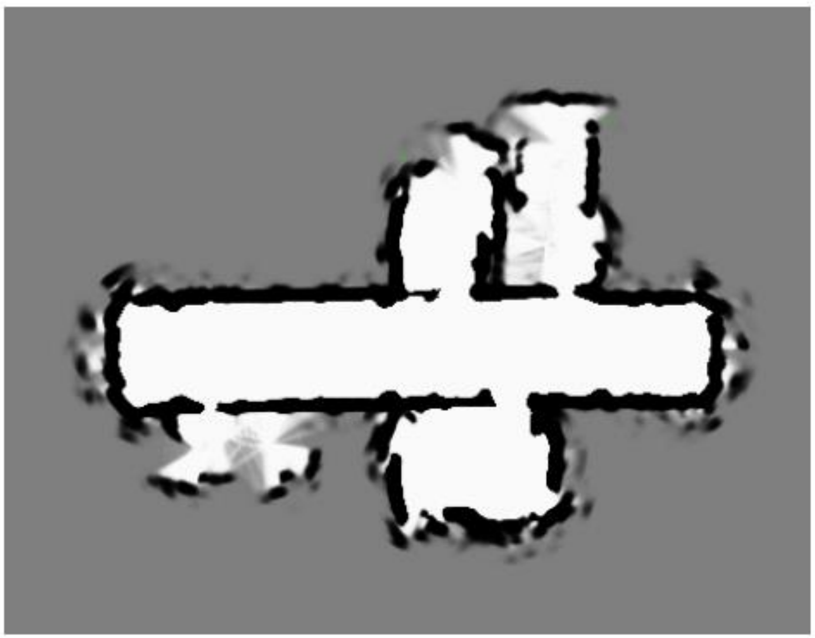
# Example: Incremental Updating of Occupancy Grids

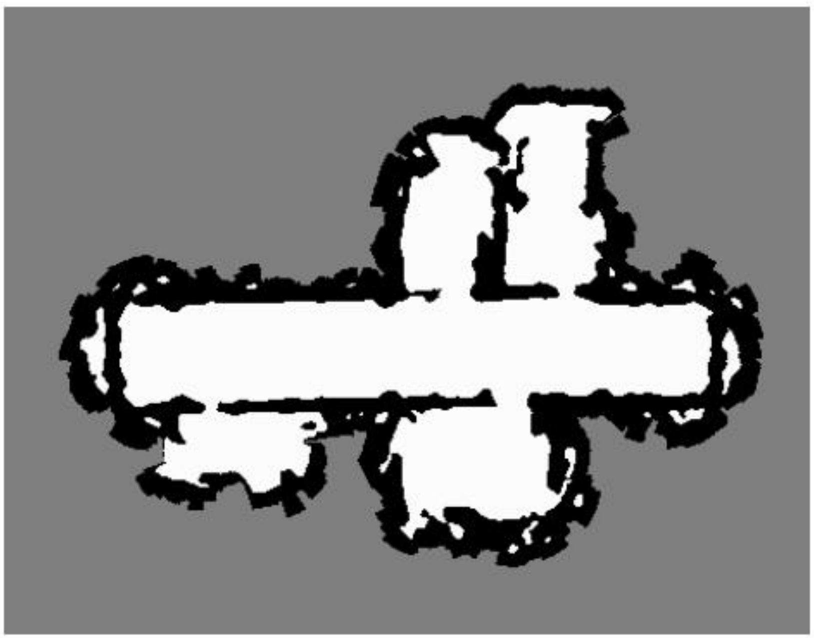


# Resulting Map Obtained from 24 Sonar Range Sensors



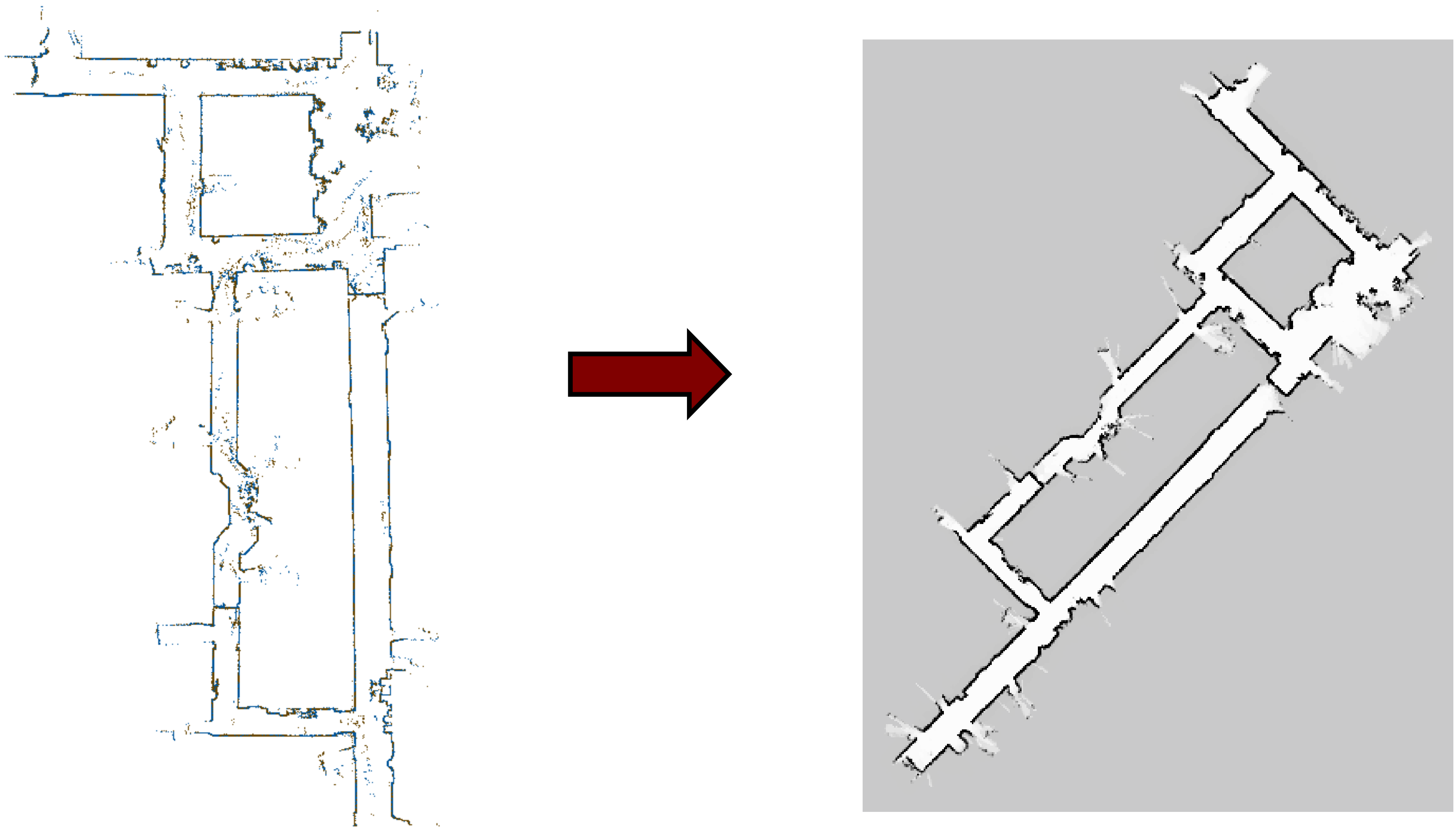
# Resulting Occupancy and Maximum Likelihood Map





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

# Occupancy Grids
# From Laser Scans to Maps



Courtesy: D. Hähnel

# Which Cells to Update for a Single Laser Beam?

Bresenham’s line algorithm

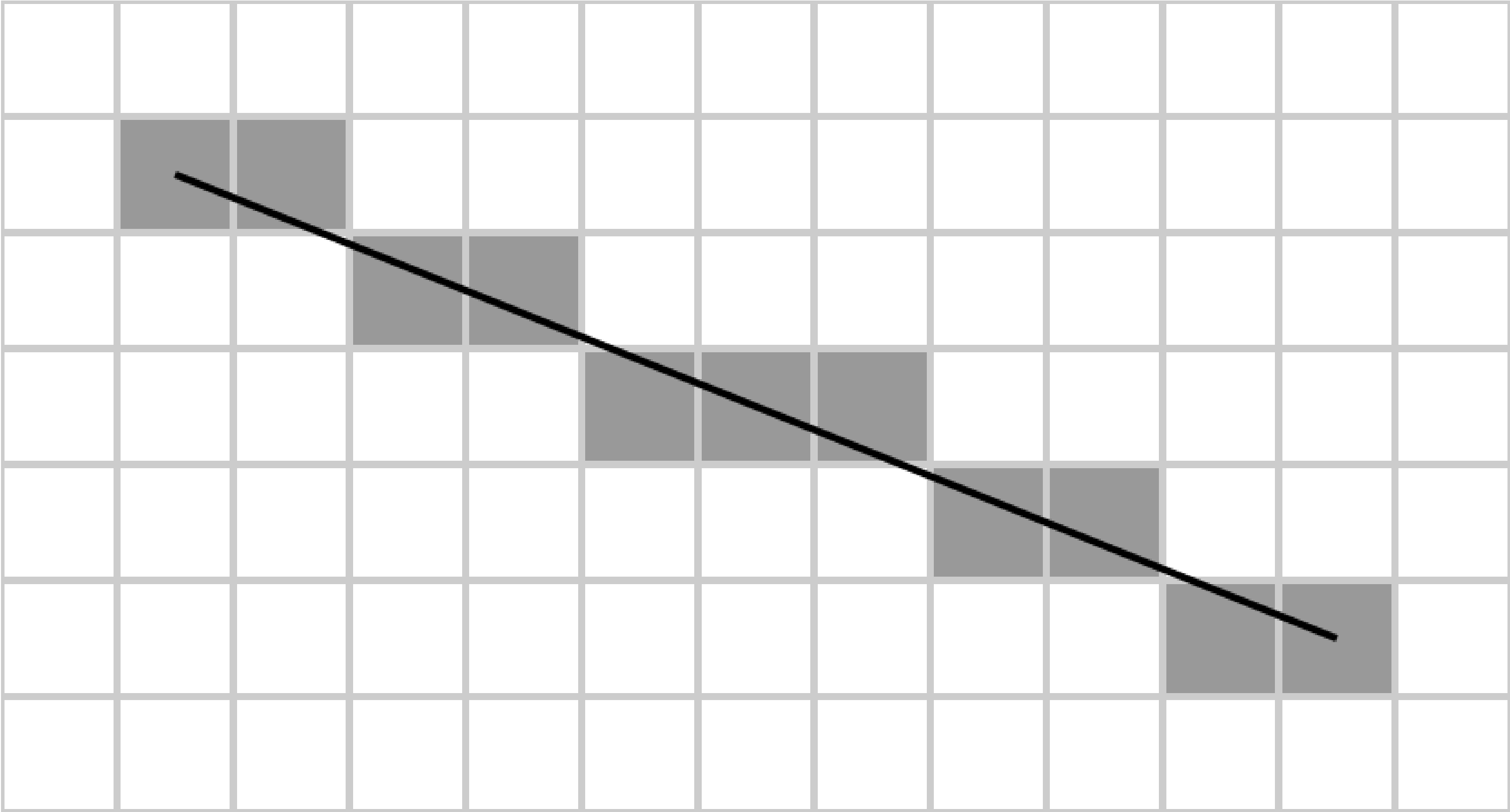
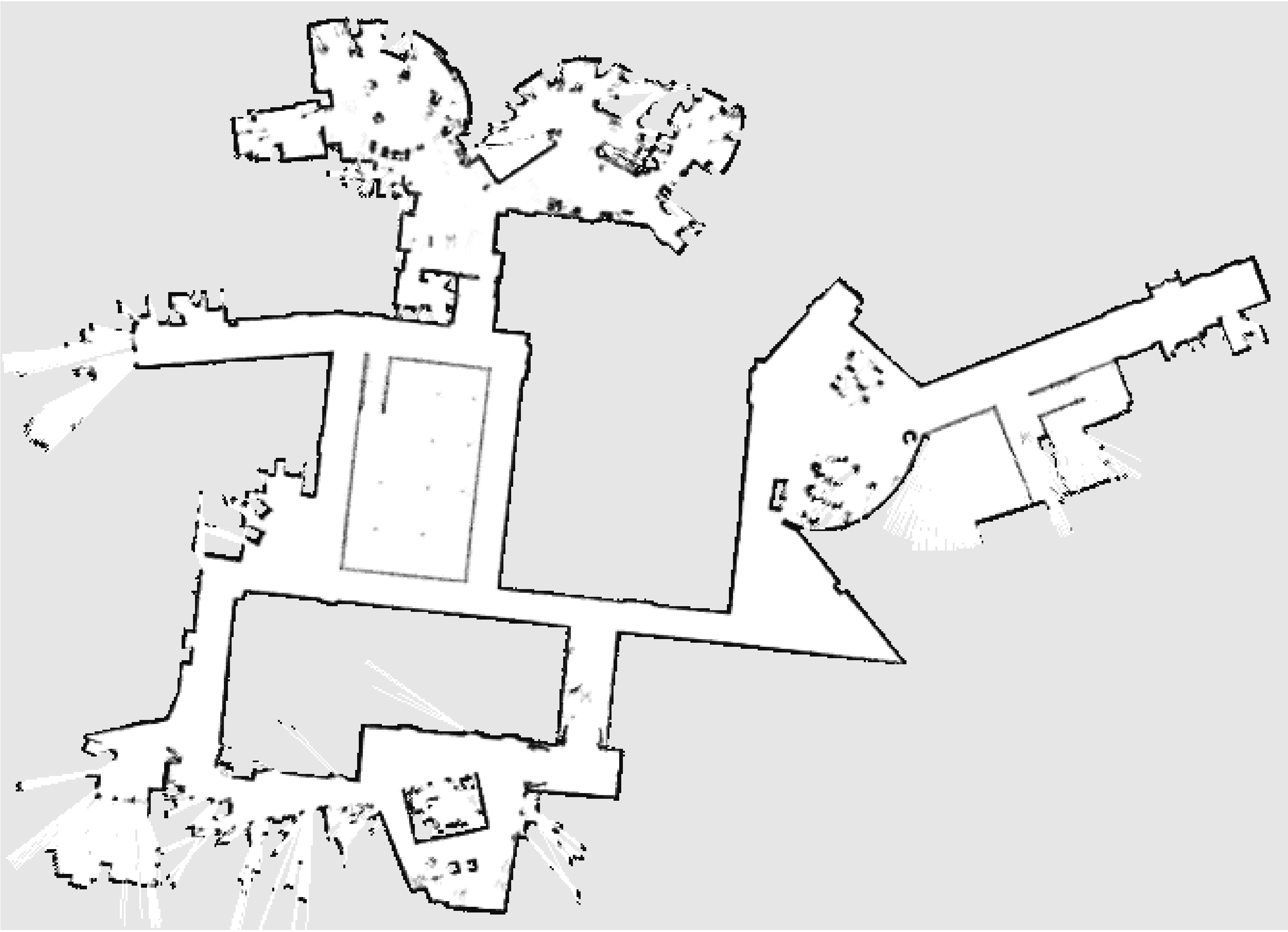
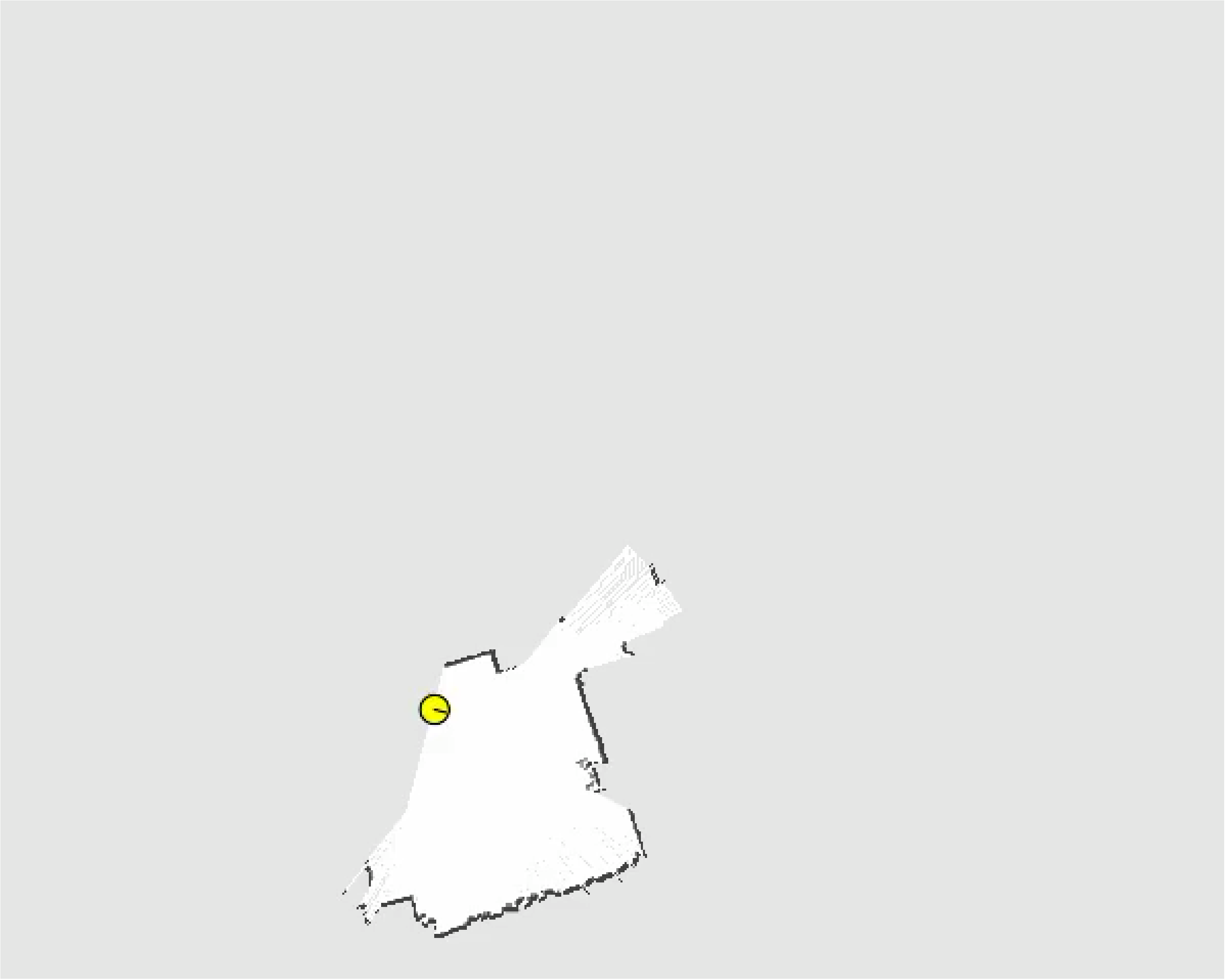


Image courtesy: Wikipedia

# Example: MIT CSAIL 3rd Floor



# Example: Uni Freiburg Bld. 106



# Summary

- 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
- We assumed known poses (so far)

# Literature

**Static state binary Bayes filter**

- Thrun et al.: “Probabilistic Robotics”, Chapter 4.2

**Occupancy Grid Mapping**

- Thrun et al.: “Probabilistic Robotics”, Chapter 9.1+9.2